

MOTION CONTROL OF A MOBILE ROBOT FOR UNEVEN TERRAIN

CALVEN CHRISTOPHER



**BACHELOR OF MECHATRONICS ENGINEERING WITH
HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

2019

MOTION CONTROL OF A MOBILE ROBOT FOR UNEVEN TERRAIN

CALVEN CHRISTOPHER

**A report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours**



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

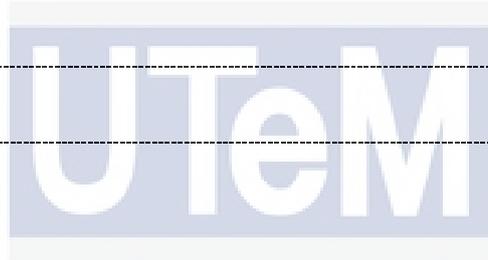
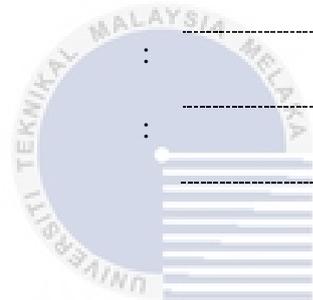
DECLARATION

I declare that this thesis entitled “MOTION CONTROL OF A MOBILE ROBOT FOR UNEVEN TERRAIN is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name :

Date :



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPROVAL

I hereby declare that I have checked this report entitled “MOTION CONTROL OF A MOBILE ROBOT FOR UNEVEN TERRAIN” and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Mechatronics Engineering with Honours

Signature :

Supervisor Name :

Date :



DEDICATIONS

To my beloved mother and father



ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Madya Dr. Mariam binti Md. Ghazaly, for her guidance, support, and encouragement throughout this project. Without her word of wisdoms, this project will not be done. I would also like to thank our Final Year Project coordinator, Assoc. Prof. Dr. Ahmad Zaki bin Haji Shukor, for guiding us throughout the process of executing out project, and to Pn. Fadilah bt Abdul Azis, my Seminar 2 Panel, for her guidance on the post editing of this thesis report.

To my fellow classmates and friends, I thank them for their moral support and knowledge that they shared with me. Lastly, I am very grateful to my family who are supporting me far from home.



ABSTRACT

Since the mobile robotic platform was first introduced in the '50s, it was a goal among researchers to develop an algorithm for the platform to be capable of generating a collision-free path for the robot to follow. Nowadays, the application of the algorithm can be seen in most mobile robots anywhere from the autonomous cleaning robot called Roomba, to the exploration rover Curiosity on Mars, and even the autonomous self-driving Tesla car. However, the most algorithm has been designed for robots navigating on a smooth surface. The problem with a rough surface is that the robot will experience slippage and lose some control of itself. Other than that, an obstacle such as hills and holes cannot be seen by standard mobile robots. Therefore, the purpose of this project is to develop a motion controller of a mobile robot specifically for uneven terrain. In this project, an Inertial Measurement Unit sensor is used together with a 180 degrees LIDAR system to detect the terrain surface orientation under the robot and possible obstacles in front of the robot. The speed of the robot is controlled based on the surface roughness for torque control and in the case when the robot meets an obstacle, it can determine the distance between itself and the obstacle to control its approaching speed. This helps give time to the robot to find an alternative path around the obstacle. Experimental methods are used to collect the information needed and it is graphed for different types of surface including tiles, asphalt, rocks, and hills.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRAK

Sejak platform robot bergerak pertama kali diperkenalkan pada tahun 50'an, telah menjadi impian para pengkaji untuk membangunkan sebuah algoritma untuk platform robot bergerak agar robot tersebut dapat mencari laluan bebas perlanggaran secara sendiri. Pada masa kini, pengaplikasian algoritma ini dapat dilihat pada kebanyakan robot bergerak seperti robot pembersih yang bernama Roomba, rover explorasi Curiosity yang berada di Mars, dan juga kereta pandu sendiri Tesla. Akan tetapi, algoritma ini hanya digunakan untuk robot yang bergerak di permukaan yang rata. Permukaan yang tidak rata akan menyebabkan robot mengalami kelicinan pada tayar yang akan menyebabkan robot hilang kawalan semasa bergerak. Selain itu, halangan-halangan seperti bukit dan lubang tidak dapat dikesan oleh robot biasa. Oleh itu, tujuan projek ini adalah untuk membina pengawal gerakan untuk robot bergerak pada permukaan yang tidak rata. Dalam projek ini, sebuah sensor Inertial Measurement Unit digunakan bersama sistem LIDAR 180 darjah bagi membaca bentuk permukaan di bawah robot dan mengesan halangan-halangan di depan robot. Kelajuan robot ini dikawal berdasarkan permukaan tanah untuk mengawal tork robot. Sekiranya terdapat halangan di hadapan, robot tersebut dapat mengenalpasti jarak halangan tersebut untuk mengawal kelajuannya ketika menghampiri halangan tersebut. Ini dapat memberi masa bagi membantu robot tersebut mencari laluan alternatif lain untuk melepasi halangan tersebut. Kaedah eksperimen digunakan dalam projek ini bagi mengumpul maklumat yang diperlukan dan maklumat itu akan disusun dalam bentuk graf bagi setiap permukaan seperti jubin, asphalt, batu dan kawasan berbukit.

TABLE OF CONTENTS

	PAGE
DECLARATION	
APPROVAL	
DEDICATIONS	
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
ABSTRAK	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF SYMBOLS AND ABBREVIATIONS	xiv
LIST OF APPENDICES	xv
CHAPTER 1 INTRODUCTION	1
1.1 Overview	1
1.2 Project Background	1
1.3 Motivation	2
1.4 Problem Statement	3
1.5 Objective	4
1.6 Scope	4
CHAPTER 2 LITERATURE REVIEW	6
2.1 Overview	6
2.2 Types of Mobile Robots	6
2.2.1 Wheeled Mobile Robot	6
2.2.1.1 Differential Drive	7
2.2.1.2 Synchronous Drive	8
2.2.1.3 Tricycle Drive	9
2.2.1.4 Ackermann Steer (Car Drive)	9
2.2.2 Legged Mobile Robot	10
2.3 Type of Sensors	11
2.3.1 Sensors for Obstacle Detection	11
2.3.1.1 Ultrasonic Sensor	12
2.3.1.2 Distance Infrared (IR) Sensor	12
2.3.1.3 Camera Sensor	13
2.3.2 Sensors for Robot Orientation	14
2.3.2.1 Accelerometer	14
2.3.2.2 Gyroscope	15

2.3.2.3 Magnetometer	16
2.3.3 Sensors for Robot Speed	16
2.3.3.1 Accelerometer	16
2.3.3.2 Optical Encoder	17
2.4 Type of Common Control System	17
2.4.1 Proportional-Integral-Derivative (PID) Controller	17
2.4.2 Fuzzy Logic Controller (FLC)	18
2.5 Overall Summary	19
CHAPTER 3 METHODOLOGY	20
3.1 Overview	20
3.2 Project Overview	21
3.3 Mobile Robot Design	21
3.4 Experiment 1: Detecting Surface Orientation	23
3.4.1 Designing Inertial Measurement Unit (IMU) Sensor's Algorithm	24
3.4.2 Open Loop Test of IMU Sensor	27
3.4.3 Intergration of IMU sensor and Motor Control Using Fuzzy Logic Control System.	28
3.5 Experiment 2: Obstacle Detection	31
3.5.1 Open Loop Test of Ultrasonic Sensor	32
3.5.2 Closed Loop Test of Ultrasonic Sensor and Motor Speed Control Using Fuzzy Logic Controller	34
3.5.3 Heading Selection	37
3.6 Designing Motion Control for Uneven Terrain	39
CHAPTER 4 RESULTS AND DISCUSSIONS	43
4.1 Experiment 1: Detecting Surface Orientation	43
4.1.1 Open Loop Test of IMU Sensor	43
4.1.2 Intergration of IMU sensor and Motor Control Using Fuzzy Logic Control System.	44
4.2 Experiment 2: Obstacle Detection	48
4.2.1 Open Loop Test of Ultrasonic Sensor	48
4.2.2 Closed Loop Test of Ultrasonic Sensor and Motor Speed Control Using Fuzzy Logic Controller	50
4.2.3 Heading Selection	51
4.3 Experiment 3: Designing Motion Control for Uneven Terrain	53
4.3.1 Tiles Surface	53
4.3.2 Asphalt Surface	54
4.3.3 Rocky Surface	55
4.3.4 Hill Surface	56
4.3.5 Overall Behavior of FLC System	57
CHAPTER 5 CONCLUSION AND RECOMMENDATIONS	58
5.1 Conclusion	58
5.2 Future Works	59
REFERENCES	60
APPENDICES	63

LIST OF TABLES

Table 2.1: Movement of robot based on wheel's direction configuration	8
Table 3.1 Mapping of Tasks to Project's Problem Statements and Objectives.	20
Table 3.2: Pitch Orientation Measurements for Open Loop Test on a Rotating Platform Rig	28
Table 3.3: Fuzzy Rules for Motor Speed Controller	30
Table 3.4: Behavior of Mobile Robot on Uneven Terrain	31
Table 3.5: Characteristic of the Ultrasonic sensor when measuring the distance of an obstacle	33
Table 3.6: Fuzzy Rule of the FLC system	36
Table 3.7: Characteristic Of Obstacle Distance And Motor Speed	37
Table 3.8: Distance Reading for a 180° scan	38
Table 3.9: Evaluation of Possible Alternative Route Heading	38
Table 3.10: Fuzzy Rules for Experiment 3	39
Table 3.11: Close Loop Test for FLC System in Experiment 3	42
Table 4.1: Evaluation Results of Possible Alternative Route Heading	52

LIST OF FIGURES

Figure 2.1: Example of Wheeled Mobile Robot	7
Figure 2.2: Example of Differential Drive Robot	7
Figure 2.3: Example of Synchronous Drive	8
Figure 2.4: Example of Tricycle Drive	9
Figure 2.5: Example of Ackermann Drive	10
Figure 2.6: Example of Legged Mobile Robot	11
Figure 2.7: HC-SR04 Ultrasonic Sensor	12
Figure 2.8: IR Distance Sensor GP2Y0A41SK0F	13
Figure 2.9: Example of Stereo Camera System	14
Figure 2.10: ADXL335 Accelerometer Sensor	15
Figure 2.11: L3G4200D Gyroscope Sensor	15
Figure 2.12: HMC5883L Magnetometer Sensor	16
Figure 2.13: Example of Optical Encoder Working Principle	17
Figure 2.14: Typical Block Diagram of PID Controller	18
Figure 3.1: Project Development Process (Flow Chart).	21
Figure 3.2: Right Side View of Mobile Robot	22
Figure 3.3: Right Side View of Mobile Robot	22
Figure 3.4: Orthographic View of Mobile Robot	23
Figure 3.5: Axis Frame of The Mobile Robot	24
Figure 3.6: Force vectors acting on the sensor in 3D spaces	25
Figure 3.7: Typical Block Diagram of Complementary Filter	26
Figure 3.8: Example of Pitch Orientation of Robot Correspond Directly to Surface Orientation	27

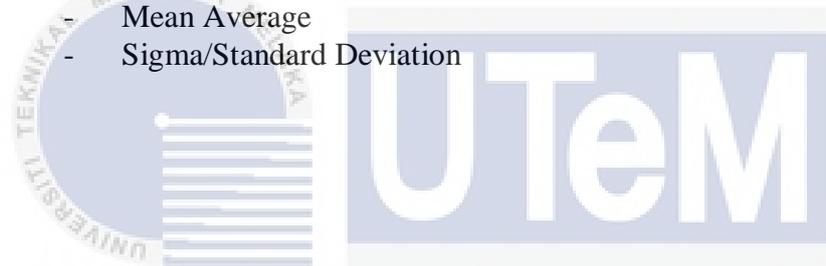
Figure 3.9: Open Loop Test setup Using a Rotating Platform Rig.	28
Figure 3.10: Block Diagram of Fuzzy Logic Controller for Motor Speed Control	29
Figure 3.11: Membership Function for Input Variable	30
Figure 3.12: Membership Function for Output Variable	30
Figure 3.13: Expected Behaviour of the FLC System	31
Figure 3.14: Setup for 100 cm Open Loop Test	33
Figure 3.15: Setup for 400 cm Open Loop Test	33
Figure 3.16: Close Loop Test Setup for Ultrasonic Sensor and Motor Control	35
Figure 3.17: FLC Block Diagram for Motor Control using Ultrasonic	35
Figure 3.18: Membership Function of the Ultrasonic reading	36
Figure 3.19: Memberhip Function of the Motor Output Speed	36
Figure 3.20: Expected Behaviour of Fuzzy logic controller	37
Figure 3.21: Experiment Setup for Heading Selection	38
Figure 3.22: Typical Fuzzy Controller Architecture	40
Figure 3.23: Block Diagram for FLC of Experiment 3	41
Figure 3.24: Surface Graph of Relationship between IMU sensor, Ultrasonic Sensor, and Motor Speed.	42
Figure 3.25: Type of Terrains Used for Experiment 3	42
Figure 4.1: Graph of Angle Reading ($^{\circ}$) vs Actual Angle($^{\circ}$)	43
Figure 4.2: Characteristic of The Pitch Orientation And Motor's Speed Over Time For Tiles Surface	45
Figure 4.3: Characteristic of The Pitch Orientation And Motor's Speed Over Time For Asphalt Surface	45
Figure 4.4: Characteristic of The Pitch Orientation And Motor's Speed Over Time For Rocky Surface	46

Figure 4.5: Characteristic of The Pitch Orientation And Motor's Speed Over Time For Hill Surface	47
Figure 4.6: Accuracy of Ultrasonic Sensor for 0 to 100 cm Range	49
Figure 4.7: Accuracy of Ultrasonic Sensor from 0 to 400 cm	50
Figure 4.8: Behavior Characteristic of the FLC System	51
Figure 4.9: Distance Reading for 180° Area	52
Figure 4.10: Characteristic Behavior of FLC System Used on Tiles Surface.	53
Figure 4.11: Characteristic Behavior of FLC System Used on Asphalt Surface	54
Figure 4.12: Characteristic Behavior of FLC System Used on Rocky Surface	55
Figure 4.13: Characteristic Behavior of FLC System Used on Hill Surface	56
Figure 4.14: Overall Behavior of The MISO FLC System	57



LIST OF SYMBOLS AND ABBREVIATIONS

AMR	-	Diameter
AI	-	Artificial Intelligence
WMR	-	Wheeled Mobile Robot
DC	-	Direct Current
DOF	-	Degree of Freedom
IMU	-	Inertial Measurement Unit
cm	-	Centimetre
%	-	percentage
IR	-	Infrared Receiver
RPM	-	Revolution per Minute
PID	-	Proportional-Integral-Derivative
SISO	-	Single Input Single Output
FLC	-	Fuzzy Logic Controller
MIMO	-	Multiple Input Multiple Output
θ	-	Degree
LIDAR	-	Light Detection and Ranging
\bar{x}	-	Mean Average
σ	-	Sigma/Standard Deviation



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF APPENDICES

APPENDIX A	EXPERIMENT 1 OPEN LOOP TEST RESULTS	63
APPENDIX B	EXPERIMENT 1 CLOSE LOOP RESULT FOR TILES SURFACE	63
APPENDIX C	EXPERIMENT 1 CLOSE LOOP RESULT FOR ASPHALT SURFACE	64
APPENDIX D	EXPERIMENT 1 CLOSE LOOP RESULT FOR ROCKY SURFACE	65
APPENDIX E	EXPERIMENT 1 CLOSE LOOP RESULT FOR HILL SURFACE	66
APPENDIX F	EXPERIMENT 2 OPEN LOOP RESULT	68
APPENDIX G	EXPERIMENT 2 CLOSE LOOP TEST RESULT	71
APPENDIX H	EXPERIMENT 2 HEADING SELECTION SCAN RESULT	83
APPENDIX I	EXPERIMENT 3 FLC BEHAVIOR ON TILES SURFACE	83
APPENDIX J	EXPERIMENT 3 FLC BEHAVIOR ON ASPHALT SURFACE	84
APPENDIX K	EXPERIMENT 3 FLC BEHAVIOR ON ROCKY SURFACE	85
APPENDIX L	EXPERIMENT 3 FLC BEHAVIOR ON HILL SURFACE	86
APPENDIX M	MAIN PROGRAMMING CODE FOR MOTION CONTROL	87
APPENDIX N	HEADER FILE FOR FLC INFERENCE SYSTEM	96
APPENDIX O	SOURCE CODE FOR ACCELEROMETER	96
APPENDIX P	HEADER FILE FOR ACCELEROMETER	97
APPENDIX Q	SOURCE CODE FOR ULTRASONIC	98
APPENDIX R	HEADER FILE FOR ULTRASONIC	98
APPENDIX S	SOURCE CODE FOR I2C	99
APPENDIX T	HEADER FILE FOR I2C	100
APPENDIX U	SOURCE CODE FOR GYROSCOPE	100
APPENDIX V	HEADER FILE FOR GYROSCOPE	101

APPENDIX W SOURCE CODE FOR MOTOR CONTROLLER	102
APPENDIX X HEADER FILE FOR MOTOR CONTROLLER	102



CHAPTER 1

INTRODUCTION

1.1 Overview

This chapter will present the project background on the motion control design of a mobile robot for uneven terrain, motivation for the project, problem statement that assist in making the proposal of the project, objectives that acts as a guideline throughout the project, scopes that shows the limitation of the project in terms of outcome, and the outline of the project.

1.2 Project Background

Nowadays, the application of the motion control system can be found in almost every type of machine and robots due to its ability to control moving parts of the system in a controlled or adaptive manner[1]. This control system can be either an open loop or closed loop system and is usually comprise of a motion controller, amplifiers, sensors, transducers, and actuators. A typical open loop system can be seen in a fan control where the controller sends a command signal through the amplifiers to the actuator, which is the motor, to rotate the fan[2]–[4]. In a closed loop system, a measuring device or sensor is added to the system to measure the output of the system and send it back to the controller[2], [4]. The motion controller will compare the measurement from the sensor with the desired outcome and compensate for any error by adjusting the command signal before sending it to the actuator again.

An Autonomous Mobile Robot (AMR) is one of the mobile robots that use the motion control system to help it navigate autonomously around obstacles in order to get from one point to another. This is achieved by getting information about the surrounding environment such as obstacle and position, to generate the appropriate behavior to overcome

the obstacle[5]. This has further helped in making it easier for an exploration mission to be done autonomously on an unknown environment such as underwater or other planets[6]. Such features also make an AMR the most innovative solution for the market, especially in the industrial manufacturing field where the AMR is used to transport materials automatically from one station to another without intervention from a human[7]. This has since increased the work efficiency of the manufacturing process since the transportation of materials is a repetitive job in which human will grow tired over time.

In this project, a motion controller is designed to help a mobile robot navigate autonomously on an uneven surface. Information about the surface orientation and presence of obstacles is gathered through the sensors and sent to the motion controller where it will be used to compute the appropriate speed of the robot using fuzzy logic approach, and the possible heading of the robot to avoid the obstacle. Analysis on the input and output of the motion controller will be done to view the behavior of the motion controller on different types of terrains.

1.3 Motivation

Robots were first developed to do simple but repetitive physical task in the industrial section such as polishing steel tubes and cutting a metal sheet. As the field of robotics started to advance, the field of artificial intelligent (AI) began to emerge and started contributing more to the advancement of robotics. AI helps robots to make simple decisions based on the input information from sensors. Soon, more types of robots are being developed for a different specific purpose and each purpose has its own form with its own mechanical limitation such as workspace area.

Mobile robotic platform was first introduced back in the '50s and since then, it has been the main endeavor for every researcher to develop a motion control system for a mobile robotic platform capable of generating a collision-free path for the robot to follow [8]. To date, the application of the motion control system can be seen in most mobile robots anywhere from the autonomous cleaning robot called Roomba, to the exploration rover

Curiosity on Mars, and even the autonomous self-driving Tesla car. It has successfully made advancement for the realm of technology.

Most of the mobile robot was using motion control system for navigation on a smooth surface and most of it is using tires as the mode of locomotion. Although mobile robots for uneven terrain has already been developing, most of it uses the concept of legged robot as it can overcome obstacle without any difficulty. Only a few Wheeled Mobile Robots (WMR) have been developed with the purpose of navigating on rough terrains such as the exploration rover Curiosity and Opportunity. Although they work great in navigating on a rough surface, it still cost a lot to develop them. WMR concept has been never been a favorite for rough terrain exploration due to its limitation which gives a disadvantage on uneven terrain.

1.4 Problem Statement

The use of wheels as a mode of locomotion usually gives a huge disadvantage to a mobile robot when navigating on uneven terrain. this is because of tire slip which is the loss of traction due to surface irregularity, tire sinkage [9], or skidding. This can prevent the mobile robot from moving any further than its current position. In order to overcome this, the traction of the wheels needed to be increased by adjusting the speed of the wheel since the speed of the wheel will affect the time of contact between the wheels and the ground surface which then affects the wheels traction.

The second problem is that in order for a mobile robot to move around autonomously, the robot must have the ability to read its environment in order to make the optimum decision on how to navigate[10], [11]. Since most WMR is designed to navigate on smooth surface, there are less information about the method to measure the orientation of an uneven terrain. Since surface orientation and obstacles are the basic information needed to allow a mobile robot to navigate on uneven surface, a method for measuring the surface orientation, in terms of X and Y plane, and obstacles is needed to allow for uneven terrain navigation.

Lastly, a mobile robot must process the information about its surrounding and its own state so that it can make the necessary action that can help it navigate on uneven terrain.

Therefore, a suitable motion controller needs to be developed for helping the robot processing all the information efficiently[12].

1.5 Objective

These objectives will serve as a guideline throughout this project. In order to call the project successful, all of these project must be achieved.

The first objective is to develop a method to detect irregular surface. For a motion controller to be developed for navigating on uneven terrain, the robot must know the characteristic of the terrain in order for it to adapt or react properly.

Next, the second objective is to develop a method to detect obstacle. Other than ground surface, detecting obstacle is also important as it can stops a robot from navigating or cause damage if the robot bumps into obstacle. By detecting it, all consequences mentioned can be avoided

Lastly, to design a motion controller of mobile robot for uneven terrain. Having a motion control that can handle more than one input will be very helpful to a mobile robot since it helps the robot to react fast.

1.6 Scope

- Using a Differential drive type of Wheeled Mobile Robot
- Navigating around uneven terrain such as asphalts, rocks and hills
- Using Arduino Uno or Arduino Mega board as microcontroller
- Uses two 60:1 gear ratio DC motor to drive the robot.
- GY-85 9DOF IMU Sensor to measure robot's orientation

- HCSR04 Ultrasonic sensor with up to 400 cm range of detection
- 180 Degree Servo Motor



CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This chapter describes some important aspect in designing a motion control for mobile robot so that a suitable motion control can be applied specifically on uneven terrain. The first part in this chapter is the description of types of mobile robots exist nowadays. The second part describes the types of sensors that can be used to detect and measure the robot's environment, which includes obstacle (steep hill, wall, objects), orientation, and moving speed. The last part in this section describes the type of common control system which can be used to design a motion controller.

2.2 Types of Mobile Robots

Mobile robots are robotic system which has the capability to move around its environment[13]. The mobile robot can either be controlled manually or move around in an autonomous manner. Nowadays, there are a variety of mobile robot designs which can be categorized into wheeled mobile robot and legged mobile robot.

2.2.1 Wheeled Mobile Robot

Wheeled Mobile Robot (WMR), as the name implied, is a robot that uses motorizes wheel to navigate around a surface. An example of WMR can be seen in Figure 2.1. WMR can be further classified into several categories depending on their arrangement of driving and steering wheel[2], [12], [13]. These categories are differential drive, synchronous drive, tricycle drive and car drive[2].



Figure 2.1: Example of Wheeled Mobile Robot

2.2.1.1 Differential Drive

Differential drive is the most common type of WMR due to its simple programming and locomotion[2]. It has only two wheels which are driven independently on the same axis as shown in Figure 2.2. To control the robot, the speed and the direction of the rotating wheel is manipulated so that the robot can move straight, curve, and spin [14]. Table 2.1 shows the movement of the robot cause by manipulating each wheel at certain configuration.



Figure 2.2: Example of Differential Drive Robot

Table 2.1: Movement of robot based on wheel's direction configuration

Speed of each wheel (%)		Direction of each wheel		Movement
Left	Right	Left	Right	
100	100	Forward	Forward	Straight Forward
100	100	Backward	Backward	Straight Backward
100	50	Forward	Forward	Curve Right
50	100	Forward	Forward	Curve Left
100	100	Forward	Backward	Spin Right
100	100	Backward	Forward	Spin Left

However, this configuration comes with an issue which is wheel slipping. This problem occurs when the robot is moving on an irregular surface thus causing the robot to stall or change its direction. It also can affect the data measurement taken for performance testing. Tipping over is also a problem for this configuration due to irregular surface.

2.2.1.2 Synchronous Drive

Synchronous drive is a configuration where all the wheel is always steerable but at the same direction all the time as shown in Figure 2.3. The robot moves around without changing the direction of its chassis. This allows the robot to navigate in a limited space where rotating or making a curve turn is near impossible[2]. While it does reduce the possibility of tipping over as seen in differential drive robots, there is also the possibility of wheel slippage when there is a large variation on the irregular surface.

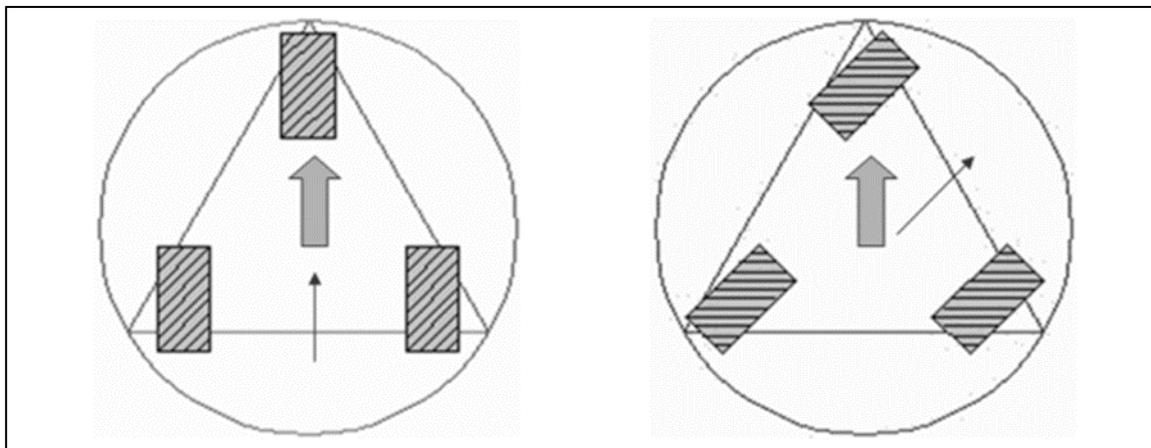


Figure 2.3: Example of Synchronous Drive

2.2.1.3 Tricycle Drive

A tricycle drive uses three wheels to move around a surface. The first type is having two driving rear wheels and 1 steering front wheel while the second type is having a driving and steering front wheel and two passive rear wheels[15]. This configuration solves the issue with wheel slippage seen on both differential driven and synchronous driven robot[2]. Figure 2.4 shows an example of tricycle drive.

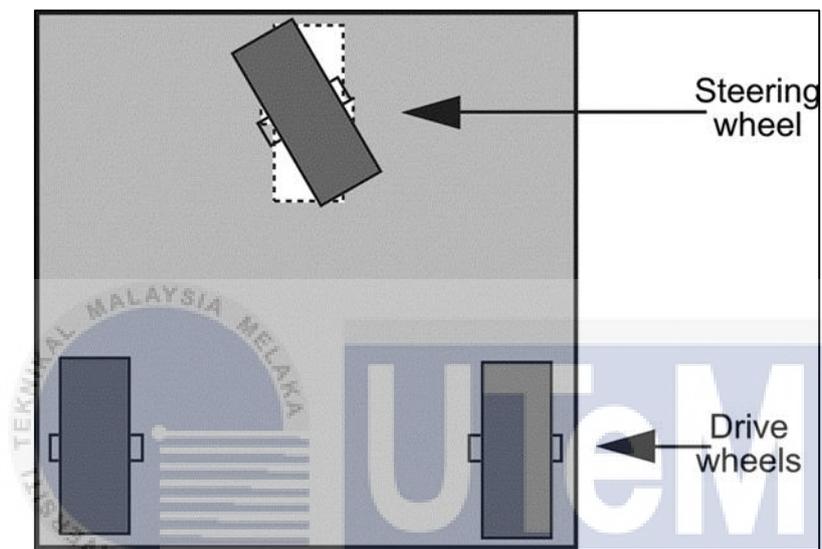


Figure 2.4: Example of Tricycle Drive

2.2.1.4 Ackermann Steer (Car Drive)

The Ackermann Steer configuration has four wheels where the two rear wheel is the driving wheel while the front two wheel is the steering wheel[2]. The difference of this configuration to the tricycle drive is that it has a link that is connecting the two front wheels to allow simultaneous steering for both wheels as shown in Figure 2.5.

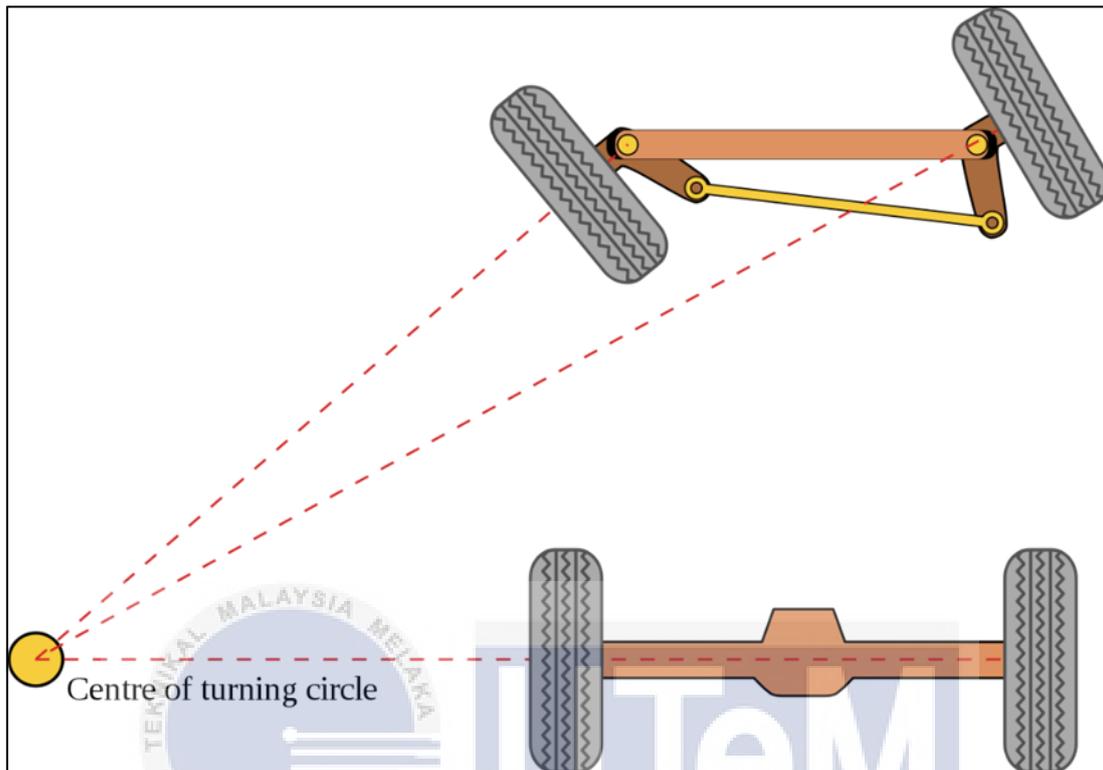


Figure 2.5: Example of Ackermann Drive

The advantage of tricycle and Ackermann drive compare to other configuration is that the velocity of the driving wheel is not considered into calculation for driving straight [15]. Although the derivation of kinematic model for this configuration is difficult, it is suitable for outdoor task that requires traveling over long distance.

2.2.2 Legged Mobile Robot

Nowadays, there are more advanced legged mobile robot being developed due to its advantage of mobility on rough terrain. The flexibility of the leg helps the robot to easily overcome large obstacle[16]. It is known that having more legs allows the robot to be balanced at all time than fewer legs due to having a larger base area[17]. However, having

this configuration increases the mechanical, electronic and programming complexity as well as building cost which is a big disadvantage. Figure 2.6 shows an example of legged mobile robot.

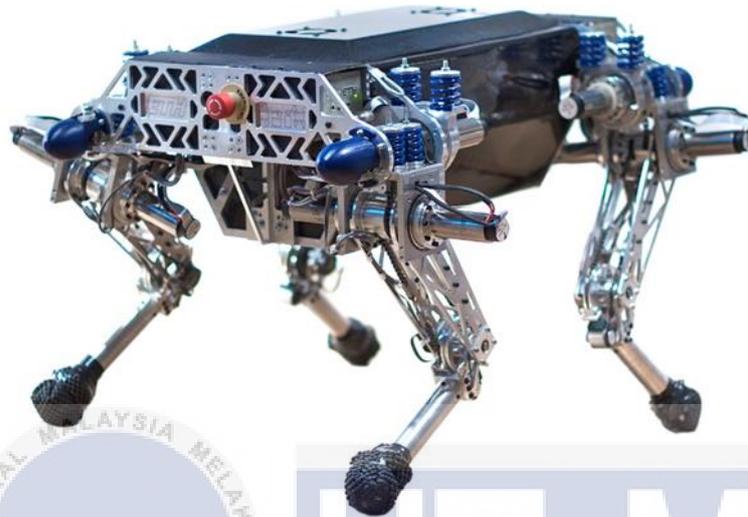


Figure 2.6: Example of Legged Mobile Robot

2.3 Type of Sensors

Sensors are important in designing a mobile robot as it serves as the eyes and ears of a robot. An autonomous mobile robot will take information about the surrounding through its sensors and process it to make navigation decision while a manned robot uses the information from the sensors for the same purpose and for real time analysis[2]. These sensors can be categorized according its purposes which are detecting obstacle, robot's orientation, and robot's moving speed.

2.3.1 Sensors for Obstacle Detection

Obstacle detection is an important odometry of an autonomous robot. It allows the robot to make decision whether to avoid the obstacle or go over it, depending on its

capability. Sensors that are suitable for this purpose are ultrasonic sensor, distance infrared (IR) sensor, and terrain mapping camera.

2.3.1.1 Ultrasonic Sensor

Ultrasonic sensor utilizes soundwave to measure the distance of an object. The sensor transmits an ultrasonic wave and detect the reflected wave from the object. By measuring the time taken between the transmitting and receiving of the wave, the sensor can determine its distance from the object [6], [18]. Figure 2.7 shows an example of ultrasonic.

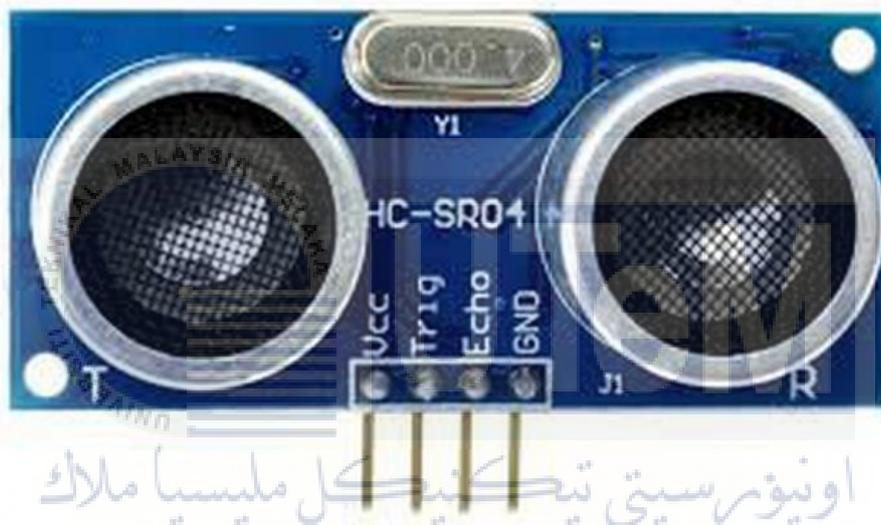
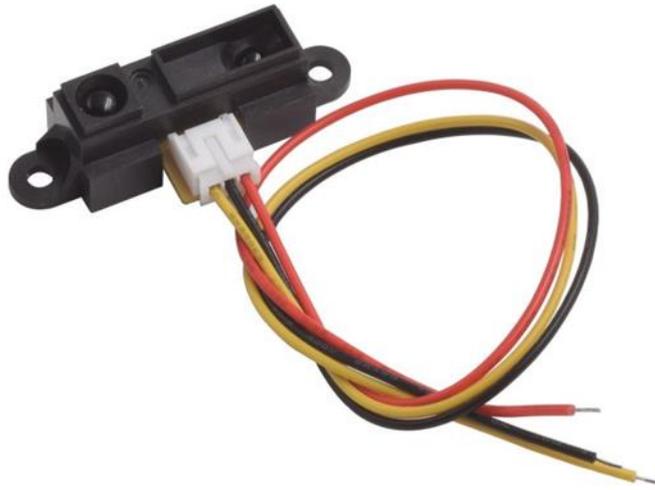


Figure 2.7: HC-SR04 Ultrasonic Sensor

2.3.1.2 Distance Infrared (IR) Sensor

Distance IR sensor uses the same concept as ultrasonic sensor but with a different medium which is light waves. The advantage of IR sensor compared to ultrasonic sensor is that it more suitable for close-proximity object detection. Farther object will produce error depending on the object surface and orientation [6], [18]. Figure 2.8 shows an example of Distance IR sensor.

Figure 2.8: IR Distance Sensor GP2Y0A41SK0F



2.3.1.3 Camera Sensor

The camera technology can actually also be used as an odometry sensor for an autonomous mobile robot. By using a stereo camera system, a disparity map can be computed from image produce by both camera for object detection. Computer vision method is then applied to filter out noises for better object detection [19], [20]. Figure 2.9 shows an example of stereo camera system



Figure 2.9: Example of Stereo Camera System

2.3.2 Sensors for Robot Orientation

Robot orientation is an important information for an autonomous robot in order for the robot to know its heading and direction as well as the state of the robot. Sensors in this category must be able to provide information about the pitch, roll, and yaw of the robot. Suitable sensors for this purpose are accelerometer, gyroscope, and magnetometer.

2.3.2.1 Accelerometer

An accelerometer measures dynamic and static forces that act upon the sensor on three axes. By computing the resultant of a static force, which is the gravity, and using the trigonometry equation of a right-angle triangle, the orientation of the robot can be measured. However, because this sensor also measures dynamic forces, it is sensitive to vibration and mechanical noise, thus producing noise in the measurement[16]. Figure shows an example of accelerometer sensor.

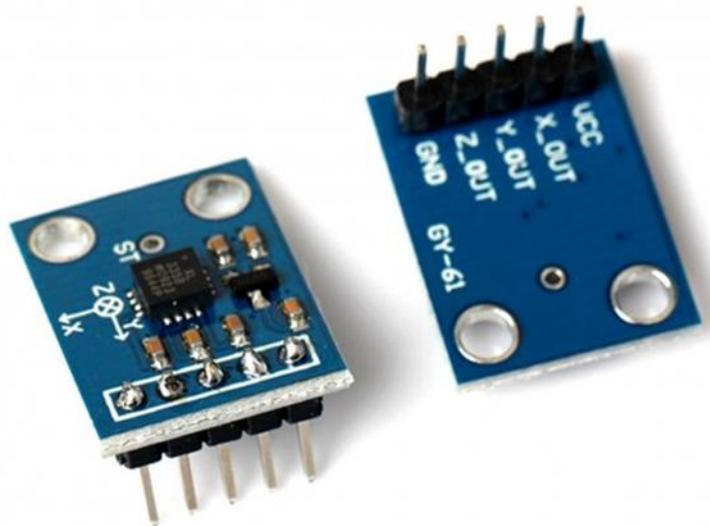


Figure 2.10: ADXL335 Accelerometer Sensor

2.3.2.2 Gyroscope

A gyroscope measures the angular rate of rotation along 3 axes of the sensor as shown in Figure 2.11. To compute the orientation of the robot, the angular rate produced by the sensor is multiplied with the time interval between each reading and added to the last angle compute. Compared with the accelerometer, gyroscope is less sensitive to mechanical noise as it only measures rotation. However, gyroscope has drift problem in which the angular rate does not come back to zero-rate when the rotation stops [16].

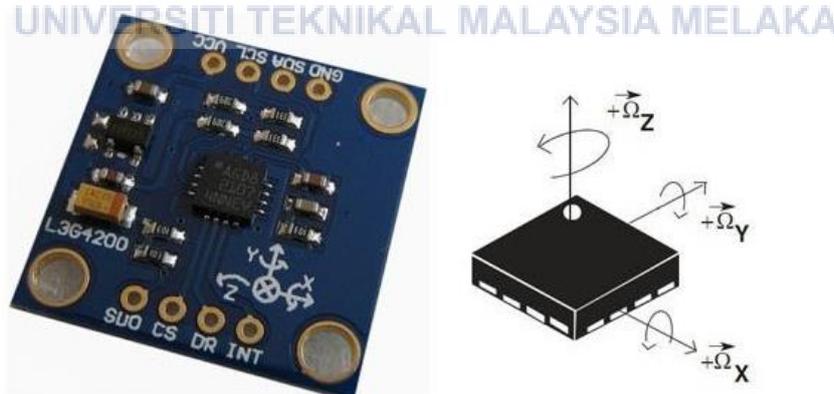


Figure 2.11: L3G4200D Gyroscope Sensor

2.3.2.3 Magnetometer

A magnetometer is essentially a compass. It measures the magnetic field of its surrounding and the earth magnetic field in order to compute the true north heading[15], [21]. By having the true north heading as a reference, the robots yaw value can be computed. Figure 2.12 shows an example of magnetometer sensor.



Figure 2.12: HMC5883L Magnetometer Sensor

2.3.3 Sensors for Robot Speed

For an autonomous mobile robot, controlling its speed is very important in order for it to maintain traction on irregular surface by avoiding tire slippage. The sensors that are suitable for this purpose is Accelerometer sensor and Optical Encoder for motor,

2.3.3.1 Accelerometer

As mention before, an accelerometer measures the static and dynamic forces act upon it. By measuring the acceleration of the mobile robot and multiply it with the interval time between each reading, the velocity of the robot can be calculated[16].

2.3.3.2 Optical Encoder

An optical encoder measures the speed of the mobile robot by detecting the position change of a patterned encoder disc as light passes through it as shown in Figure 2.13. As the disc rotates, the position change represents a square wave where after several specific wavelengths, 1 rotation of the disc is reached. Using simple equation, the velocity of the motor can be computed in the unit of revolution per minute (RPM)[22].

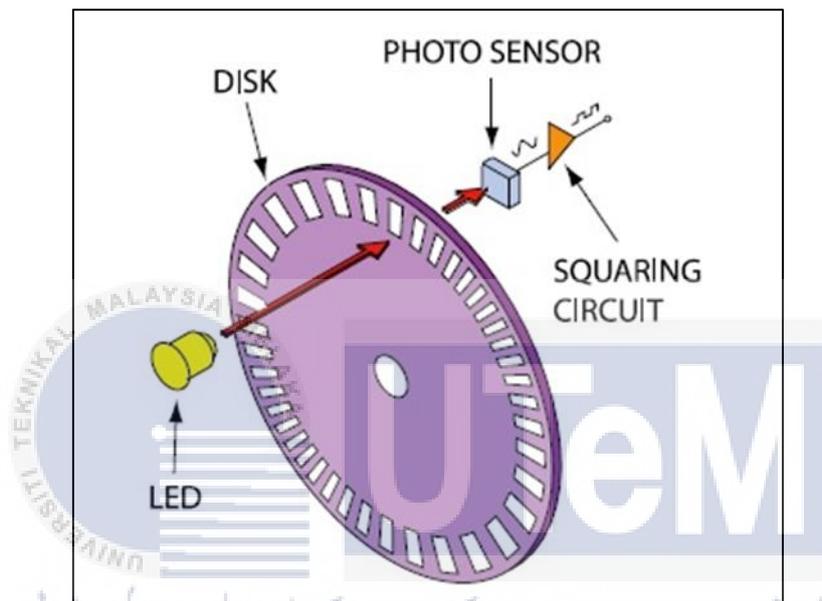


Figure 2.13: Example of Optical Encoder Working Principle

2.4 Type of Common Control System

A control system is made of subsystem and processes that is assembled with the purpose of producing a desired output and performance given specific input. There are two most commonly used control system which is the Proportional-Integral-Derivative (PID) Controller and Fuzzy Logic Controller.

2.4.1 Proportional-Integral-Derivative (PID) Controller

A PID controller is a control algorithm that uses three gain coefficients which are proportional, integral, and derivative, to tune an output signal in reference with the input sensor signal as shown in Figure 2.14. By varying the coefficients, the PID controller can produce a desired actuator output signal within a closed loop feedback mechanism. However, this method is only available for a Single Input Single Output (SISO) system[15], [23], [24].

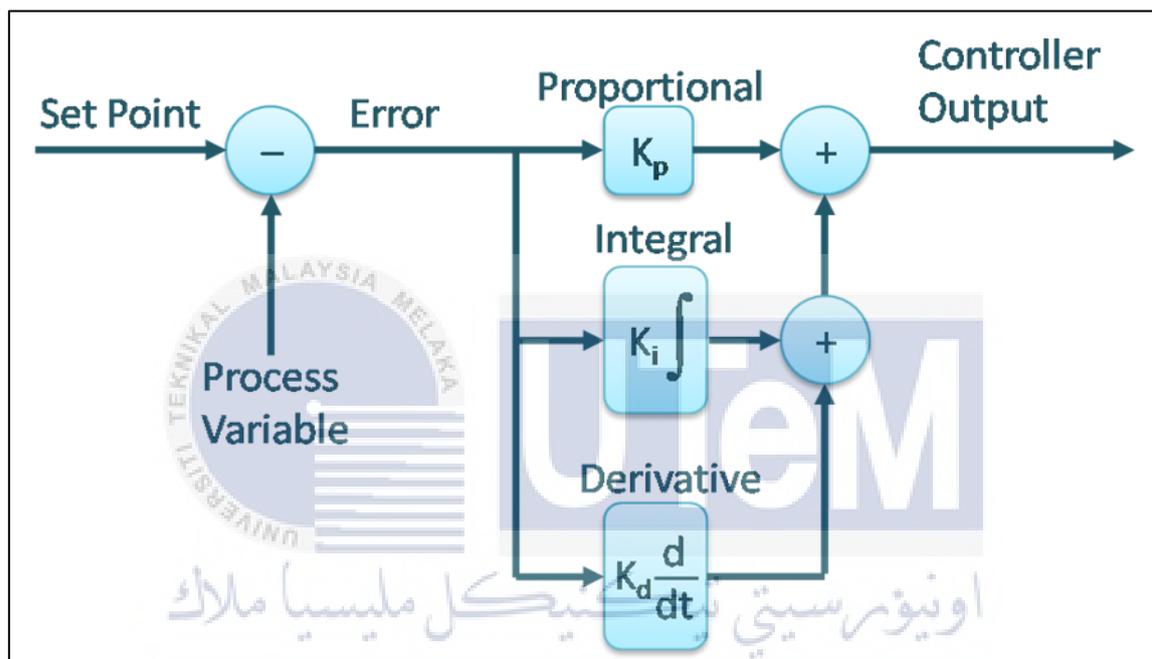


Figure 2.14: Typical Block Diagram of PID Controller

2.4.2 Fuzzy Logic Controller (FLC)

A fuzzy logic controller is a decision-making method that closely resembles the human decision-making skills. It deals with vague and imprecise information by looking at the degree of truth rather than the usual true or false Boolean logic. FLC is commonly used to handle Multiple Input Single Output (MISO) system only. For a Multiple Input Multiple

Output (MISO) system, a Mixed Fuzzy Controller is used where multiple FLC is integrated to control the MIMO system[25]–[27].

2.5 Overall Summary

Aspects	Sub-Aspects	Chosen	Reasons
Type of Mobile Robots	Wheeled Mobile robot	Differential Drive	Easy to control the movement of motor
Type of Sensor	For Robot Orientation	Accelerometer & Gyroscope	Able to use IMU sensor system
	For Obstacle Detection	Ultrasonic Sensor	Long ranged detection
Type of common Control System		Fuzzy Logic Control	Able to use for Multiple Input Single Output (MISO) control system

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter describes the method and techniques used to design the experiments for this project. A good motion control for a mobile robot is designed by collecting vital information about the robot's state and its surrounding and putting it into the control algorithm so that the robot can make the optimum decision. The area of study interested in mobile robot on uneven terrains are the robot's orientation, which correspond directly with the surface orientation, and unexpected obstacles. This information allows the robot to adapt to the environment and navigate around without any difficulty.

In this project, experimental method is used to collect data on each of the area of study. Several different methods are used to obtain each information and an experiment is design based on those methods to collect data of each information. Before any experiment is done, a mobile robot is designed to help conduct the experiments. Table 3.1 below shows the mapping of the designing task and experiments task to the project's problem statements and objectives.

Table 3.1 Mapping of Tasks to Project's Problem Statements and Objectives.

Task	Problem Statements			Objectives		
	#					
Designing Mobile Robot	#					
Experiment 1: Detecting Surface Orientation		#		#		
Experiment 2: Obstacle Detection		#			#	
Experiment 3: Designing Motion Control for Uneven Terrain			#			#

3.2 Project Overview

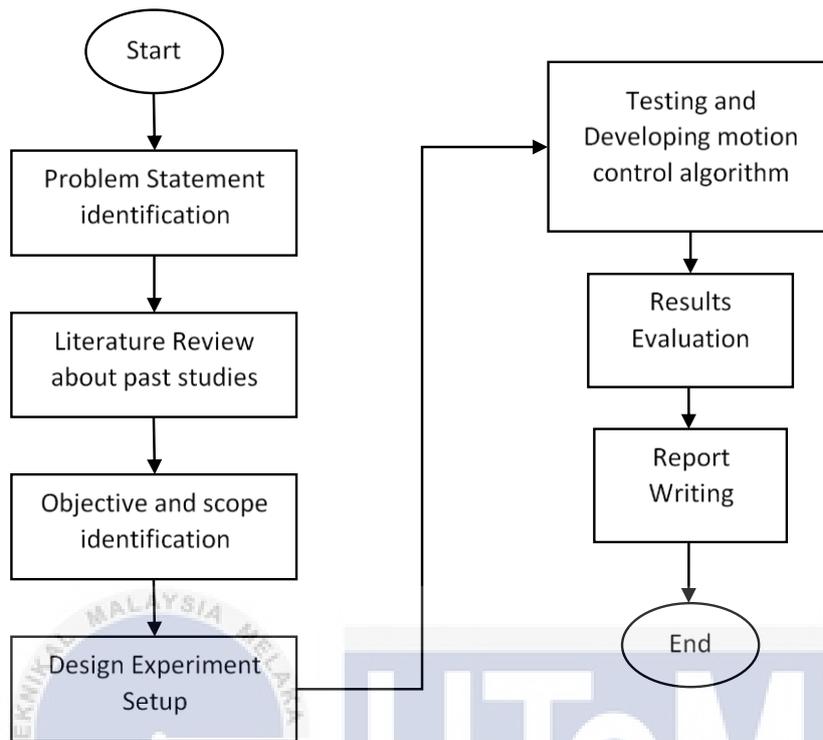


Figure 3.1: Project Development Process (Flow Chart).

3.3 Mobile Robot Design

The design of the mobile robot has a great effect onto its performance. By designing the mobile robot to overcome a problem, it can save a lot of time from having to tune its program until the problem resolves. The design of the mobile robot uses 4 wheels as the mode of locomotion and the concept of differential drive configuration at the front wheel. Since the main problem of this project is tires slip, an absorber is added to each wheel to compensate for the tire slip individually. An ultrasonic sensor is placed in the front of the robot to detect incoming obstacles from the front. **Error! Reference source not found.**, Figure 3.3, and Figure 3.4 below shows the design of the mobile robot used.

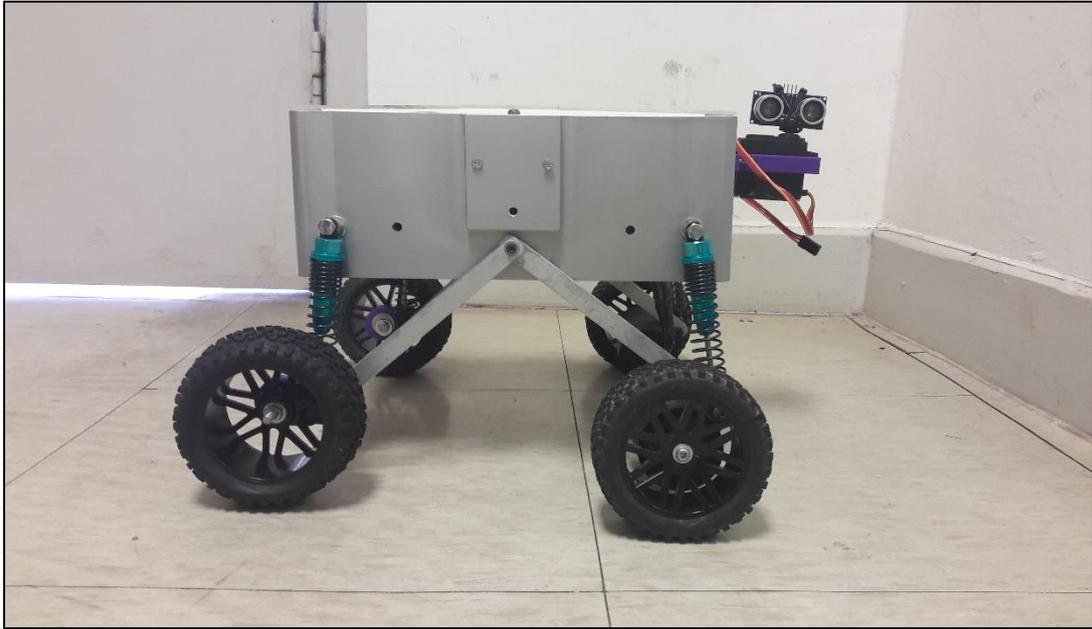


Figure 3.2: Right Side View of Mobile Robot



Figure 3.3: Right Side View of Mobile Robot

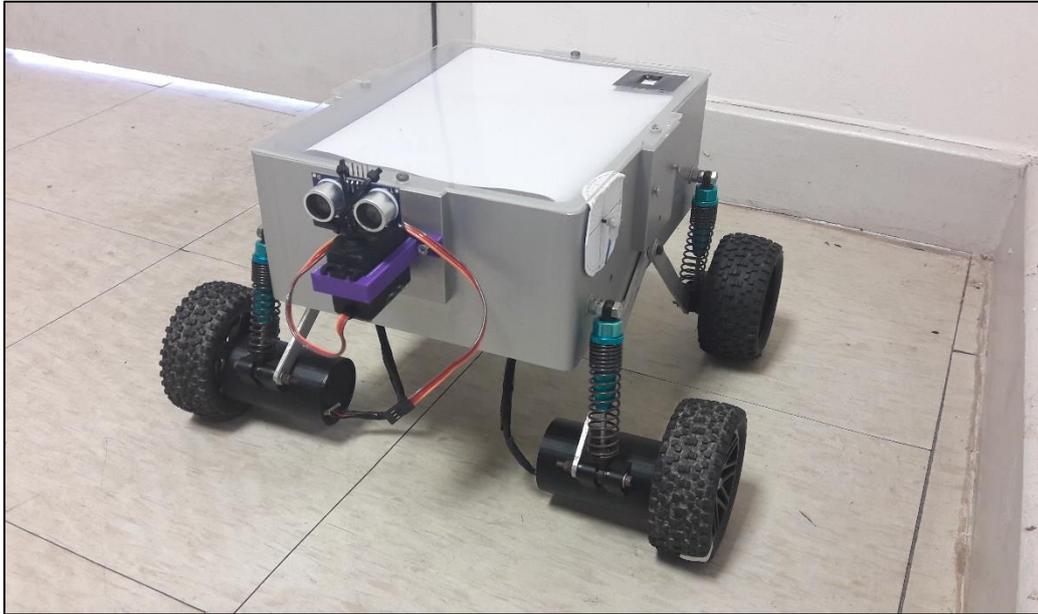


Figure 3.4: Orthographic View of Mobile Robot

3.4 Experiment 1: Detecting Surface Orientation

This experiment is designed to collect data on the characteristic of an irregular surface based on the changes of the robot's orientation in terms of pitch and roll. In most cases, the mobile robot's orientation corresponds directly to the surface orientation of a terrain, except when tire slip occurred. This information helps the robot to make decision based on the characteristic of the terrain.

The experiment is divided into three parts. The first part is designing the Inertial Measurement Unit (IMU) sensor algorithm by obtaining the raw value of the accelerometer and gyroscope sensor reading and pass it through a complementary filter to get an accurate estimation of the robot's orientation. The second part is doing an open loop test on the IMU sensor by testing the sensor on a rig that changes its orientation around 1 axis. Since the changes on the terrain's surface mostly affects the pitch of the mobile robot, only the angle of orientation around Y-axis is taken into consideration. The third part is the integration of the IMU sensor and motor speed control of the robot. A FLC system is designed to control the motor speed of the robot based on its pitch orientation. The robot is then tested on several different type of terrain to evaluate the behavior of the FLC system.

For this experiment, a GY-85 9DOF IMU Sensor is used to get values from an accelerometer and gyroscope sensor, and the axis frame of the mobile robot is shown in Figure 3.5. The pitch reading of the robot is defined as angle of rotation around Y-axis while roll reading is defined as the angle of rotation around X-axis.

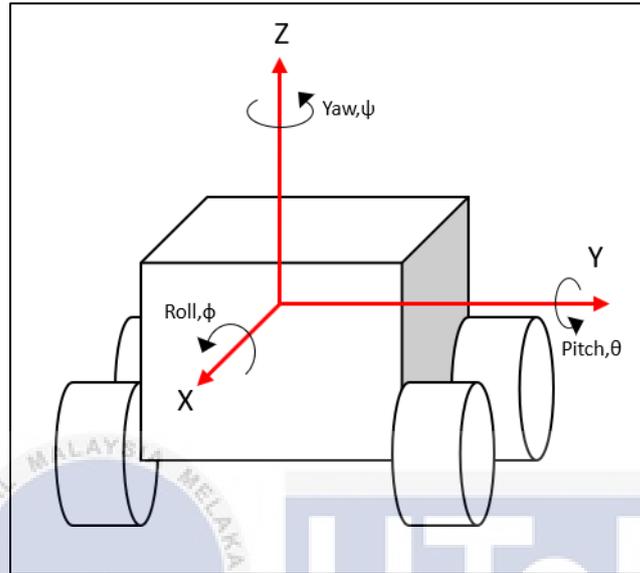


Figure 3.5: Axis Frame of The Mobile Robot

3.4.1 Designing Inertial Measurement Unit (IMU) Sensor's Algorithm

Since the accelerometer measures static forces that act upon each axis, the output of the accelerometer will show the value of gravity force acting on each axis in the unit LSB. To find the orientation of each axis, the raw output value of each axis from the accelerometer sensor is first divided with the scale factor as shown in (3.1), (3.2), and (3.3), to get the value of the gravity force in the unit 'g'.

$$F_x = \frac{AxRaw}{256} \quad (3.1)$$

$$F_y = \frac{AyRaw}{256} \quad (3.2)$$

$$F_z = \frac{AzRaw}{256} \quad (3.3)$$

At any orientation, the force of gravity, F_g , can be assumed to be at any direction in between all three axes as shown in Figure 3.6. By using 3D Pythagorean Theorem, the resultant force of gravity, F_g , is calculated using the equation (3.4).

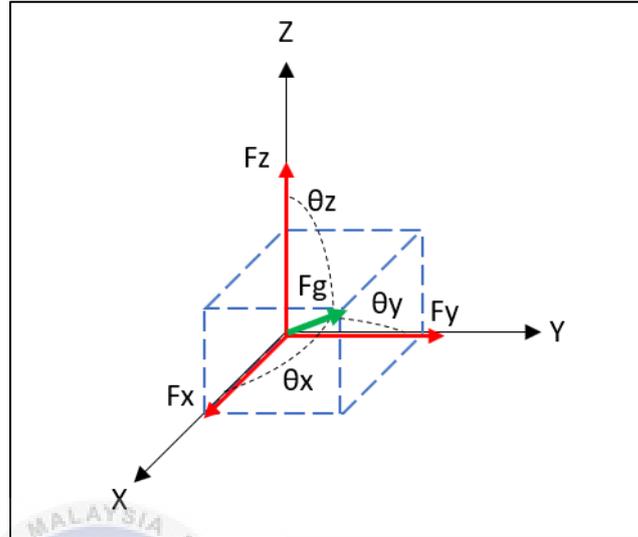


Figure 3.6: Force vectors acting on the sensor in 3D spaces

$$F_g = \sqrt{F_x^2 + F_y^2 + F_z^2} \quad (3.4)$$

Then, the angle value of pitch and roll orientation is identified as the angle between the X-axis and Y-Axis, and the force of gravity respectively. This angle value is obtained by feeding the value of gravity force on each axis and the resultant gravity into the trigonometric equation in (3.5) and (3.6) below.

$$\theta_x = \cos^{-1} \left(\frac{F_x}{F_g} \right) \quad (3.5)$$

$$\theta_y = \cos^{-1} \left(\frac{F_y}{F_g} \right) \quad (3.6)$$

Gyroscope sensor measures the rate of rotation around each axis. To get the rate of rotation, the raw output value is divided with the scale factor to convert it into the unit ‘°/s’, and multiplied with the time duration from the last reading as shown in (3.7) and (3.8), to get the value of rate of rotation in ‘°’.

$$Rx = \left(\frac{GxRaw}{14.375} \right) \times dt \quad (3.7)$$

$$Ry = \left(\frac{GyRaw}{14.375} \right) \times dt \quad (3.8)$$

$$dt = T_n - T_{n-1} \quad (3.9)$$

The value of the axis orientation from the accelerometer is sent through a low pass filter to remove any noise signal cause by sudden forces while the rate of rotation from the gyroscope is multiplied with the duration time since the last reading being sent through a high pass filter to remove the drift effect cause during resting. This can be seen in the typical block diagram of the complementary filter shown in Figure 3.5.

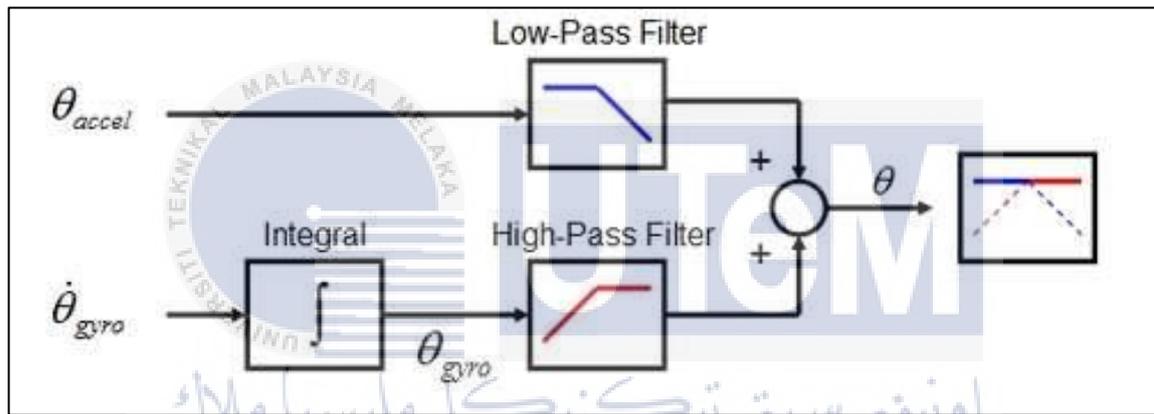


Figure 3.7: Typical Block Diagram of Complementary Filter

The complementary filter technique is used to obtain the most accurate estimation of the robot's orientation by adding 20% of the filtered value of axis orientation and 80% of the summation of filtered value of rate of rotation and the previous angle together as shown in (3.10) and (3.11).

$$X = 0.8(X_{n-1} + Ry) + 0.2(Ax) \quad (3.10)$$

$$Y = 0.8(Y_{n-1} + Rx) + 0.2(Ay) \quad (3.11)$$

3.4.2 Open Loop Test of IMU Sensor

Open loop test is a basic test that can be used to observe the different characteristic of several type of surface. By measuring the robot's orientation, it is also possible to measure the surface orientation of a terrain. This is due to how the robot's orientation corresponds directly to the surface orientation when it is in direct contact with the surface. In this test, the pitch orientation of the robot is chosen as the characteristic to be observed since it corresponds with the incline or decline of an uneven terrain as shown in Figure 3.8. This characteristic is the most suitable option since the mobile robot will mostly be moving forward in this project.

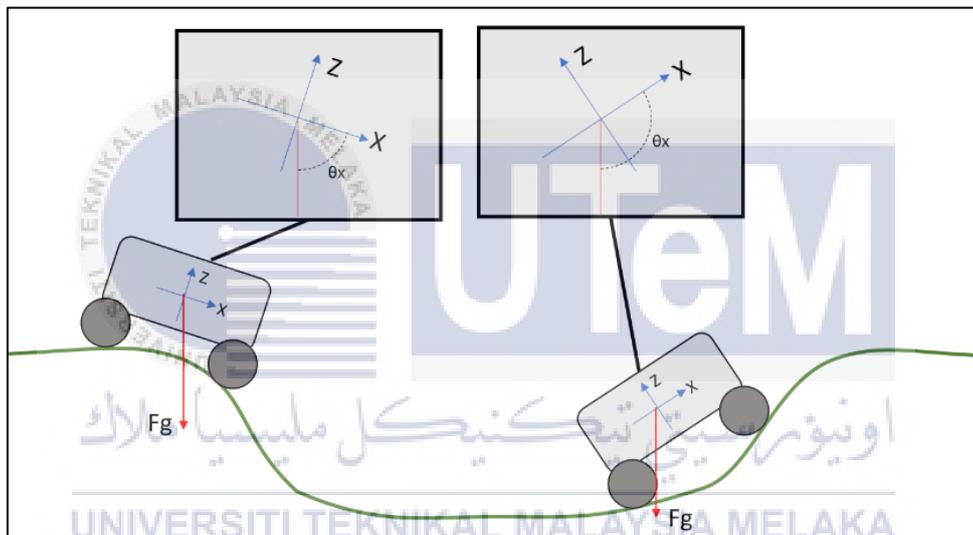


Figure 3.8: Example of Pitch Orientation of Robot Correspond Directly to Surface Orientation

Firstly, the IMU sensor is placed on a platform of a rotating rig as shown in Figure 3.9 below. The platform is then rotated to 0 to 180 at an interval of 10 to observe the accuracy of the of the IMU sensor. The pitch orientation measurement from the accelerometer, gyroscope sensor, and complementary filter are recorded into the Figure 3.9: Open Loop Test setup Using a Rotating Platform Rig.

Table 3.2 below. A graph is then generated to show the accuracy of the sensor by comparing each measurement mentioned above.

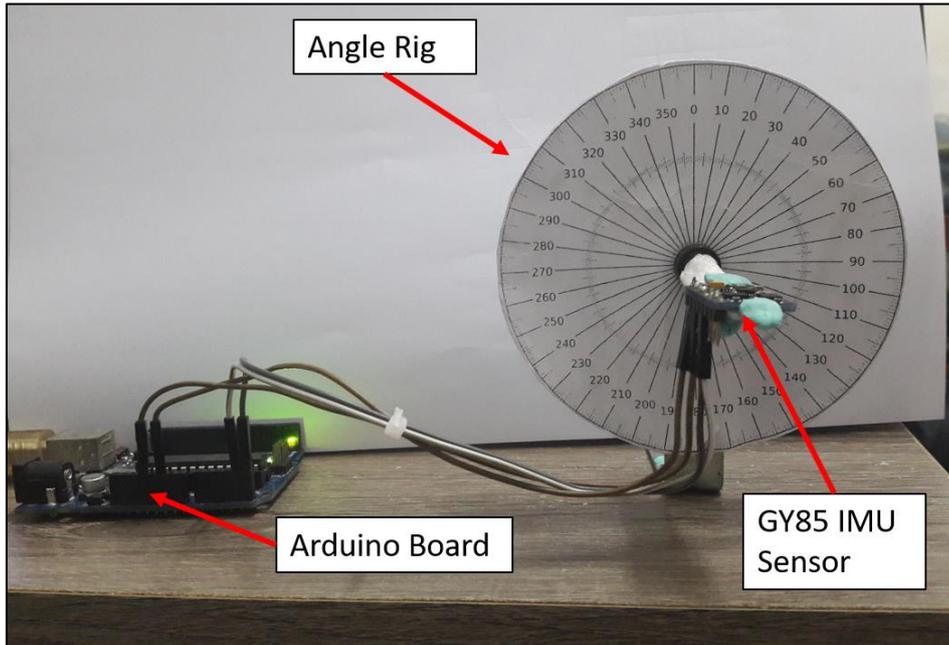


Figure 3.9: Open Loop Test setup Using a Rotating Platform Rig.

Table 3.2: Pitch Orientation Measurements for Open Loop Test on a Rotating Platform Rig

Angle (°)	Accelerometer (°)	Gyroscope (°)	Complementary Filter (°)
0			
10			
...			
170			
180			

3.4.3 Intergration of IMU sensor and Motor Control Using Fuzzy Logic Control System.

A FLC system is made by integrating the IMU sensor with a simple motor speed control system in order to control the speed of the robot when navigating on uneven terrain. Since the robot orientation correspond directly with the ground surface, the pitch of the robot

is taken as an input of the fuzzy logic controller since as it helps with detecting an incline or decline on the ground surface while the output of the fuzzy logic is set as the speed of the motor. Using the software Matlab, the fuzzy logic controller is designed based on the block diagram shown in Figure 3.10 below.

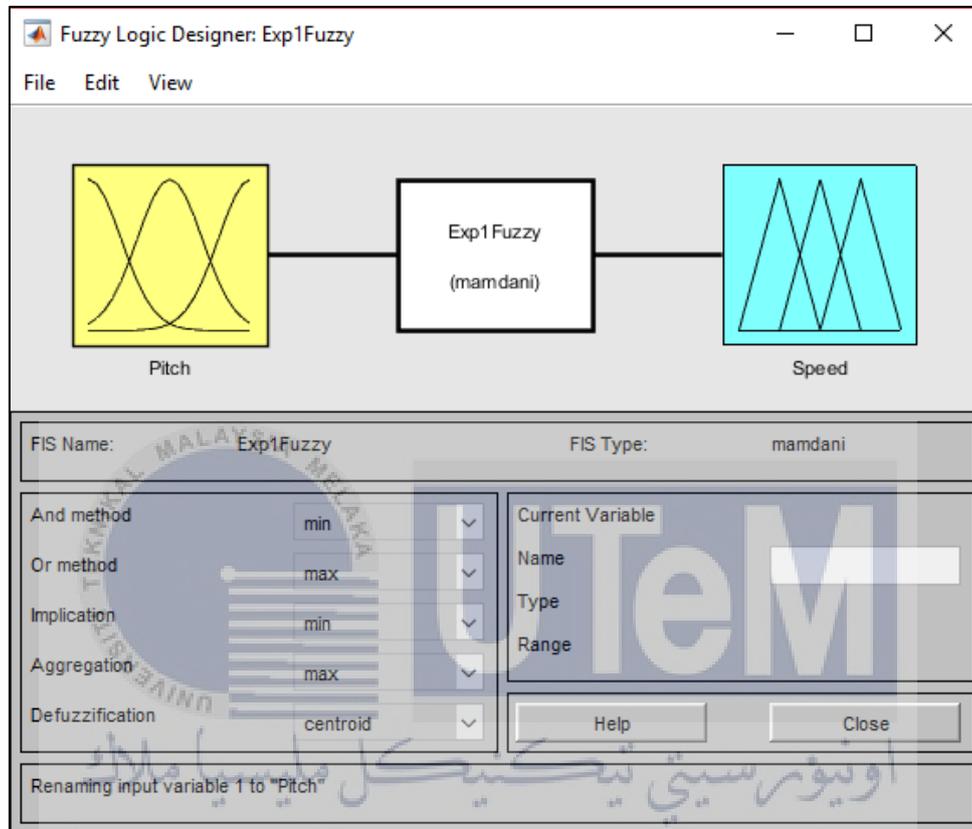


Figure 3.10: Block Diagram of Fuzzy Logic Controller for Motor Speed Control

The range of the pitch measurements is set from 0° to 180° with 80-90 as the mid-range. While the range of the speed of the motor which uses Pulse Width Modulation (PWM), is set between 20-300. With this, the membership function for input and output is designed as shown in Figure 3.11 and Figure 3.12 below.

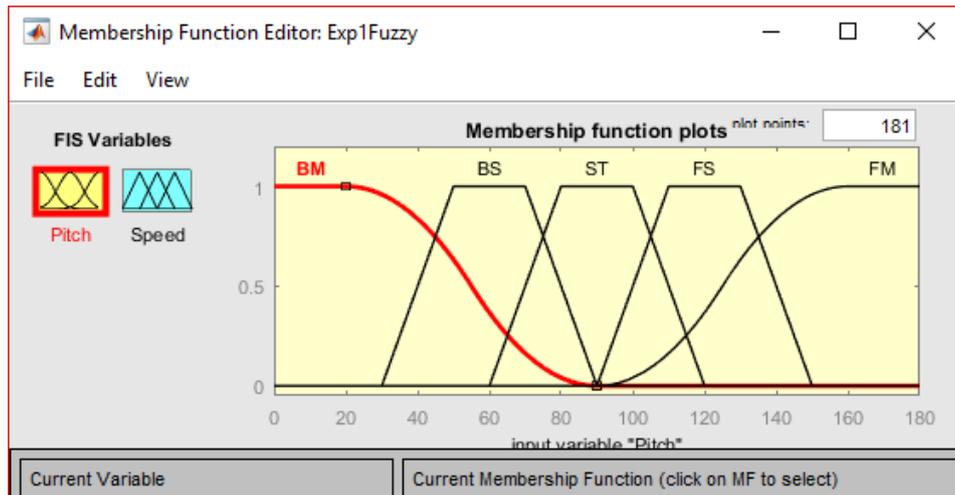


Figure 3.11: Membership Function for Input Variable

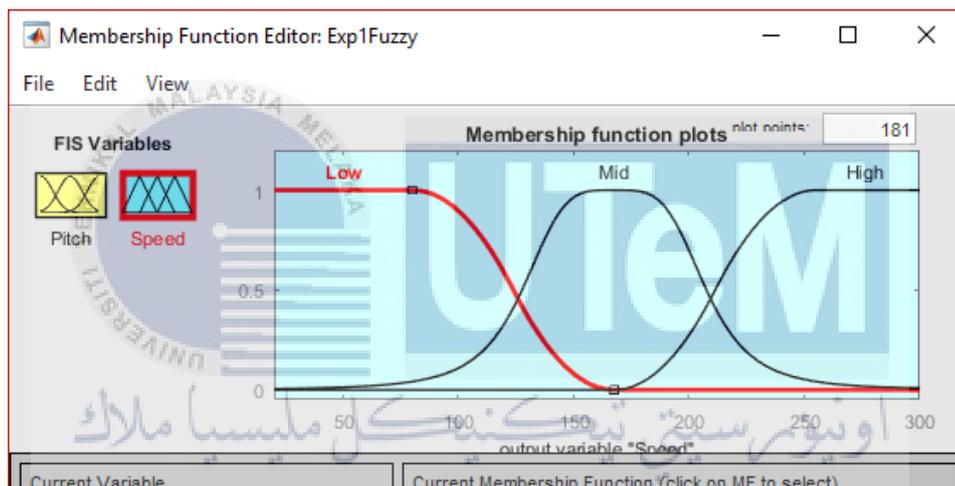


Figure 3.12: Membership Function for Output Variable

For this FLC system, five simple rules are used to relate the input membership function to its outputs as shown in the Table 3.3.

Table 3.3: Fuzzy Rules for Motor Speed Controller

Rule	Pitch	Motor Speed
1	Straight (ST)	HIGH
2	Forward Small (FS)	MID
3	Forward Medium (FM)	LOW
4	Backward Small (BS)	MID
5	Backward Medium (BM)	LOW

By following these rules, a relationship between the input and output variables of the FLC system can be seen in the generated graph shown in Figure 3.13. It also shows the expected behavior of the controller when navigating on uneven surface.

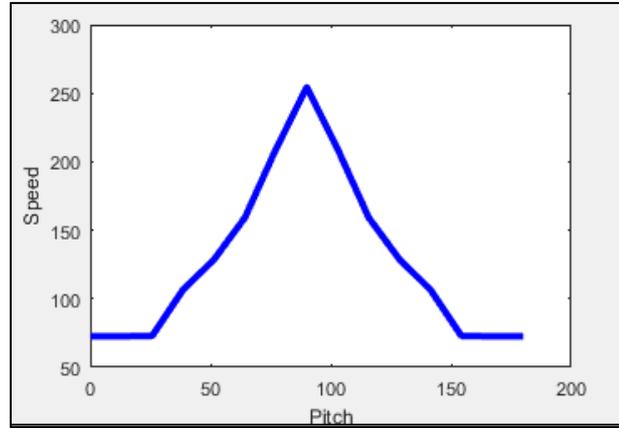


Figure 3.13: Expected Behaviour of the FLC System

The fuzzy logic controller developed is then used to observe the behavior of the mobile robot on several different terrain such as tiles, asphalt, rocks, and hill. The robot's pitch, speed, and time taken are recorded in Table 3.4 as the robot move straight for 100 cm. A graph is then generated to represent the behavior of the mobile robot on different terrain.

Table 3.4: Behavior of Mobile Robot on Uneven Terrain

No of reading, n	Pitch of robot (°)	Speed of motor (PWM)	Time taken (s)
1			
2			
.....			
n-1			
n			

3.5 Experiment 2: Obstacle Detection

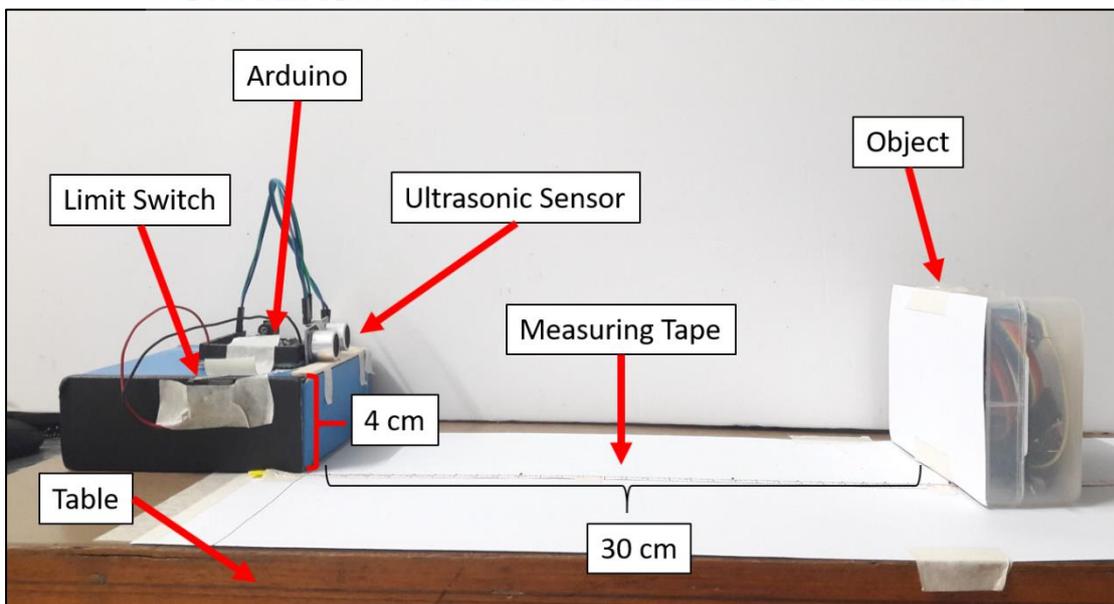
This experiment is designed to develop a method which allows the mobile robot to detect the position of an obstacle in front of it, control its approaching speed, and find an alternative path to avoid the obstacle. A 180 degrees Light Detection and Ranging (LIDAR) sensor is mounted at the front of the mobile robot and an obstacle is put in front of the robot

on different direction. But instead of light detection, an ultrasonic is used to measure the distance of an obstacle in every angle.

This experiment is divided to three parts. The first part is an open loop test on the ultrasonic sensor to observe its accuracy on measuring the distance of the obstacle in front of it. The second part is a close loop test in which the distance of the obstacle measured is used to control the speed of the robot by using a FLC system. The third part is an integration of the ultrasonic sensor with a servo motor to create a 180 degrees LIDAR sensor system which can detect the position of an obstacle and find the best alternative path to avoid the obstacle.

3.5.1 Open Loop Test of Ultrasonic Sensor

In this part, the open loop test is done by placing an ultrasonic sensor on a stand or box and an obstacle is placed in front of to measure its distance. There are two setups prepared for this part. For the first setup, the sensor is set up as shown in Figure 3.14. The obstacle's distance from the sensor is changed from 0 cm to 100 cm at an interval of 1 cm. This is done to observe the accuracy of the sensor to measure the distance of obstacles within 100 cm range from the robot. While for the second setup, the sensor is set up as shown in Figure 3.11. The obstacle's distance from the sensor is changed from 0 cm to 400 cm at an



interval of 50 cm. This is done to observe the accuracy of the sensor to detect obstacles that are further than 100 cm range from the robot.

Figure 3.14: Setup for 100 cm Open Loop Test

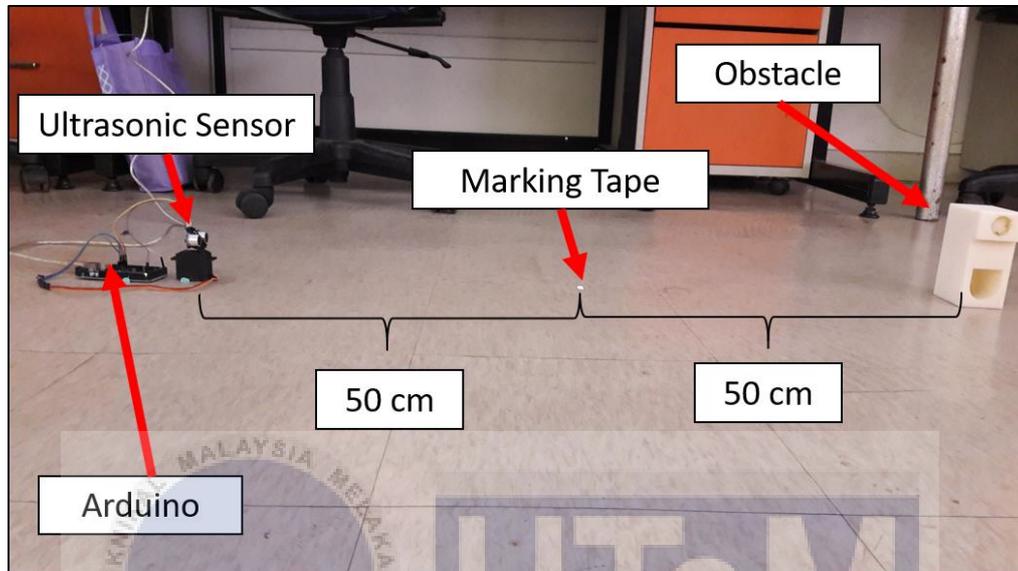


Figure 3.15: Setup for 400 cm Open Loop Test

Each measurement is taken three times in order to calculate the average reading and standard deviation using the equation () and (3.13) respectively. The distance based on the average reading is also calculated using the equation (3.15). For each actual distance, the theoretical echo duration is calculated based on the equation (3.14) as a reference for the echo duration readings. All measurements are recorded in Table shown below and the data is graphed to observe the accuracy of the sensor's measurements.

Table 3.5: Characteristic of the Ultrasonic sensor when measuring the distance of an obstacle

Actual Distance (cm)	Calculated Echo Duration (μs)	Echo Duration Reading (μs)					Calculated Distance (cm)
		1	2	3	Average	Standard Deviation	
0							
1							
...							
n-1							
n							

$$\text{Average Reading, } \bar{x} = \frac{\sum \text{Echo Duration Reading}}{3} \quad (3.12)$$

$$\text{Standard Deviation, } \sigma = \sqrt{\frac{\sum (\text{Echo Duration Reading} - \bar{x})^2}{3 - 1}} \quad (3.13)$$

$$\text{Calculated Echo Duration} = \frac{\text{Actual Distance}}{0.0343} * 2 \quad (3.14)$$

$$\text{Calculated Distance} = \frac{\bar{x} \times 0.0343}{2} \quad (3.15)$$

3.5.2 Closed Loop Test of Ultrasonic Sensor and Motor Speed Control Using Fuzzy Logic Controller

In this part, close loop test done by using a FLC system to control the speed of the robot according to the distance of the obstacle measured from the robot. Two motors and a motor controller are added to the open loop setup to form a close loop setup system as shown in Figure 3.16

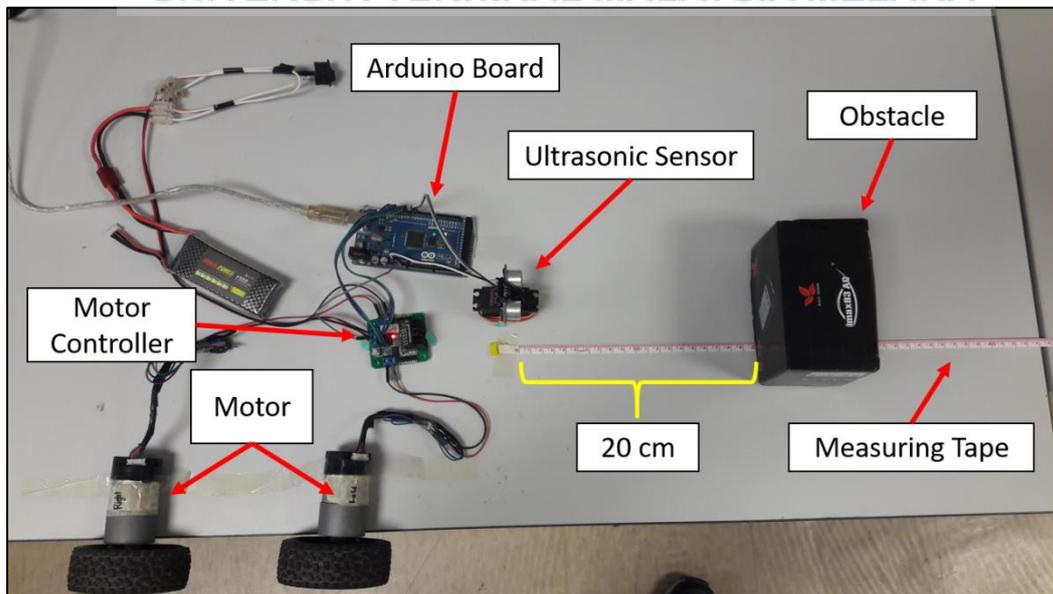


Figure 3.16: Close Loop Test Setup for Ultrasonic Sensor and Motor Control

Using the Matlab Software, the fuzzy logic controller is designed according to the block diagram shown in Figure 3.17. The distance of the obstacle is set as the input of the system while the speed of the motor is set as the output of the system.

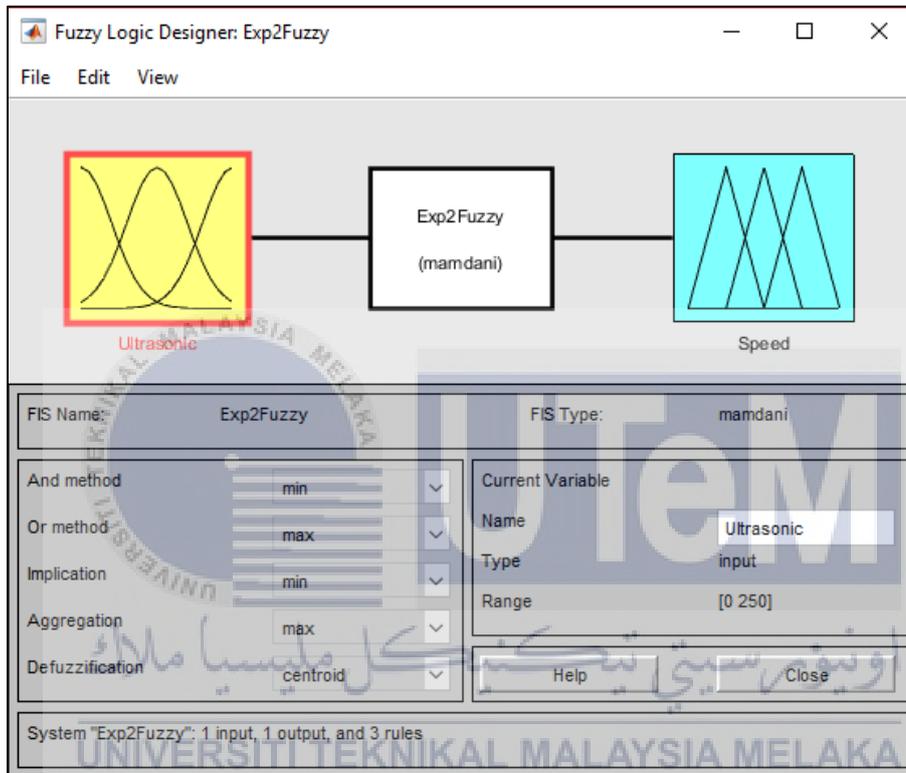


Figure 3.17: FLC Block Diagram for Motor Control using Ultrasonic

The range of the measured distance is set from 0 cm to 250 cm while the range for motor speed is set from 20 to 300 as shown in Figure 3.18 and Figure 3.19. The FLC system utilizes only three rules as shown in Figure 3.19: Membership Function of the Motor Output Speed

Table 3.6.

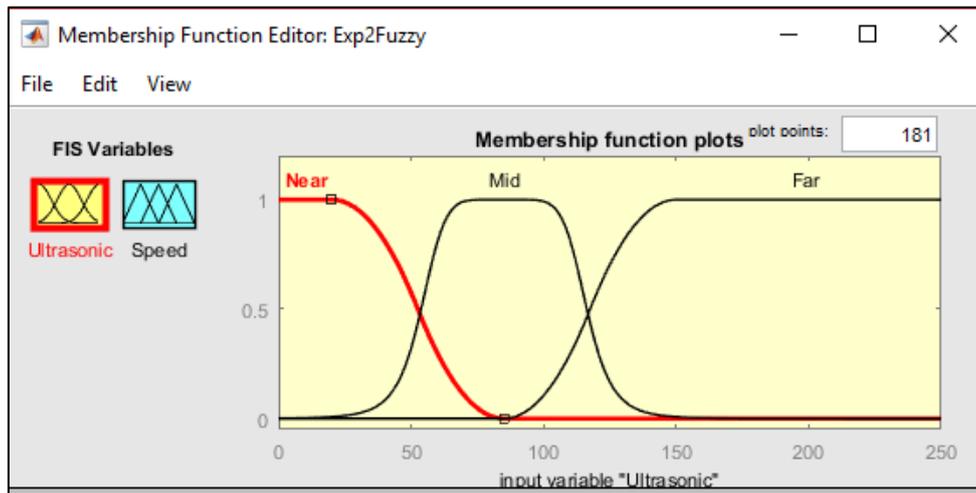


Figure 3.18: Membership Function of the Ultrasonic reading

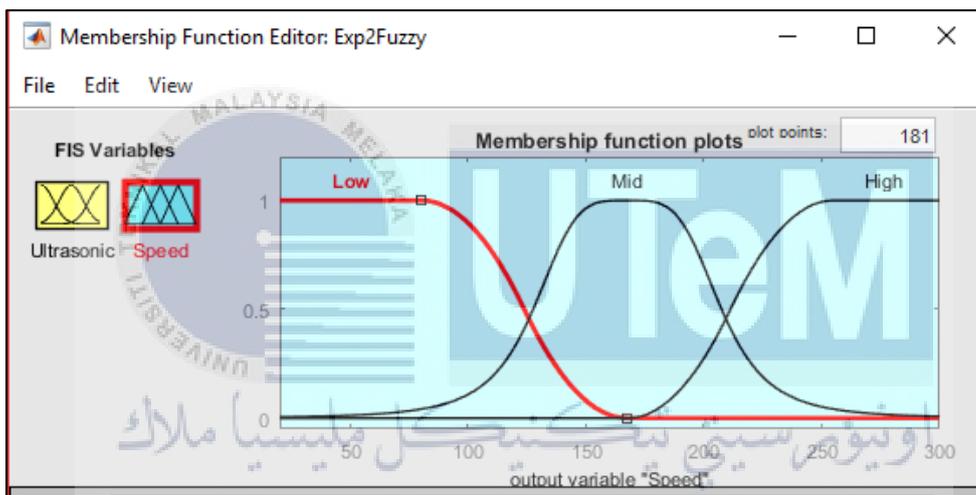


Figure 3.19: Membership Function of the Motor Output Speed

Table 3.6: Fuzzy Rule of the FLC system

Rule	Distance	Motor Speed
1	Near	LOW
2	Mid	MID
3	Far	HIGH

Following these three rules, a relationship between the input and output variables of the Fuzzy Logic Controller is graphed as shown in Figure 3.20 which shows the expected behavior of the controller when detecting an obstacle.

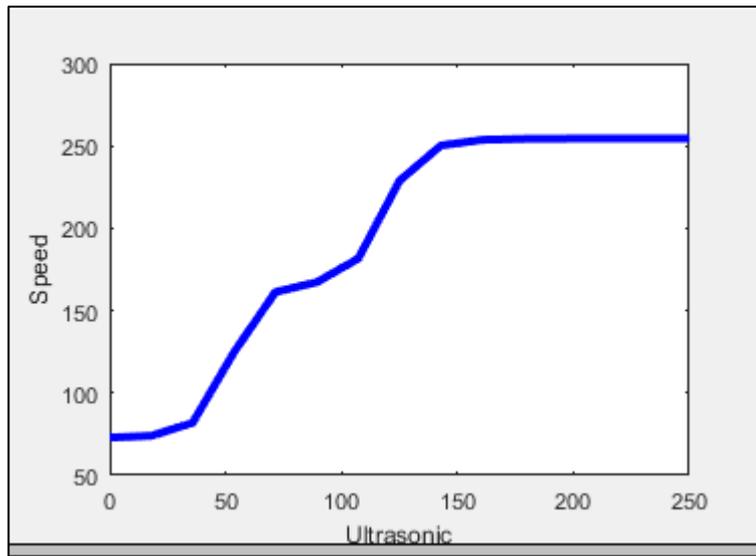


Figure 3.20: Expected Behaviour of Fuzzy logic controller

The behavior of the developed fuzzy logic controller is observed as it detects obstacle at a random distance from the ultrasonic sensor. The data is then recorded in the Table 3.7 below and graphed to represent the behavior of the controller.

Table 3.7: Characteristic Of Obstacle Distance And Motor Speed

No	Distance (cm)	Motor Speed (PWM)
1		
2		
...		
n-1		
n		

3.5.3 Heading Selection

In this part, a 180° LIDAR system is created by attaching an ultrasonic sensor to a servo motor. The LIDAR system is then put on a large scale protractor with a radius of 30 cm as shown in Figure 3.21. The large protractor helps with comparing the direction of the obstacle to the measurements reading.

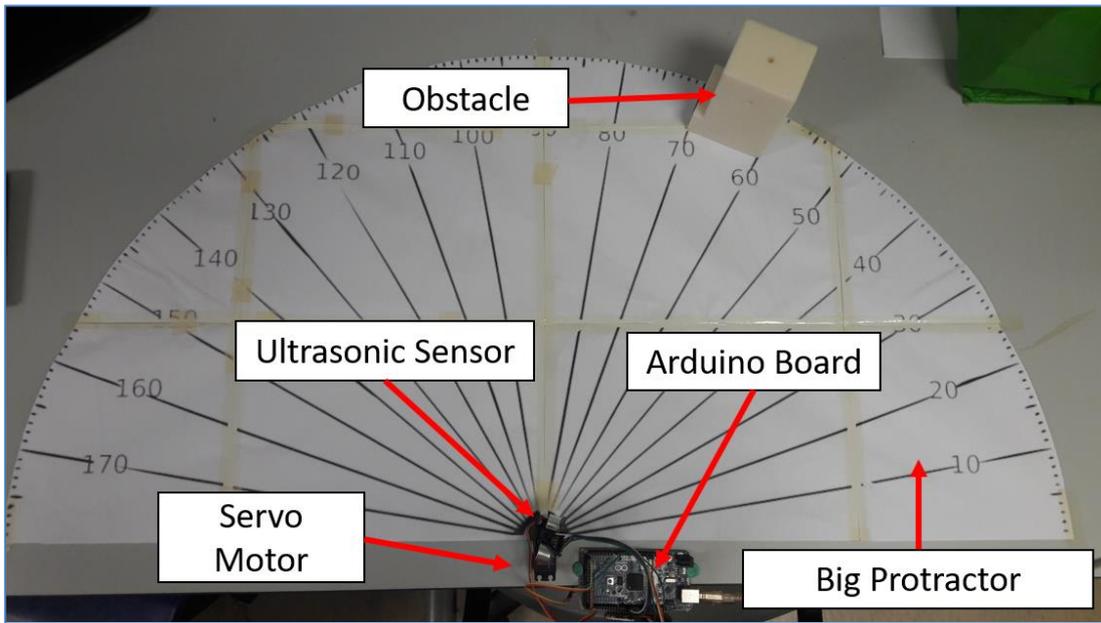


Figure 3.21: Experiment Setup for Heading Selection

The LIDAR system first scan the front area at an interval of 10° for 180° range. The top three furthest distance measured is taken as possible alternative routes. For each possible route, an average reading for the 20° around the possible direction is calculated. However, a possible direction is neglected if there exist a distance below 60 cm. The highest average reading of 20° around each possible route with no distance below 60 cm is chosen as the best alternative route. All readings are recorded in the table

Table 3.8: Distance Reading for a 180° scan

Heading Direction ($^\circ$)	Distance (cm)
0	
10	
...	
180	

Table 3.9: Evaluation of Possible Alternative Route Heading

Possible Alternative Route Heading	Average Distance Reading	Safety Threshold State

3.6 Designing Motion Control for Uneven Terrain

In this experiment, both Fuzzy Logic controller from Experiment 1 and 2 are brought together and combine into one Fuzzy Logic Control System. By adding the 180 degrees LIDAR system from Experiment 2, it will then become the suitable Motion Control for Mobile Robot on Uneven Terrain. Figure 3.22 shows the overall flowchart of the Motion Control for Uneven Terrain.

The Figure 3.23 shows the block diagram used to design the Fuzzy Logic Control System for this experiment. The membership function used for this system is the same as in Experiment 1 and 2. The only difference is the rules used has increased due to combination of two system. The rules used is shown in Table 3.8 below.

Table 3.10: Fuzzy Rules for Experiment 3

		Pitch Orientation				
		BM	BS	ST	FS	FM
Obstacle Distance	Near	LOW	LOW	LOW	LOW	LOW
	Mid	LOW	MID	MID	MID	LOW
	Far	LOW	MID	HIGH	MID	LOW

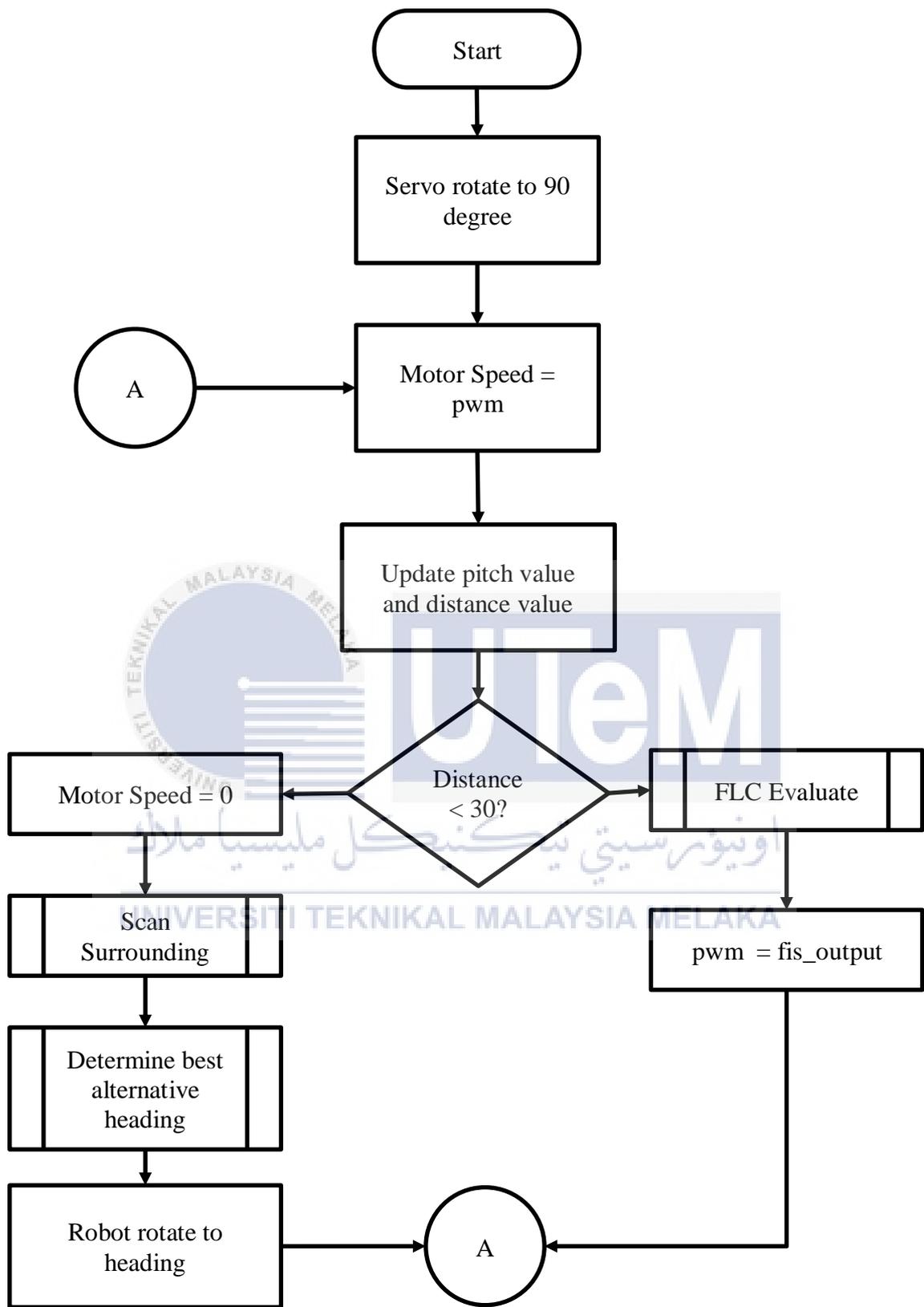


Figure 3.22: Typical Fuzzy Controller Architecture

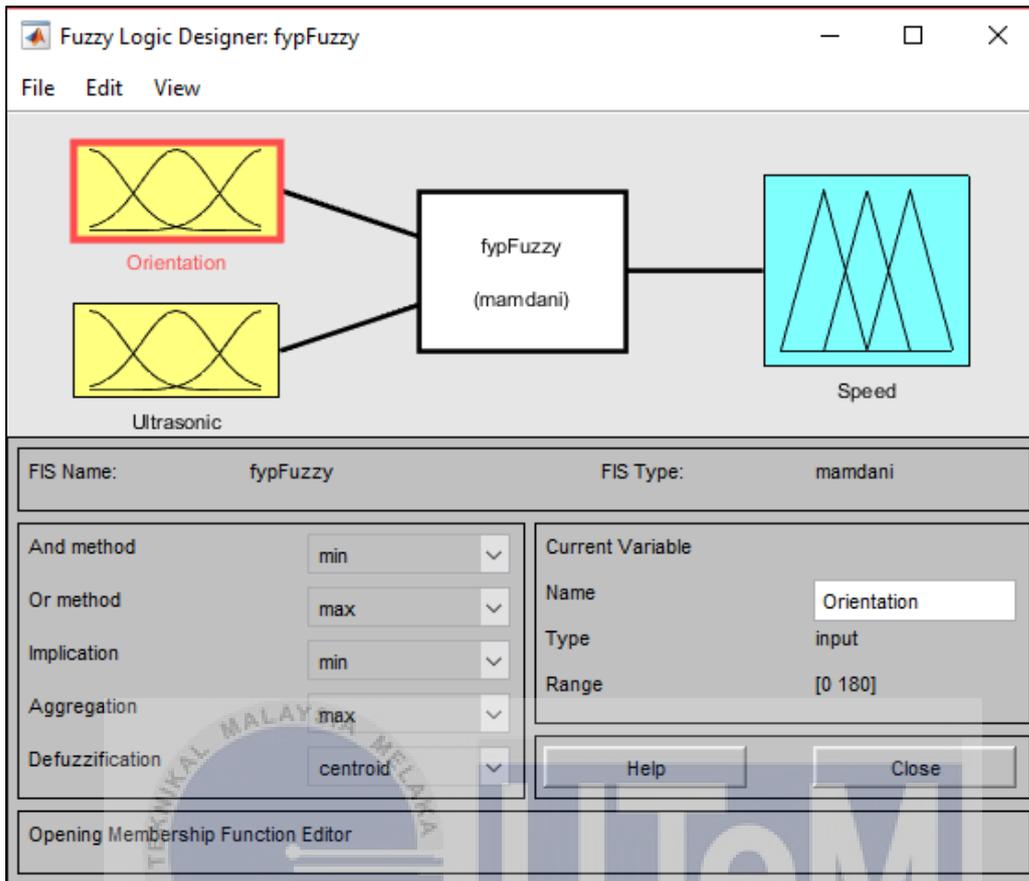


Figure 3.23: Block Diagram for FLC of Experiment 3

After combining the rules from Table 3.3 and Table 3.6, a relationship between the IMU sensor, Ultrasonic Sensor and the Motor Speed is graphed as shown in Figure 3.24 below.

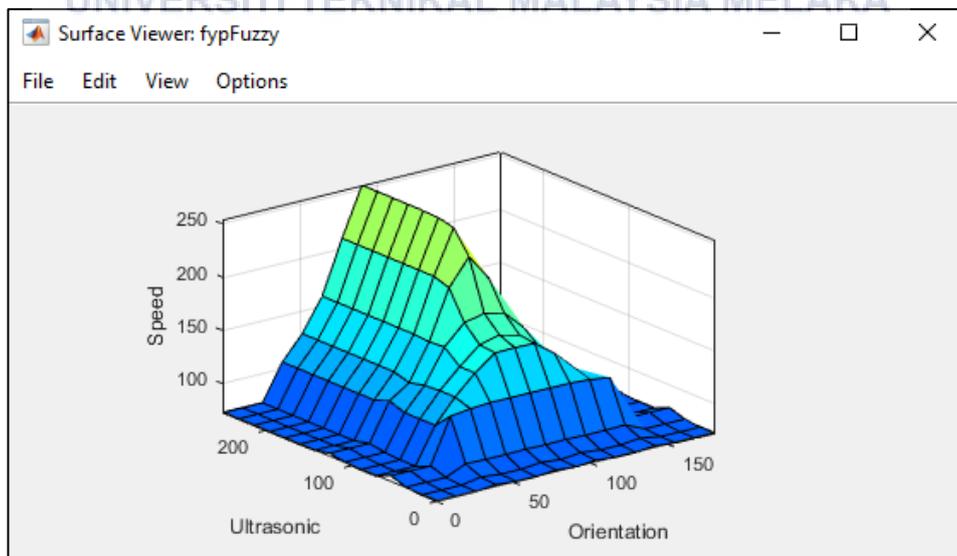


Figure 3.24: Surface Graph of Relationship between IMU sensor, Ultrasonic Sensor, and Motor Speed.

The FLC System is tested on the mobile robot by having the robot navigate on several surface shown in Figure 3.25. All measurements are recorded in the Table 3.11 below and graphed.



Figure 3.25: Type of Terrains Used for Experiment 3

Table 3.11: Close Loop Test for FLC System in Experiment 3

No	Pitch (°)	Distance (cm)	Motor Speed (PWM)	Time Taken (s)
1				
2				
...				
n-1				
n				

CHAPTER 4

RESULTS AND DISCUSSIONS

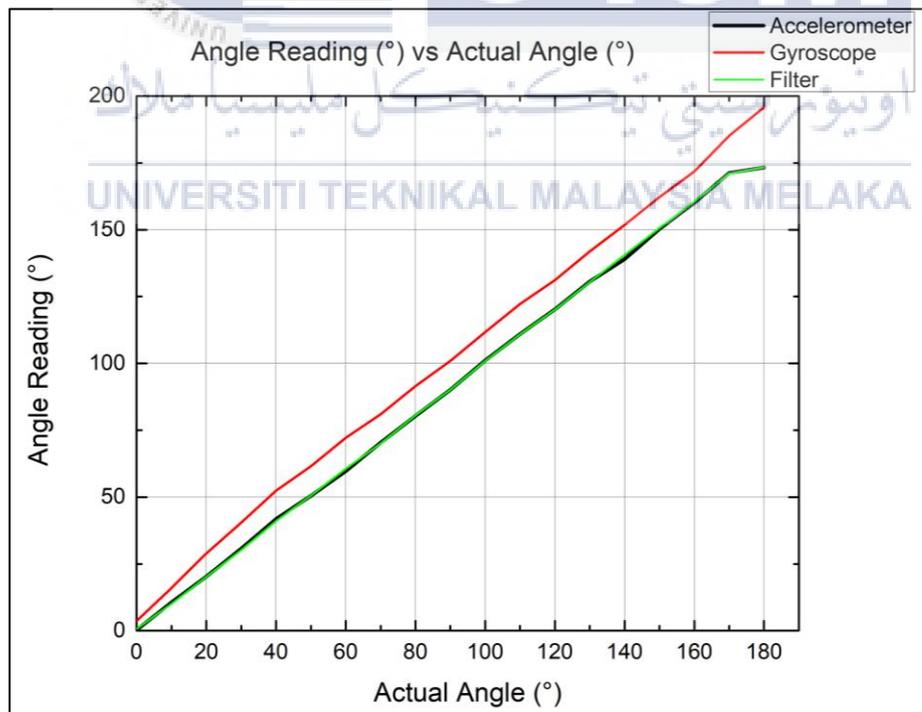
4.1 Experiment 1: Detecting Surface Orientation

In this section, the accuracy of the IMU sensor and the characteristic of the pitch reading on different terrain is analyzed and discussed.

4.1.1 Open Loop Test of IMU Sensor

In this experiment, the measurements of robot orientation in terms of pitch from accelerometer, gyroscope sensor, and complementary filter is graphed on Figure 4.1 below.

Figure 4.1: Graph of Angle Reading ($^{\circ}$) vs Actual Angle ($^{\circ}$)



From the graph shown in Figure 4.1, the readings from the accelerometer sensor shows some deviation from the actual angle at 40°, 60°, and 140°, while the readings from the gyroscope shows a drift effect occurring causing it to have a reading with 10 degree more than the actual angle set. However, the angle reading from the complementary filter accurately follows the actual angle set on the rig.

This shows how the complementary filter was able to remove the noise and drift effect and produce an accurate reading of the angle set on the rig despite some error from the sensor and drift effect from the gyroscope sensor.

4.1.2 Intergration of IMU sensor and Motor Control Using Fuzzy Logic Control System.

In this experiment, the characteristic graph of the pitch orientation and motor's speed over time on several type of terrain are shown in Figure 4.2, Figure 4.3, Figure 4.4, and Figure 4.5.

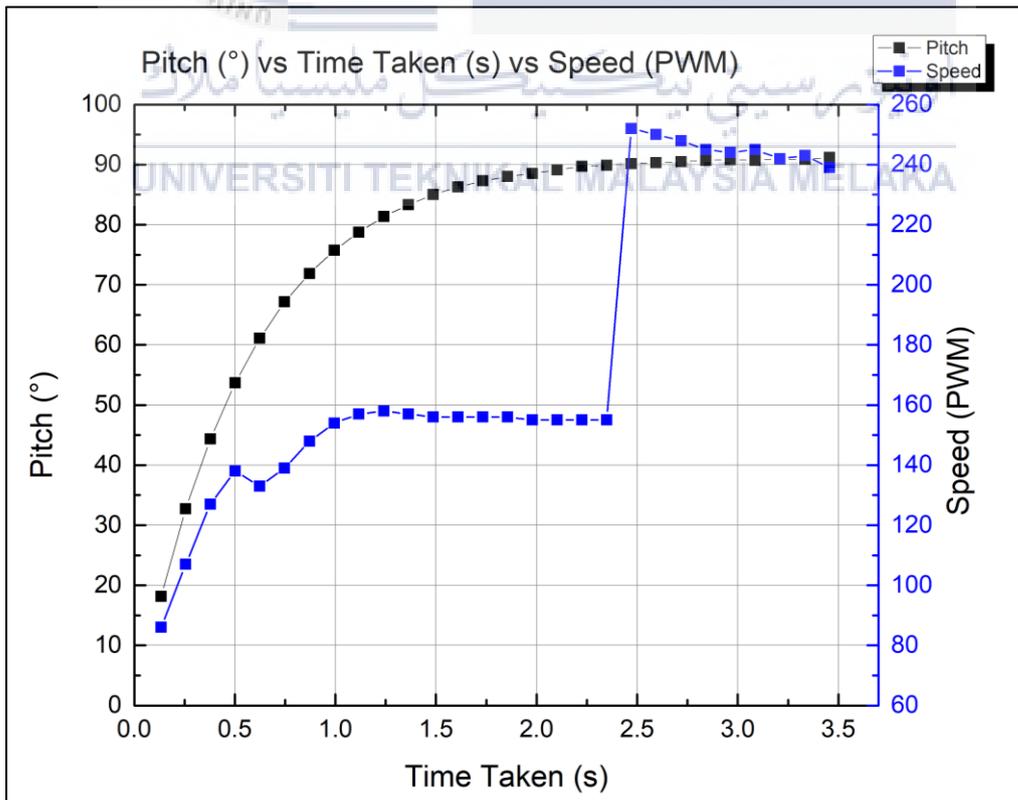
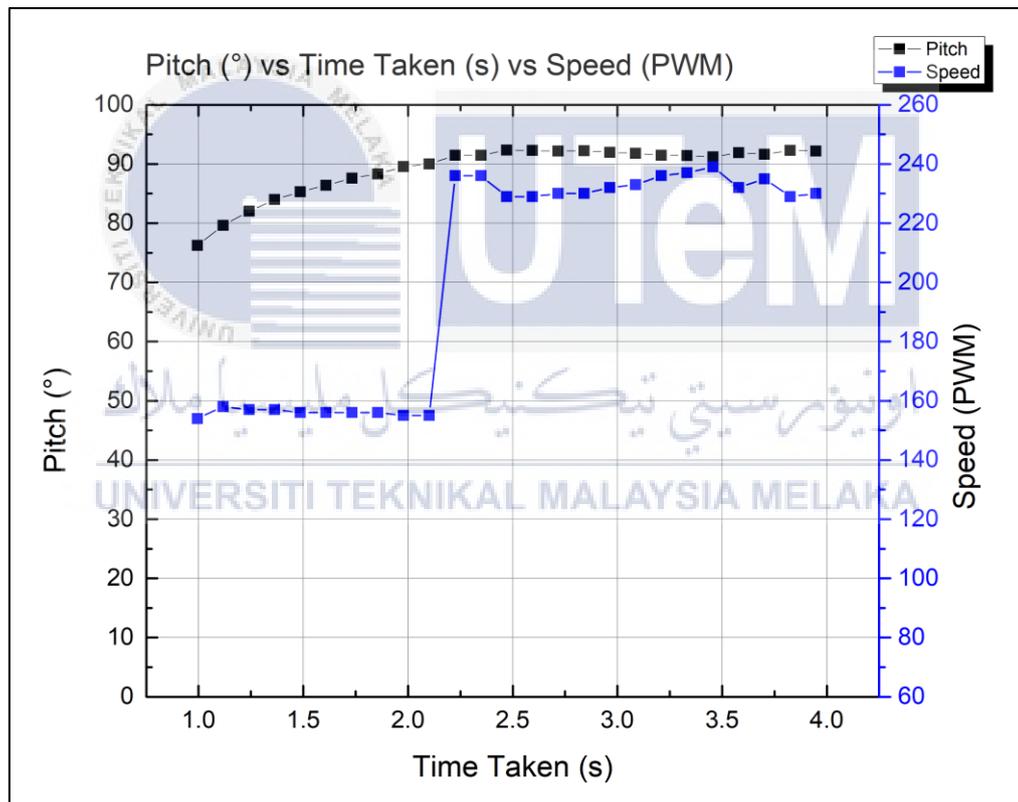


Figure 4.2: Characteristic of The Pitch Orientation And Motor's Speed Over Time For Tiles Surface

From the graph in Figure 4.2, the robot takes approximately 3.4 seconds to move 100 cm forward on a tiles surface. After 2.4 seconds, the IMU sensor reading stays around 90° while the output motor speed signal jumps from below 160 to above 240 and maintain there until the robot reached 100 cm.

This shows that after reaching the steady state reading of the IMU sensor, FLC system can detect the smooth tiles surface and maintain the high output speed since there is no irregularity on the terrain. This helps the robot reach 100 cm at a faster duration of time.

Figure 4.3: Characteristic of The Pitch Orientation And Motor's Speed Over



Time For Asphalt Surface

The graph from Figure 4.3 shows that the robot takes approximately 3.9 seconds to move 100 cm forward on an asphalt surface. After reaching the steady state reading of the IMU sensor at 2.2 seconds, the IMU sensor reading fluctuates between 91° and 92° while the output speed signal of the motor fluctuates from 229 to 237. This shows how the FLC

controller can detect the small irregularity of the asphalt surface and tries to compensate it by slightly adjusting the output motor speed signal.

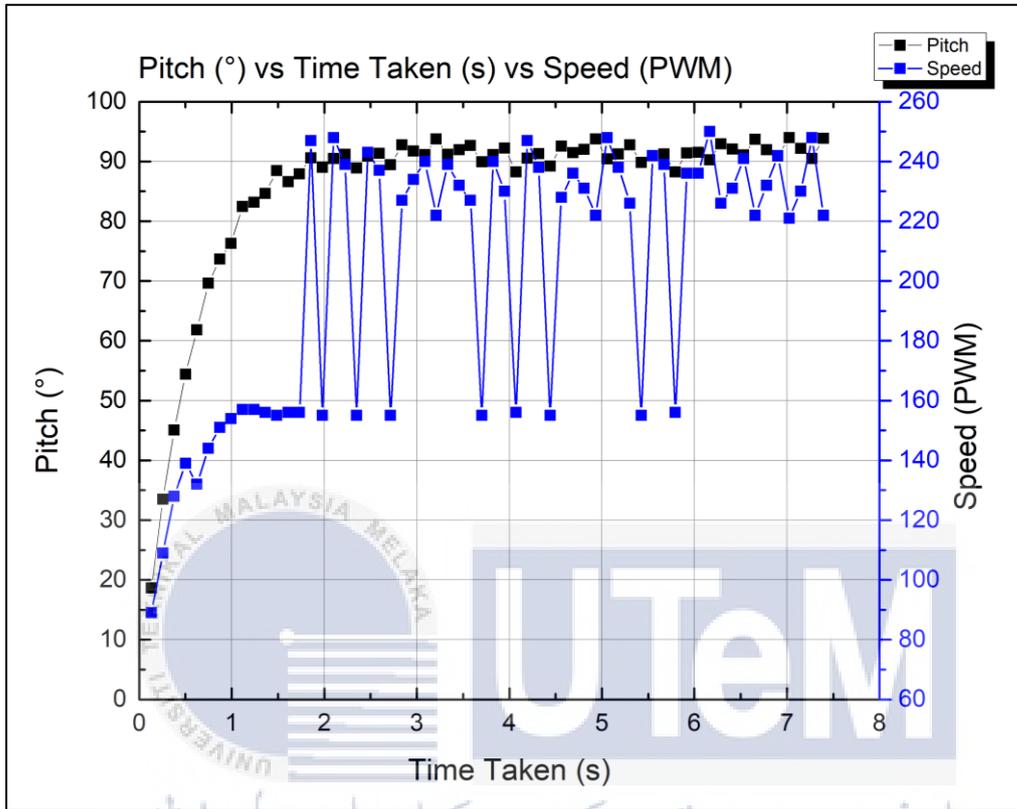


Figure 4.4: Characteristic of The Pitch Orientation And Motor's Speed Over Time For Rocky Surface

The graph from Figure 4.4 shows that the robot takes approximately 7.4 seconds to move 100 cm forward on rocky surface. After reaching the steady state reading of the IMU sensor at 1.8 seconds, the IMU sensor reading fluctuates between 88° and 93° while the output speed signal of the motor fluctuates from 155 to 248. This shows that the FLC controller detects a constant irregularity of the rocky surface and tries to compensate it with the corresponding output motor speed signal.

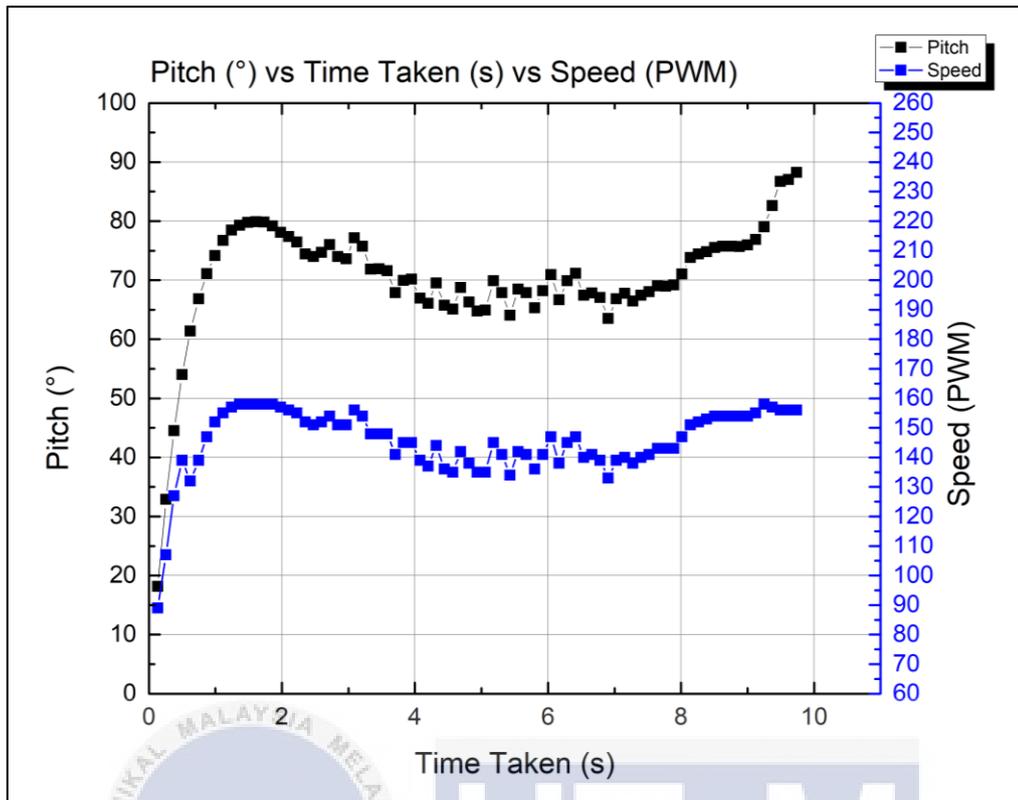


Figure 4.5: Characteristic of The Pitch Orientation And Motor's Speed Over Time For Hill Surface

The graph from Figure 4.5 shows how the robot takes approximately 9.7 seconds to move 100 cm forward on a hill surface. After reaching the steady state reading of the IMU sensor at 1.4 seconds, the IMU sensor reading drops from 79° at 1.8 seconds, to 64° at 4.9 seconds before climbing back up to 82° at 9.3 seconds. The output speed signal of the motor follows the trend of the IMU sensor reading by dropping from 158 at 1.8 seconds, to 135 at 4.9 seconds, and climbing back up to 157 at 9.3 seconds.

This shows that the FLC controller detects an incline surface of the hill terrain and compensate it by decreasing the speed of the motor. This helps overcome the tire slip that usually occurred on an incline surface by increasing the time of contact between the surface and the wheel. The controller then steadily increased the speed of the motor when the surface incline is decreasing at the top of the hill.

From all graphs, its can be seen that the controller needs an average of 1.95 seconds to reach a steady state reading from the IMU sensor. The time taken also decreased from the

as the surface irregularity increased. This shows that a higher irregularity of the IMU sensor reading helps increase the time taken for a steady state reading of the sensor since the complementary filter also depends on previous reading to compensate for the error occurred.

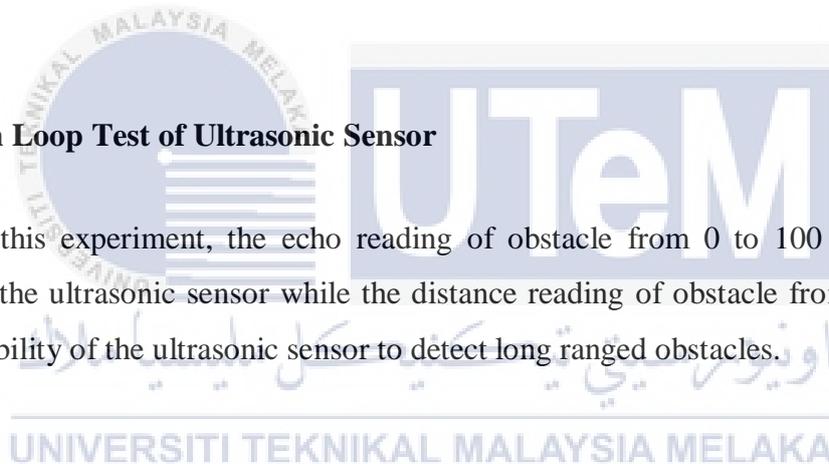
The time taken for the robot to move 100 cm forward also increased as the surface irregularity increased on each surface. This shows that the robot takes more time to overcome any surface irregularity since it slows down to increase the torque of the motor.

4.2 Experiment 2: Obstacle Detection

This section shows the accuracy of the ultrasonic sensor reading and the characteristic of the FLC system used to control the output motor speed based on the distance of the detected obstacles.

4.2.1 Open Loop Test of Ultrasonic Sensor

In this experiment, the echo reading of obstacle from 0 to 100 cm shows the accuracy of the ultrasonic sensor while the distance reading of obstacle from 0 to 400 cm shows the ability of the ultrasonic sensor to detect long ranged obstacles.



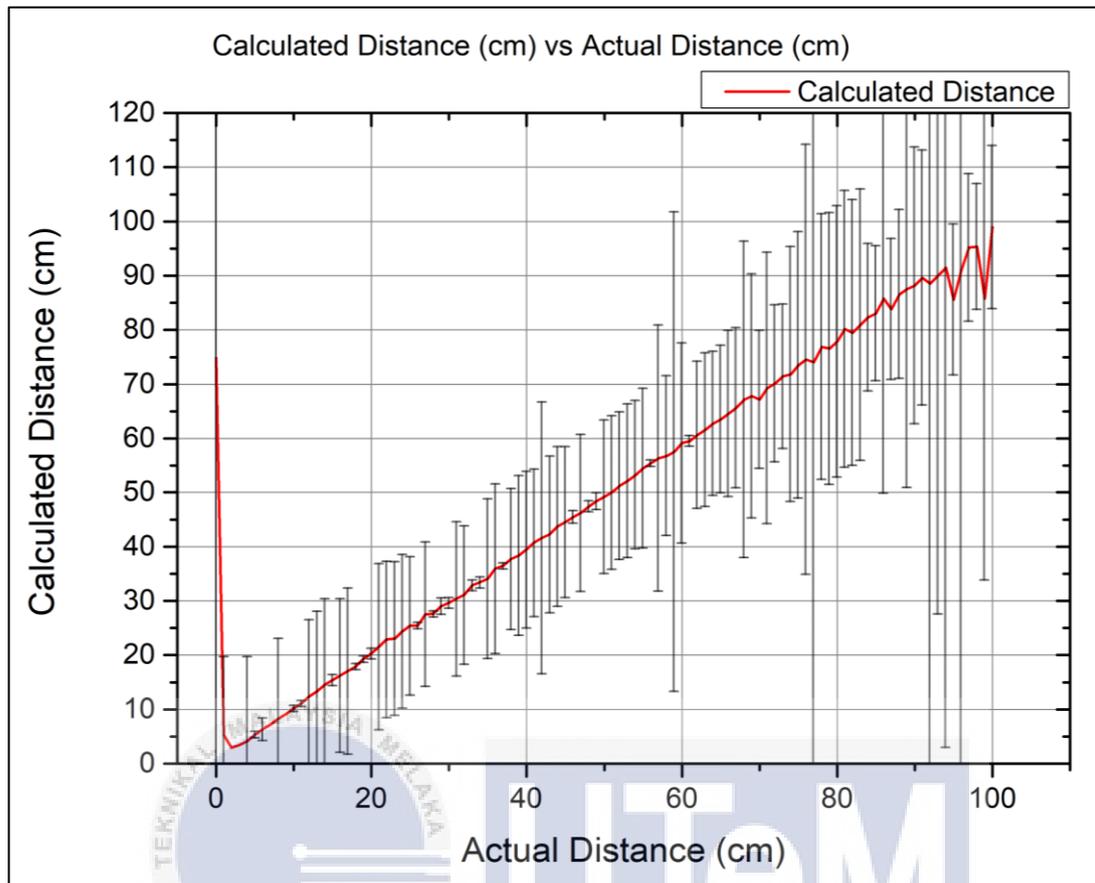


Figure 4.6: Accuracy of Ultrasonic Sensor for 0 to 100 cm Range

In the graph shown in Figure 4.6, the distance measured is accurate from 10 cm to 50 cm. After that, the reading is accurate with ± 5 cm error. This shows that the sensor can be used to measure the obstacle's distance accurately up to 50 cm. At 0 cm, the distance measured is at 76 cm. However, this will not affect the robot's movement since there will be a fixed distance between the robot and the obstacles when the robot stops moving.

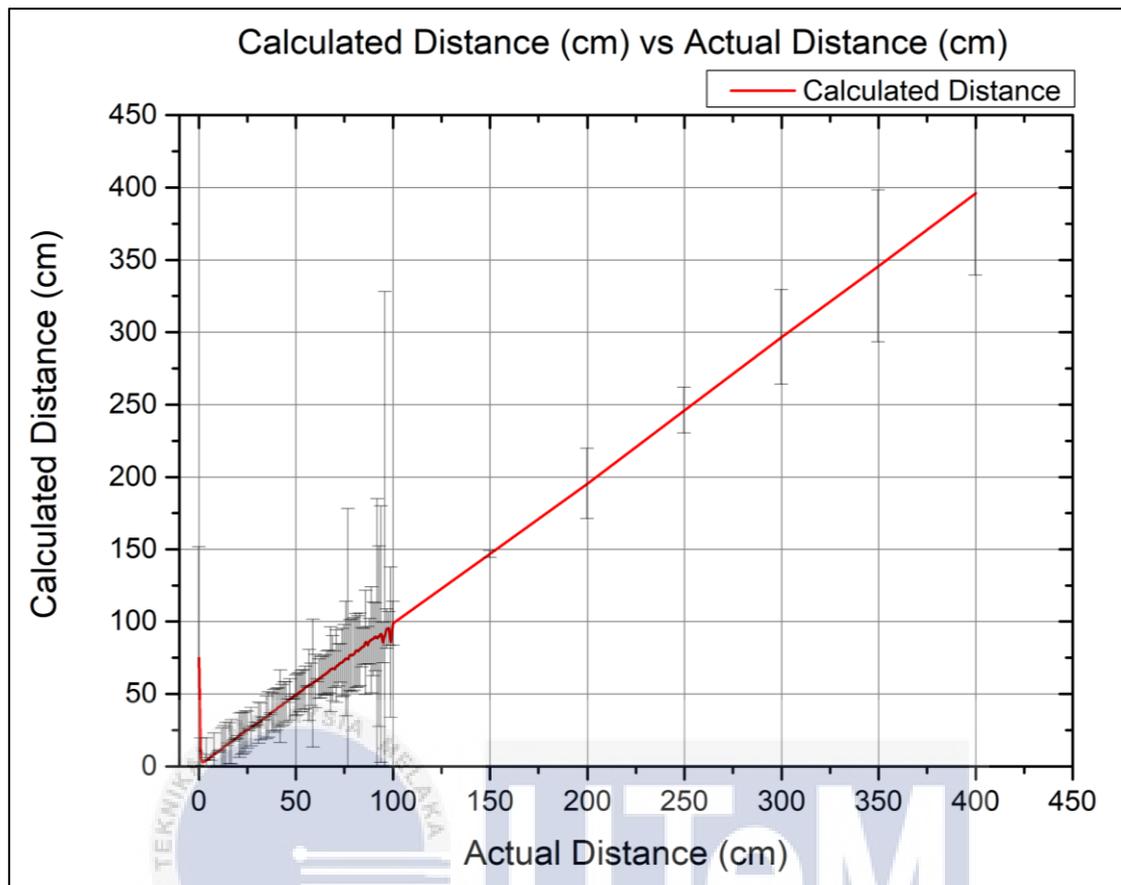


Figure 4.7: Accuracy of Ultrasonic Sensor from 0 to 400 cm

Graph in Figure 4.7 shows that from 100 to 400 cm, the measurements reading is 5 cm less than the actual distance. However, this will not matter as long as the robot is able to detect long ranged obstacle to adjust the approaching speed.

4.2.2 Closed Loop Test of Ultrasonic Sensor and Motor Speed Control Using Fuzzy Logic Controller

In this experiment, the data is graphed in a scattered manner to generate the characteristic of the FLC system used.

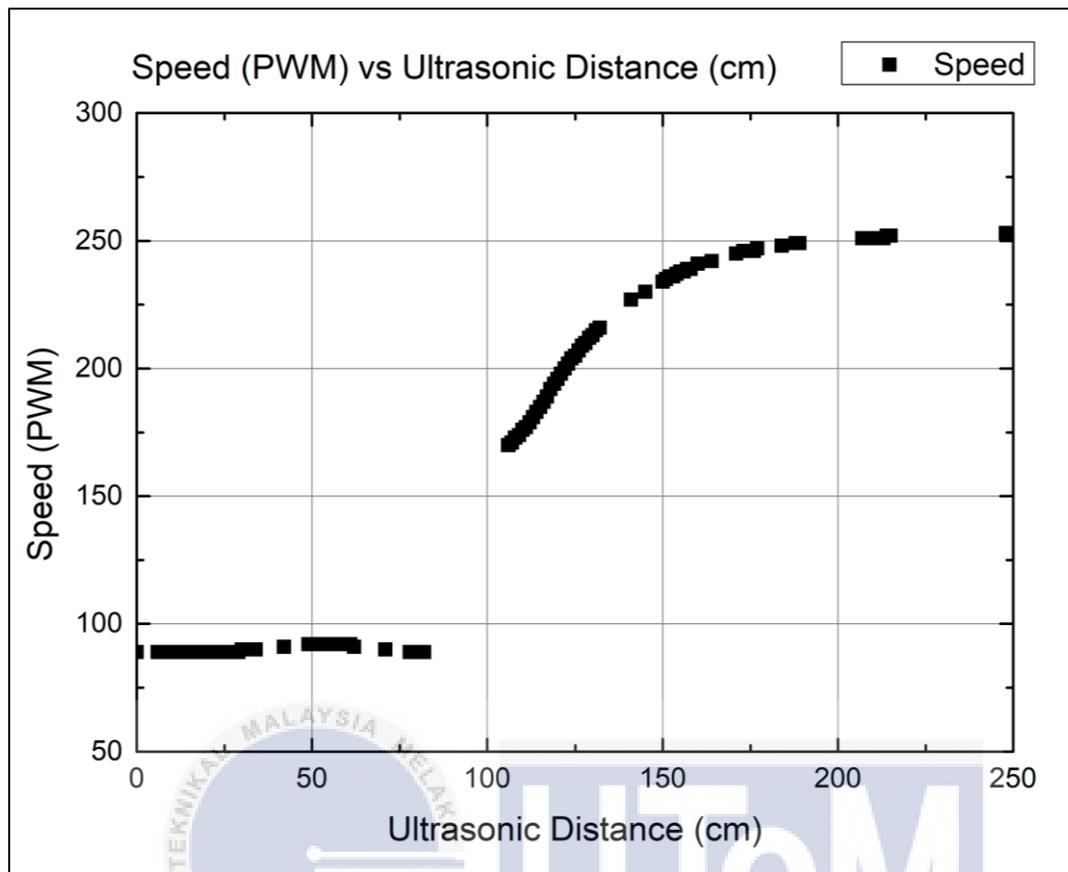


Figure 4.8: Behavior Characteristic of the FLC System

By comparing the graph in Figure 4.8 with the expected behavior in Figure, we can see a similarity on which the speed will increased when the distance is increasing. This also shows the ability of the robot to decrease its approaching speed as it is getting nearer to an obstacle.

4.2.3 Heading Selection

In this experiment, the distance measured are graphed to see the characteristic of the surrounding 180° area in front of the LIDAR sensor and the top three heading with the furthest distance is evaluated to choose the possible alternative route.

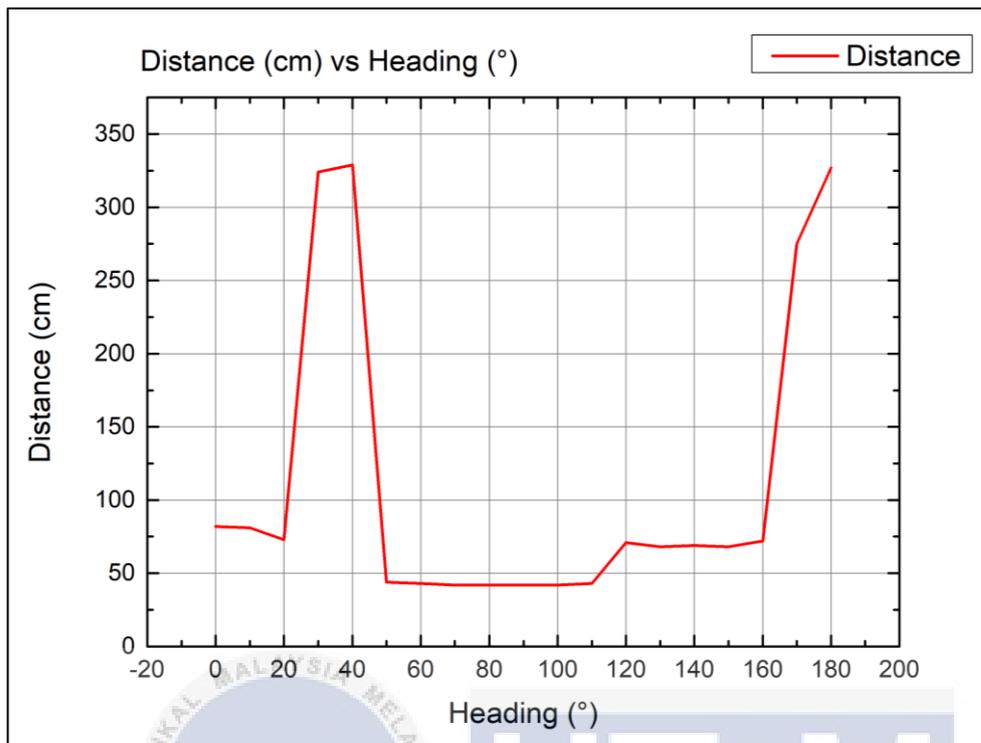


Figure 4.9: Distance Reading for 180° Area

From the graph in Figure 4.9, it is shown that the LIDAR system detects a presence of obstacle from 50° to 110° heading. This shows that the system is able to detect the obstacle that is below 60 cm.

Table 4.1: Evaluation Results of Possible Alternative Route Heading

Possible Alternative Route Heading (°)	Average Distance Reading for 20° Around Heading (cm)	Safety Threshold State
40	162.6	Not safe
180	534.8	Safe
30	170.2	Not Safe

In Table 4.1, the chosen possible alternative route is at 40° heading since it is the furthest and is deemed safe for the 20° around that direction.

4.3 Experiment 3: Designing Motion Control for Uneven Terrain

In this section, the measurements of the robot's pitch orientation, obstacle distance, and time taken to cover a distance of 100 cm is graphed to observe its characteristic on different terrains.

4.3.1 Tiles Surface

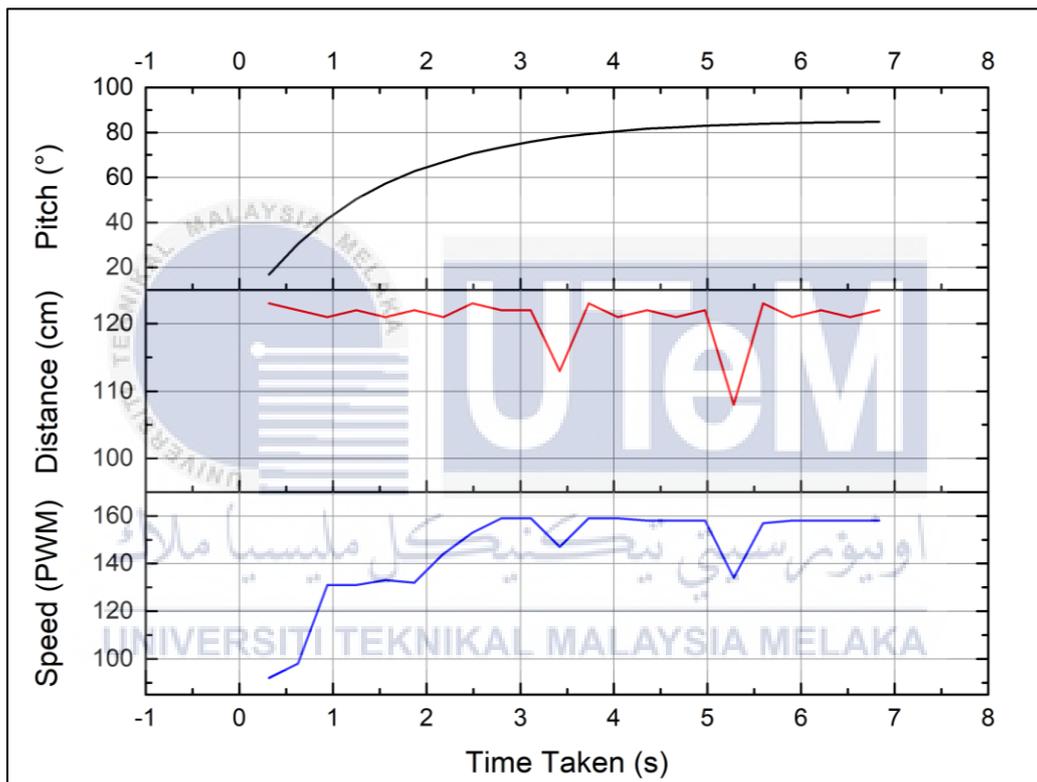


Figure 4.10: Characteristic Behavior of FLC System Used on Tiles Surface.

On tiles surface, the pitch orientation remains the same after reaching the steady state reading at 5 seconds. However, the distance reading and the output motor speed signal drops at at 3.4 seconds and 4.2 seconds. This is due to when the robot interacts with the grooves between the tiles, the ultrasonic sensor is pointer downward for a moment causing a decrease in the reading and the output signal. This shows that the robot is able to response to sudden appearance of obstacles while on smooth surface.

The highest output speed signal is 159 while the lowest output speed signal is 92. The time taken for the robot to move 100 cm forward is approximately 6.8 seconds which is lower than the time taken in Experiment 1. This is due to the additional ultrasonic sensor input to the FLC system causing it to have a more constricting rule.

4.3.2 Asphalt Surface

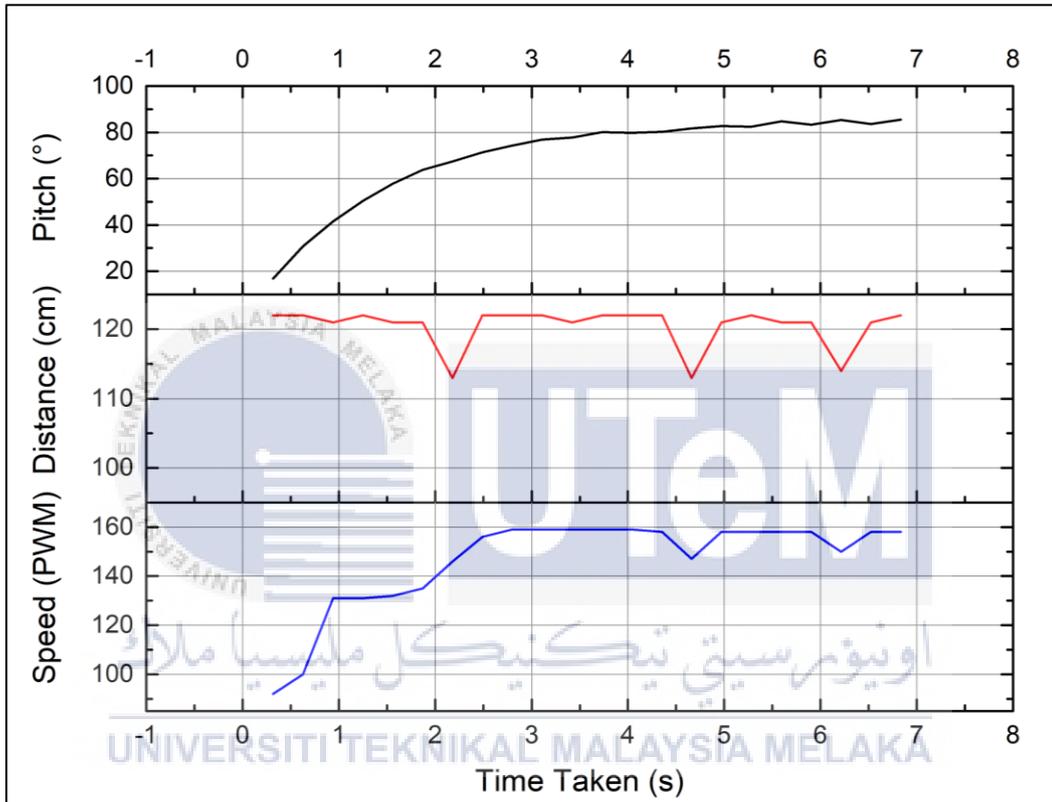


Figure 4.11: Characteristic Behavior of FLC System Used on Asphalt Surface

On asphalt surface, after reaching a steady state reading at 4 seconds, a slight variation of the pitch orientation can be seen occurring due to the slight irregularity of the asphalt surface. A drop in distance reading can be seen at 2.2 seconds, 4.6 seconds, and 6.2 seconds due to the sensor pointing downward for a moment during the slight irregularity. However, only the latter two are compensated since the IMU sensor reading reach a steady state at 4 seconds. This shows how the robot is able to adjust the output motor speed signal to compensate for two kind of errors occurring simultaneously.

4.3.3 Rocky Surface

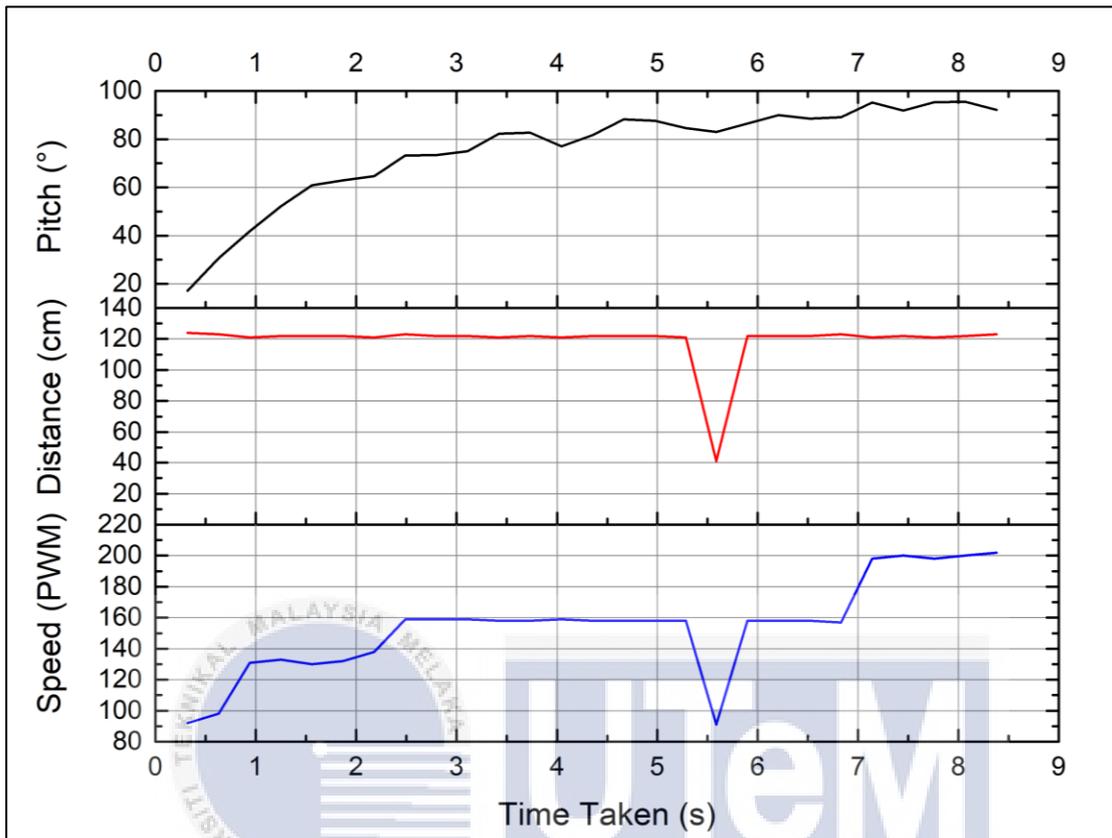


Figure 4.12: Characteristic Behavior of FLC System Used on Rocky Surface

On a rocky surface, the pitch reading from the IMU sensor shows the irregularity of the surface fluctuating as the reading reaches its steady state. A drop in the distance reading can be seen at 5.5 seconds which shows that the robot detects a sudden presence of obstacle when the robot is overcoming a slight incline. This is compensated by the FLC system by decreasing the output speed signal to 41. The robot takes 8.3 seconds to cover a distance of 100 cm on rocky surface. For most of the time, the output speed signal stays around 122 which cause the robot to move 100 cm forward slower than on asphalt and tiles surface.

4.3.4 Hill Surface

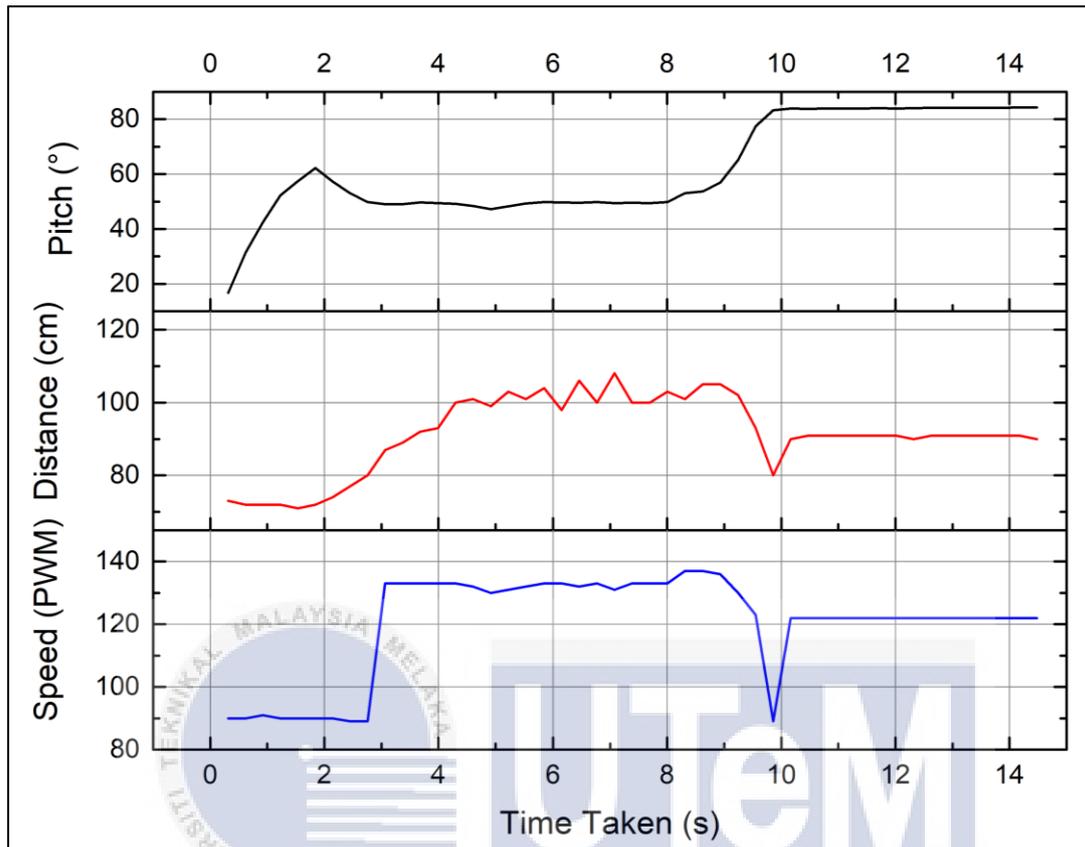


Figure 4.13: Characteristic Behavior of FLC System Used on Hill Surface

On hill surface, the pitch orientation reading starts to decrease at 1.8 seconds and increased back at 8.3 second. The distance reading started at a low value before climbing up as the pitch reading decreased and decreased again as the pitch reading increased. This shows that the robot started to climb the hill at 1.8 seconds and took 6.5 seconds to overcome the hill. During the climb, the output motor speed signal is maintained around 100 which helps maintain the torque of the robot's motor. During the declining surface, the output motor speed signal can be seen decreasing to avoid the robot from descending too fast. At 10.1 seconds and above, the output speed stays at 122 since the robot has reached the top of the surface.

4.3.5 Overall Behavior of FLC System

The readings of distance, pitch orientation and speed for each surface are collected and graph to observe its overall behavior.

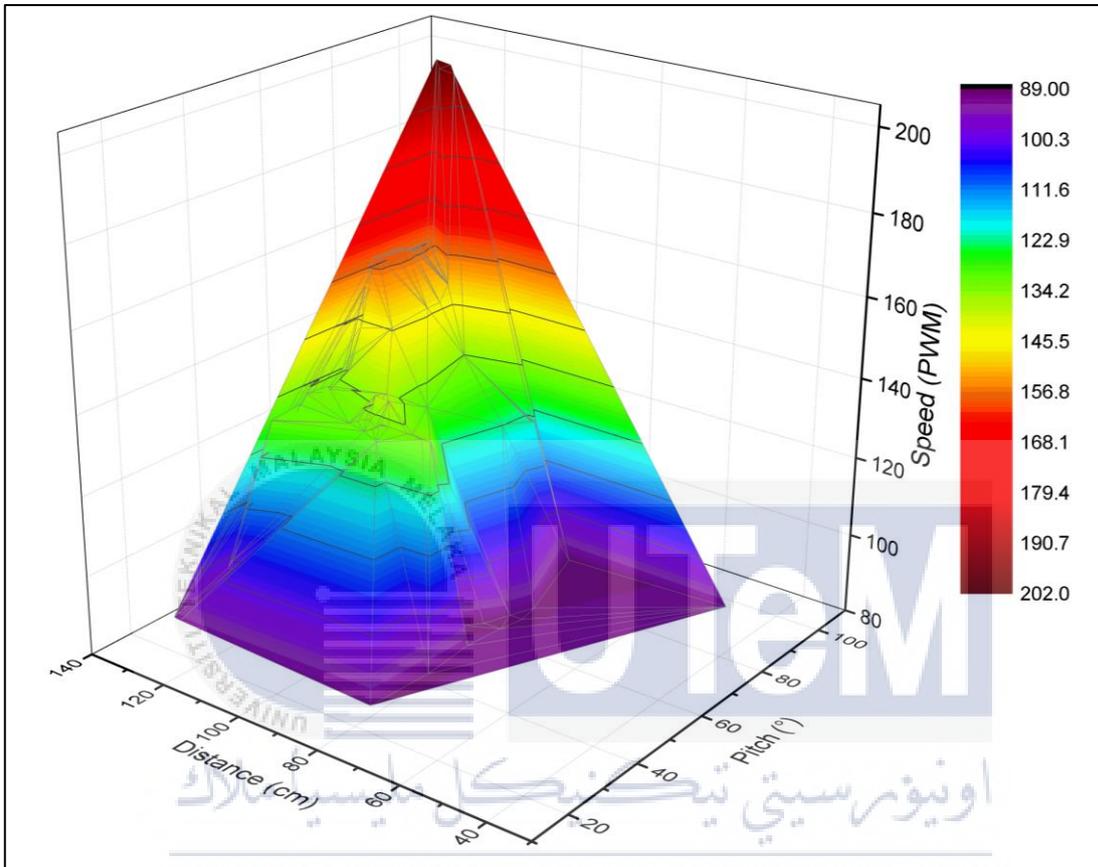


Figure 4.14: Overall Behavior of The MISO FLC System

By comparing the behavior graphed on Figure 4.14 to the expected behavior graphed in Figure, we can see that there are a lot of similarity to the behavior. When the distance increase, and the pitch is on the straight orientation, the output speed signal increased while when the distance increase, and the pitch is on the below 40° , the output speed signal maintains around 120.

This shows that the FLC system successfully creates a motion controller that can control the speed of the robot according to its surrounding as expected in the design of the system. However, since the readings are not taken to the full range of the inputs, only half of the behavior is can be graphed and compared.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In a conclusion, by combining information of the surface orientation and obstacle detection, it is possible to create sufficient fuzzy rules to develop a good motion controller. The methods used to collect data information on each area of study can be categorized as the simplest and cost effective for this project since even the sensors used are not more than 50 Malaysian Ringgit.

The IMU sensor used is the simplest method to get the orientation of the robot. Although it is still prone to noise and drift problems if used for a long time, through the experiment done in this project, the sensor can be said to have successfully help the robot to measure the surface orientation thus achieving the first objective.

The 180 degrees LIDAR sensor system consisting of a servo motor and an ultrasonic sensor helps the robot control its approaching speed and choose the best alternative route when being confronted with an obstacle which achieved the second objective. Although it takes a long time to for the sensor to scan for obstacle and may decrease the overall performance of the mobile robot.

Lastly, the motion controller developed using Fuzzy Logic Controller was able to help the robot adapt to its surrounding successfully with only motor speed control as its output. This shows that the concept proposed for building the Motion Controller is achievable thus achieving the third objective. However, this does not include the heading selection since the combination of all programming code is more than the Arduino board's memory can handle.

5.2 Future Works

For future improvements, the accuracy of the surface orientation measurements needs to be improved since in this project, drift and noise can occur when the sensor is used for a long time. The LIDAR coding needs to be redesign since there is a problem with time execution of the coding where the size of the code exceeds the memory of the Arduino board. A new body design is also needed since the current body actually gives a big disadvantage to the robot especially when turning or rotating.



REFERENCES

- [1] A. S. Andreyev and O. A. Peregudova, "The motion control of a wheeled mobile robot," *J. Appl. Math. Mech.*, vol. 79, no. 4, pp. 316–324, 2015.
- [2] C. A. Berry, *Mobile robotics for multidisciplinary study*, vol. 3, no. 1. 2012.
- [3] B. Puskas and Z. Rajnai, "Requirements of the Installation of the Critical Informational Infrastructure and Its Management," *Interdiscip. Descr. Complex Syst.*, vol. 13, no. 1, pp. 41–49, 2015.
- [4] O. Mokhiamar and S. Amine, "Lateral motion control of skid steering vehicles using full drive-by-wire system," *Alexandria Eng. J.*, vol. 56, no. 4, pp. 383–394, 2017.
- [5] V. Yerubandi, Y. M. Reddy, and M. V. K. Reddy, "Navigation system for an autonomous robot using fuzzy logic," vol. 5, no. 2, pp. 5–8, 2015.
- [6] J. Clark and R. Fierro, "Mobile robotic sensors for perimeter detection and tracking," *ISA Trans.*, vol. 46, no. 1, pp. 3–13, 2007.
- [7] H. Li and A. V. Savkin, "An algorithm for safe navigation of mobile robots by a sensor network in dynamic cluttered industrial environments," *Robot. Comput. Integr. Manuf.*, vol. 54, no. March 2017, pp. 65–82, 2018.
- [8] M. Duguleana and G. Mogan, "Neural networks based reinforcement learning for mobile robots obstacle avoidance," *Expert Syst. Appl.*, vol. 62, pp. 104–115, 2016.
- [9] H. Gao *et al.*, "Sinkage definition and visual detection for planetary rovers wheels on rough terrain based on wheel–soil interaction boundary," *Rob. Auton. Syst.*, vol. 98, no. June 2005, pp. 222–240, 2017.
- [10] N. H. Singh and K. Thongam, "Mobile Robot Navigation Using Fuzzy Logic in Static Environments," *Procedia Comput. Sci.*, vol. 125, pp. 11–17, 2018.
- [11] A. Zdešar, S. Blažic, and G. Klančar, "Engineering Education in Wheeled Mobile Robotics," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12173–12178, 2017.
- [12] I. Matraji, A. Al-Durra, A. Haryono, K. Al-Wahedi, and M. Abou-Khousa, "Trajectory tracking control of Skid-Steered Mobile Robot based on adaptive Second Order Sliding Mode Control," *Control Eng. Pract.*, vol. 72, no. October 2017, pp. 167–176, 2018.

- [13] S. Noga, "Kinematics and dynamics of some selected two-wheeled mobile robots," *Arch. Civ. Mech. Eng.*, vol. 6, no. 3, pp. 55–70, 2006.
- [14] N. Leena and K. K. Saju, "Modelling and Trajectory Tracking of Wheeled Mobile Robots," *Procedia Technol.*, vol. 24, pp. 538–545, 2016.
- [15] H.R. Everett, *SensorsForMobileRobots book*. CRC Press, 1995.
- [16] J. Candiotti *et al.*, "Design and evaluation of a seat orientation controller during uneven terrain driving," *Med. Eng. Phys.*, vol. 38, no. 3, pp. 241–247, 2016.
- [17] V. P. Eathakota, A. K. Singh, and K. Madhava Krishna, "Two models of force actuator based active suspension mechanisms for mobility on uneven terrain," *Acta Astronaut.*, vol. 67, no. 9–10, pp. 1233–1247, 2010.
- [18] P. Skrzypczyński, "Genetic algorithm based learning in a fuzzy logic mobile robot controller," *IFAC Proc. Vol.*, vol. 36, no. 17, pp. 503–508, 2003.
- [19] S. B. Mane and S. Vhanale, "Real time obstacle detection for mobile robot navigation using stereo vision," *Int. Conf. Comput. Anal. Secur. Trends, CAST 2016*, pp. 637–642, 2017.
- [20] B. Bovcon, R. Mandeljc, J. Perš, and M. Kristan, "Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation," *Rob. Auton. Syst.*, vol. 104, pp. 1–13, 2018.
- [21] Y. Wang and R. Rajamani, "Direction cosine matrix estimation with an inertial measurement unit," *Mech. Syst. Signal Process.*, vol. 109, pp. 268–284, 2018.
- [22] B. Anoohya B and R. Padhi, "Trajectory Tracking of Autonomous Mobile Robots Using Nonlinear Dynamic Inversion," *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 202–207, 2018.
- [23] I. Carlucho, M. De Paula, S. A. Villar, and G. G. Acosta, "Incremental Q-learning strategy for adaptive PID control of mobile robots," *Expert Syst. Appl.*, vol. 80, pp. 183–199, 2017.
- [24] L. Chun-rui, K. Li-jun, Q. Qing-xing, M. De-bing, L. Quan-ming, and X. Gang, "The numerical analysis of borehole blasting and application in coal mine roof-weaken," *Procedia Earth Planet. Sci.*, vol. 1, no. 1, pp. 451–459, 2009.
- [25] O. Yazdanbakhsh and S. Dick, "A systematic review of complex fuzzy sets and logic," *Fuzzy Sets Syst.*, vol. 338, pp. 1–22, 2018.
- [26] P. Reignier, "Fuzzy logic techniques for mobile robot obstacle avoidance," *Rob.*

Auton. Syst., vol. 12, no. 3–4, pp. 143–153, 1994.

- [27] S. Parasuraman, “Sensor fusion for mobile robot navigation: Fuzzy Associative Memory,” *Procedia Eng.*, vol. 41, no. Iris, pp. 251–256, 2012.



APPENDICES

APPENDIX A EXPERIMENT 1 OPEN LOOP TEST RESULTS

Angle (°)	Accelerometer (°)	Gyroscope (°)	Complementary Filter (°)
0	0.22	3.6	0.68
10	10.65	15.95	10.1
20	20.29	28.9	20.15
30	30.7	40.43	30.15
40	41.8	52.33	40.98
50	50.54	61.53	50.61
60	59.72	72.2	60.55
70	70.38	80.95	70.05
80	80.32	91.51	80.54
90	90.05	100.94	90.1
100	101.15	111.65	100.86
110	110.89	122.18	110.8
120	120.23	131.19	120.1
130	130.61	141.89	130.4
140	139.06	151.75	140.5
150	150.31	162.14	150.57
160	160.14	171.76	160.5
170	171.2	185.09	170.92
180	173.27	195.77	173.3

APPENDIX B EXPERIMENT 1 CLOSE LOOP RESULT FOR TILES SURFACE

No of reading, n	Pitch of robot (°)	Speed of motor (PWM)	Time taken (s)
1	18.17	86	0.133
2	32.74	107	0.256
3	44.35	127	0.378
4	53.69	138	0.501
5	61.11	133	0.624
6	67.16	139	0.747
7	71.89	148	0.871
8	75.74	154	0.994
9	78.75	157	1.117
10	81.36	158	1.241
11	83.33	157	1.363
12	85	156	1.486

13	86.26	156	1.609
14	87.3	156	1.733
15	88.01	156	1.856
16	88.52	155	1.979
17	89.09	155	2.103
18	89.71	155	2.226
19	89.87	155	2.349
20	90.16	252	2.471
21	90.3	250	2.594
22	90.49	248	2.718
23	90.72	245	2.841
24	90.79	244	2.964
25	90.72	245	3.087
26	90.93	242	3.21
27	90.87	243	3.334
28	91.2	239	3.457

APPENDIX C EXPERIMENT 1 CLOSE LOOP RESULT FOR ASPHALT SURFACE

No of reading, n	Pitch of robot (°)	Speed of motor (PWM)	Time taken (s)
1	76.24	154	0.994
2	79.6	158	1.117
3	82.02	157	1.241
4	83.96	157	1.363
5	85.31	156	1.486
6	86.4	156	1.609
7	87.6	156	1.733
8	88.3	156	1.856
9	89.55	155	1.979
10	89.98	155	2.102
11	91.46	236	2.226
12	91.45	236	2.349
13	92.35	229	2.471
14	92.28	229	2.594
15	92.15	230	2.717
16	92.24	230	2.841
17	91.98	232	2.964
18	91.79	233	3.087
19	91.49	236	3.21
20	91.42	237	3.333

21	91.2	239	3.456
22	91.9	232	3.579
23	91.62	235	3.702
24	92.31	229	3.825
25	92.19	230	3.948

APPENDIX D EXPERIMENT 1 CLOSE LOOP RESULT FOR ROCKY SURFACE

No of reading, n	Pitch of robot (°)	Speed of motor (PWM)	Time taken (s)
1	18.63	89	0.133
2	33.5	109	0.256
3	45.09	128	0.378
4	54.41	139	0.501
5	61.85	132	0.624
6	69.61	144	0.748
7	73.66	151	0.871
8	76.28	154	0.994
9	82.44	157	1.117
10	83.18	157	1.241
11	84.66	156	1.363
12	88.45	155	1.486
13	86.61	156	1.609
14	87.94	156	1.733
15	90.58	247	1.856
16	88.99	155	1.979
17	90.5	248	2.102
18	91.18	239	2.225
19	88.91	155	2.349
20	90.89	243	2.471
21	91.35	237	2.594
22	89.45	155	2.717
23	92.78	227	2.841
24	91.74	234	2.964
25	91.15	240	3.087
26	93.76	222	3.21
27	91.2	239	3.333
28	91.98	232	3.457
29	92.69	227	3.579
30	89.96	155	3.702
31	91.11	240	3.825
32	92.23	230	3.948

33	88.25	156	4.072
34	90.54	247	4.195
35	91.31	238	4.318
36	89.25	155	4.441
37	92.53	228	4.563
38	91.49	236	4.687
39	92.02	231	4.81
40	93.78	222	4.933
41	90.45	248	5.056
42	91.27	238	5.179
43	92.8	226	5.302
44	89.8	155	5.426
45	90.99	242	5.549
46	91.23	239	5.671
47	88.26	156	5.794
48	91.43	236	5.917
49	91.51	236	6.041
50	90.28	250	6.164
51	92.96	226	6.287
52	92.04	231	6.41
53	91.06	241	6.533
54	93.71	222	6.657
55	91.94	232	6.779
56	90.94	242	6.902
57	93.96	221	7.025
58	92.2	230	7.148
59	90.48	248	7.271
60	93.87	222	7.395

APPENDIX E EXPERIMENT 1 CLOSE LOOP RESULT FOR HILL SURFACE

No of reading, n	Pitch of robot (°)	Speed of motor (PWM)	Time taken (s)
1	18.16	89	0.133
2	32.87	107	0.256
3	44.52	127	0.378
4	54	139	0.501
5	61.4	132	0.624
6	66.82	139	0.747
7	71.1	147	0.871
8	74.14	152	0.994
9	76.73	155	1.117

10	78.48	157	1.241
11	79.29	158	1.363
12	79.81	158	1.486
13	79.91	158	1.61
14	79.86	158	1.733
15	79.18	158	1.856
16	78.08	157	1.98
17	77.37	156	2.103
18	76.48	155	2.226
19	74.46	152	2.35
20	74.02	151	2.472
21	74.68	152	2.595
22	76.04	154	2.718
23	73.98	151	2.842
24	73.64	151	2.965
25	77.15	156	3.089
26	75.77	154	3.212
27	71.86	148	3.335
28	71.91	148	3.459
29	71.58	148	3.581
30	67.91	141	3.704
31	69.97	145	3.827
32	70.15	145	3.951
33	66.95	139	4.074
34	66.06	137	4.197
35	69.52	144	4.321
36	65.75	136	4.444
37	65.11	135	4.567
38	68.74	142	4.689
39	66.31	138	4.812
40	64.75	135	4.936
41	64.94	135	5.059
42	69.88	145	5.182
43	67.89	141	5.306
44	64.04	134	5.429
45	68.46	142	5.552
46	67.9	141	5.675
47	65.33	136	5.798
48	68.19	141	5.921
49	70.92	147	6.044
50	66.66	138	6.167
51	69.91	145	6.291

52	71.17	147	6.414
53	67.44	140	6.537
54	67.83	141	6.661
55	67.08	139	6.784
56	63.52	133	6.906
57	66.82	139	7.029
58	67.76	140	7.153
59	66.49	138	7.276
60	67.45	140	7.399
61	68.03	141	7.522
62	69.04	143	7.646
63	68.95	143	7.769
64	69.21	143	7.891
65	71.05	147	8.015
66	73.81	151	8.138
67	74.42	152	8.261
68	74.84	153	8.385
69	75.51	154	8.508
70	75.73	154	8.631
71	75.75	154	8.755
72	75.68	154	8.878
73	75.97	154	9
74	76.88	155	9.124
75	79.04	158	9.247
76	82.62	157	9.37
77	86.73	156	9.494
78	87.03	156	9.617
79	88.22	156	9.74

APPENDIX F EXPERIMENT 2 OPEN LOOP RESULT

Actual Distance (cm)	Calculated Echo Duration (μ s)	Echo Duration Reading (μ s)					Calculated Distance (cm)
		1	2	3	Average	Standard Deviation	
0	0	134357	134292	134445	134364.66	-134364.66	35.8
1	58.30904	292	317	317	308.66	14.43375673	5.29
2	116.61808	174	174	174	174	0	2.98
3	174.92711	201	201	201	201	0	3.44
4	233.23615	253	253	226	244	15.58845727	4.18
5	291.54519	314	314	313	313.66	0.577350269	5.37

6	349.85423	372	371	368	370.33	2.081665999	6.35
7	408.16327	420	420	420	420	0	7.2
8	466.4723	502	478	475	485	14.79864859	8.31
9	524.78134	533	533	533	533	0	9.14
10	583.09038	593	594	594	593.66	0.577350269	10.18
11	641.39942	648	649	648	648.33	0.577350269	11.11
12	699.70845	706	731	730	722.33	14.15391583	12.38
13	758.01749	761	786	787	778	14.73091986	13.34
14	816.32653	840	839	867	848.66	15.88500341	14.55
15	874.63557	899	898	900	899	1	15.41
16	932.94461	955	956	931	947.33	14.15391583	16.24
17	991.25364	986	1013	987	995.33	15.30795	17.06
18	1049.56268	1046	1045	1046	1045.66	0.577350269	17.93
19	1107.87172	1126	1126	1125	1125.66	0.577350269	19.3
20	1166.18076	1184	1185	1183	1184	1	20.3
21	1224.4898	1266	1265	1239	1256.66	15.30795	21.55
22	1282.79883	1344	1319	1344	1335.66	14.43375673	22.9
23	1341.10787	1329	1353	1354	1345.33	14.15391583	23.07
24	1399.41691	1414	1439	1415	1422.66	14.15391583	24.39
25	1457.72595	1476	1497	1474	1482.33	12.7410099	25.42
26	1516.03499	1485	1485	1484	1484.66	0.577350269	25.46
27	1574.34402	1600	1622	1598	1606.66	13.31665624	27.55
28	1632.65306	1610	1609	1609	1609.33	0.577350269	27.6
29	1690.9621	1692	1693	1695	1693.33	1.527525232	29.04
30	1749.27114	1728	1729	1727	1728	1	29.63
31	1807.58017	1756	1782	1779	1772.33	14.2243922	30.39
32	1865.88921	1829	1806	1808	1814.33	12.7410099	31.11
33	1924.19825	1918	1917	1919	1918	1	32.89
34	1982.50729	1948	1950	1949	1949	1	33.42
35	2040.81633	1980	2005	1979	1988	14.73091986	34.09
36	2099.12536	2088	2114	2086	2096	15.62049935	35.94
37	2157.4344	2125	2126	2126	2125.66	0.577350269	36.45
38	2215.74344	2195	2217	2194	2202	13	37.76
39	2274.05248	2230	2231	2256	2239	14.73091986	38.39
40	2332.36152	2308	2310	2284	2300.66	14.46835628	39.45
41	2390.67055	2367	2368	2391	2375.33	13.57694124	40.73
42	2448.97959	2426	2453	2403	2427.33	25.02665246	41.62
43	2507.28863	2458	2483	2458	2466.33	14.43375673	42.29
44	2565.59767	2534	2560	2559	2551	14.73091986	43.74
45	2623.90671	2614	2591	2589	2598	13.89244399	44.55
46	2682.21574	2652	2650	2652	2651.33	1.154700538	45.47
47	2740.52478	2680	2706	2704	2696.66	14.46835628	46.24

48	2798.83382	2767	2768	2766	2767	1	47.45
49	2857.14286	2822	2823	2825	2823.33	1.527525232	48.42
50	2915.4519	2878	2853	2877	2869.33	14.15391583	49.2
51	2973.76093	2908	2933	2909	2916.66	14.15391583	50.02
52	3032.06997	2998	2997	2974	2989.66	13.57694124	51.27
53	3090.37901	3051	3026	3050	3042.33	14.15391583	52.17
54	3148.68805	3092	3117	3114	3107.66	13.65039682	53.29
55	3206.99708	3170	3195	3169	3178	14.73091986	54.5
56	3265.30612	3231	3230	3231	3230.66	0.577350269	55.4
57	3323.61516	3286	3311	3262	3286.33	24.50170062	56.36
58	3381.9242	3303	3304	3329	3312	14.73091986	56.8
59	3440.23324	3329	3330	3406	3355	44.17012565	57.53
60	3498.54227	3433	3469	3444	3448.66	18.44812547	59.14
61	3556.85131	3469	3471	3470	3470	1	59.51
62	3615.16035	3528	3527	3551	3535.33	13.57694124	60.63
63	3673.46939	3584	3609	3585	3592.66	14.15391583	61.61
64	3731.77843	3667	3644	3667	3659.33	13.27905619	62.75
65	3790.08746	3721	3698	3697	3705.33	13.57694124	63.54
66	3848.3965	3755	3756	3782	3764.33	15.30795	64.55
67	3906.70554	3810	3835	3836	3827	14.73091986	65.63
68	3965.01458	3900	3899	3950	3916.33	29.16047553	67.16
69	4023.32362	3931	3976	3954	3953.66	22.50185178	67.8
70	4081.63265	3909	3931	3909	3916.33	12.70170592	67.16
71	4139.94169	4016	4066	4040	4040.66	25.00666578	69.29
72	4198.25073	4096	4072	4098	4088.66	14.46835628	70.12
73	4256.55977	4156	4180	4158	4164.66	13.31665624	71.42
74	4314.8688	4188	4211	4164	4187.66	23.50177298	71.81
75	4373.17784	4285	4312	4263	4286.66	24.54248018	73.51
76	4431.48688	4311	4338	4389	4346	39.61060464	74.53
77	4489.79592	4244	4270	4436	4316.66	104.1601331	74.03
78	4548.10496	4459	4508	4483	4483.33	24.50170062	76.88
79	4606.41399	4440	4490	4463	4464.33	25.02665246	76.56
80	4664.72303	4563	4513	4540	4538.66	25.02665246	77.83
81	4723.03207	4673	4649	4700	4674	25.51470164	80.15
82	4781.34111	4611	4635	4660	4635.33	24.50170062	79.49
83	4839.65015	4695	4745	4721	4720.33	25.00666578	80.95
84	4897.95918	4808	4784	4807	4799.66	13.57694124	82.31
85	4956.26822	4834	4856	4835	4841.66	12.42309677	83.03
86	5014.57726	4992	5041	4971	5001.33	35.92121008	85.77
87	5072.8863	4903	4880	4881	4888	13	83.82
88	5131.19534	5058	5031	5058	5049	15.58845727	86.59
89	5189.50437	5145	5094	5074	5104.33	36.61056314	87.53

90	5247.81341	5166	5142	5115	5141	25.51470164	88.16
91	5306.12245	5247	5223	5200	5223.33	23.50177298	89.58
92	5364.43149	5251	5179	5060	5163.33	96.45897228	88.55
93	5422.74052	5183	5307	5257	5249	62.38589584	90.02
94	5481.04956	5232	5395	5373	5333.33	88.44395589	91.46
95	5539.3586	4984	5007	4982	4991	13.89244399	85.59
96	5597.66764	5453	5429	5031	5304.33	237.0175802	90.96
97	5655.97668	5534	5557	5558	5549.66	13.57694124	95.17
98	5714.28571	5558	5572	5549	5559.66	11.59022577	95.34
99	5772.59475	5060	4986	4960	5002	51.88448708	85.78
100	5830.90379	5785	5760	5758	5767.66	15.0443788	98.91
150	8746.35	8561	8557	8557	8558.33	188.02	146.77
200	11661.8	11405	11422	11374	11400.33	261.47	195.51
250	14577.25	14369	14342	14341	14350.66	226.59	246.11
300	17492.71	17283	17284	17340	17302.33	190.38	296.73
350	20408.16	20113	20147	20216	20158.66	249.5	345.72
400	23323.61	23024	23137	23087	23082.66	240.95	395.86

APPENDIX G EXPERIMENT 2 CLOSE LOOP TEST RESULT

No	Distance (cm)	Motor Speed (PWM)
1	209	251
2	209	251
3	209	251
4	209	251
5	207	251
6	207	251
7	209	251
8	208	251
9	208	251
10	208	251
11	208	251
12	209	251
13	209	251
14	208	251
15	211	251
16	211	251
17	210	251
18	210	251

19	209	251
20	211	251
21	209	251
22	209	251
23	211	251
24	210	251
25	209	251
26	210	251
27	208	251
28	209	251
29	210	251
30	209	251
31	252	253
32	174	246
33	175	246
34	175	246
35	126	207
36	141	227
37	130	213
38	124	204
39	126	207
40	132	216
41	129	212
42	124	204
43	123	202
44	128	210
45	127	209
46	127	209
47	128	210
48	128	210
49	128	210
50	127	209
51	115	185
52	125	205
53	118	192
54	124	204
55	124	204
56	175	246
57	289	253
58	212	251
59	209	251
60	209	251

61	210	251
62	210	251
63	209	251
64	209	251
65	210	251
66	210	251
67	211	251
68	211	251
69	248	253
70	173	246
71	125	205
72	124	204
73	115	185
74	114	183
75	110	176
76	111	177
77	110	176
78	111	177
79	110	176
80	112	179
81	113	181
82	110	176
83	111	177
84	111	177
85	113	181
86	61	92
87	59	92
88	58	92
89	59	92
90	109	174
91	109	174
92	123	202
93	115	185
94	113	181
95	112	179
96	112	179
97	112	179
98	122	200
99	215	252
100	214	252
101	212	251

102	214	252
103	213	251
104	211	251
105	209	251
106	209	251
107	209	251
108	210	251
109	210	251
110	211	251
111	210	251
112	210	251
113	210	251
114	210	251
115	211	251
116	210	251
117	212	251
118	210	251
119	215	252
120	211	251
121	212	251
122	211	251
123	210	251
124	210	251
125	211	251
126	212	251
127	211	251
128	211	251
129	212	251
130	211	251
131	212	251
132	211	251
133	211	251
134	211	251
135	212	251
136	211	251
137	211	251
138	212	251
139	210	251
140	211	251
141	211	251
142	211	251
143	212	251

144	213	251
145	211	251
146	211	251
147	210	251
148	213	251
149	211	251
150	212	251
151	211	251
152	214	252
153	215	252
154	123	202
155	124	204
156	124	204
157	112	179
158	110	176
159	112	179
160	111	177
161	111	177
162	0	89
163	61	92
164	59	92
165	57	92
166	71	90
167	56	92
168	62	91
169	61	92
170	109	174
171	109	174
172	122	200
173	114	183
174	110	176
175	111	177
176	110	176
177	112	179
178	112	179
179	112	179
180	125	205
181	123	202
182	112	179
183	123	202
184	112	179
185	124	204

186	126	207
187	125	205
188	124	204
189	125	205
190	115	185
191	26	89
192	25	89
193	23	89
194	23	89
195	22	89
196	22	89
197	22	89
198	23	89
199	23	89
200	24	89
201	24	89
202	24	89
203	24	89
204	24	89
205	78	89
206	24	89
207	31	90
208	26	89
209	29	89
210	27	89
211	27	89
212	28	89
213	29	89
214	28	89
215	31	90
216	30	90
217	29	89
218	28	89
219	27	89
220	26	89
221	24	89
222	26	89
223	27	89
224	27	89
225	27	89
226	29	89
227	29	89

228	106	170
229	30	90
230	30	90
231	31	90
232	106	170
233	49	92
234	109	174
235	107	171
236	106	170
237	106	170
238	108	173
239	150	234
240	155	238
241	152	236
242	156	238
243	152	236
244	157	239
245	155	238
246	164	242
247	175	246
248	156	238
249	130	213
250	153	236
251	131	215
252	157	239
253	151	235
254	151	235
255	25	89
256	25	89
257	23	89
258	23	89
259	25	89
260	27	89
261	24	89
262	24	89
263	25	89
264	34	90
265	20	89
266	19	89
267	22	89
268	20	89
269	21	89

270	21	89
271	57	92
272	19	89
273	15	89
274	15	89
275	15	89
276	15	89
277	14	89
278	14	89
279	14	89
280	50	92
281	53	92
282	52	92
283	7	89
284	7	89
285	6	89
286	6	89
287	8	89
288	9	89
289	13	89
290	13	89
291	82	89
292	22	89
293	25	89
294	22	89
295	22	89
296	20	89
297	17	89
298	16	89
299	15	89
300	16	89
301	18	89
302	127	209
303	126	207
304	126	207
305	125	205
306	114	183
307	113	181
308	114	183
309	119	194
310	114	183
311	119	194

312	117	189
313	125	205
314	126	207
315	131	215
316	128	210
317	127	209
318	152	236
319	174	246
320	175	246
321	210	251
322	208	251
323	207	251
324	208	251
325	209	251
326	208	251
327	210	251
328	208	251
329	207	251
330	211	251
331	210	251
332	176	246
333	174	246
334	154	237
335	130	213
336	128	210
337	126	207
338	127	209
339	127	209
340	131	215
341	124	204
342	118	192
343	116	187
344	118	192
345	117	189
346	118	192
347	117	189
348	118	192
349	128	210
350	120	196
351	117	189
352	117	189
353	116	187

354	117	189
355	116	187
356	129	212
357	126	207
358	127	209
359	174	246
360	177	247
361	210	251
362	209	251
363	208	251
364	207	251
365	208	251
366	207	251
367	207	251
368	208	251
369	207	251
370	207	251
371	207	251
372	207	251
373	207	251
374	208	251
375	207	251
376	207	251
377	208	251
378	160	241
379	158	239
380	208	251
381	157	239
382	209	251
383	209	251
384	208	251
385	207	251
386	207	251
387	208	251
388	209	251
389	208	251
390	208	251
391	209	251
392	208	251
393	209	251
394	209	251
395	209	251

396	188	249
397	189	249
398	184	248
399	174	246
400	145	230
401	128	210
402	126	207
403	129	212
404	126	207
405	126	207
406	128	210
407	132	216
408	129	212
409	132	216
410	126	207
411	121	198
412	119	194
413	118	192
414	117	189
415	118	192
416	119	194
417	117	189
418	118	192
419	117	189
420	117	189
421	118	192
422	118	192
423	128	210
424	116	187
425	126	207
426	129	212
427	117	189
428	122	200
429	117	189
430	116	187
431	118	192
432	42	91
433	117	189
434	119	194
435	25	89
436	21	89
437	21	89

438	20	89
439	20	89
440	20	89
441	21	89
442	116	187
443	129	212
444	128	210
445	117	189
446	116	187
447	121	198
448	117	189
449	118	192
450	118	192
451	118	192
452	119	194
453	118	192
454	121	198
455	127	209
456	129	212
457	174	246
458	209	251
459	210	251
460	207	251
461	208	251
462	208	251
463	209	251
464	208	251
465	208	251
466	208	251
467	209	251
468	209	251
469	208	251
470	208	251
471	208	251
472	208	251
473	208	251
474	208	251
475	208	251
476	208	251
477	209	251
478	211	251
479	213	251

480	171	245
481	126	207
482	0	89
483	114	183
484	118	192
485	114	183
486	112	179
487	116	187
488	112	179
489	115	185
490	116	187
491	113	181
492	124	204
493	114	183
494	112	179

APPENDIX H EXPERIMENT 2 HEADING SELECTION SCAN RESULT

Heading Direction (°)	Distance (cm)
0	82
10	81
20	73
30	324
40	329
50	44
60	43
70	42
80	42
90	42
100	42
110	43
120	71
130	68
140	69
150	68
160	72
170	275
180	327

APPENDIX I EXPERIMENT 3 FLC BEHAVIOR ON TILES SURFACE

No	Pitch (°)	Distance (cm)	Motor Speed (PWM)	Time Taken (s)
1	16.89	123	92	0.319
2	30.57	122	98	0.629
3	41.49	121	131	0.94
4	50.43	122	131	1.25
5	57.27	121	133	1.56
6	62.77	122	132	1.87
7	66.92	121	144	2.181
8	70.59	123	153	2.491
9	73.4	122	159	2.801
10	75.87	122	159	3.111
11	77.85	113	147	3.421
12	79.3	123	159	3.732
13	80.55	121	159	4.042
14	81.68	122	158	4.353
15	82.22	121	158	4.663
16	82.99	122	158	4.973
17	83.39	108	134	5.282
18	83.84	123	157	5.593
19	84.08	121	158	5.903
20	84.48	122	158	6.213
21	84.52	121	158	6.523
22	84.74	122	158	6.834

اوتنور ستي تكنيكل مليسيا ملاك
APPENDIX JEXPERIMENT 3 FLC BEHAVIOR ON ASPHALT SURFACE

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

No	Pitch (°)	Distance (cm)	Motor Speed (PWM)	Time Taken (s)
1	16.87	122	92	0.319
2	30.8	122	100	0.629
3	41.49	121	131	0.94
4	50.37	122	131	1.25
5	57.87	121	132	1.56
6	63.77	121	135	1.87
7	67.4	113	146	2.18
8	71.39	122	156	2.49
9	74.21	122	159	2.8
10	76.89	122	159	3.11
11	77.8	121	159	3.421
12	80.12	122	159	3.732

13	79.8	122	159	4.042
14	80.19	122	158	4.353
15	81.63	113	147	4.662
16	82.69	121	158	4.972
17	82.44	122	158	5.282
18	84.76	121	158	5.594
19	83.31	121	158	5.904
20	85.27	114	150	6.213
21	83.62	121	158	6.523
22	85.49	122	158	6.834

APPENDIX K EXPERIMENT 3 FLC BEHAVIOR ON ROCKY SURFACE

No	Pitch (°)	Distance (cm)	Motor Speed (PWM)	Time Taken (s)
1	17.13	124	92	0.319
2	30.67	123	98	0.629
3	41.8	121	131	0.94
4	52.18	122	133	1.25
5	60.87	122	130	1.56
6	62.87	122	132	1.87
7	64.72	121	138	2.181
8	73.3	123	159	2.491
9	73.41	122	159	2.801
10	75.03	122	159	3.111
11	82.19	121	158	3.423
12	82.64	122	158	3.733
13	77.01	121	159	4.043
14	81.68	122	158	4.354
15	88.24	122	158	4.664
16	87.7	122	158	4.974
17	84.56	121	158	5.284
18	82.99	41	91	5.587
19	86.53	122	158	5.898
20	89.97	122	158	6.208
21	88.46	122	158	6.519
22	89.14	123	157	6.83
23	95.19	121	198	7.14
24	91.88	122	200	7.45
25	95.29	121	198	7.76
26	95.43	122	200	8.071

27	92.17	123	202	8.381
----	-------	-----	-----	-------

APPENDIX L EXPERIMENT 3 FLC BEHAVIOR ON HILL SURFACE

No	Pitch (°)	Distance (cm)	Motor Speed (PWM)	Time Taken (s)
1	16.72	73	90	0.315
2	31.42	72	90	0.62
3	42.44	72	91	0.925
4	52.2	72	90	1.23
5	57.48	71	90	1.536
6	62.23	72	90	1.84
7	57.27	74	90	2.145
8	53.01	77	89	2.45
9	49.85	80	89	2.756
10	49.09	87	133	3.064
11	49.06	89	133	3.373
12	49.74	92	133	3.681
13	49.38	93	133	3.989
14	49.17	100	133	4.298
15	48.4	101	132	4.606
16	47.24	99	130	4.916
17	48.25	103	131	5.225
18	49.25	101	132	5.534
19	49.81	104	133	5.842
20	49.69	98	133	6.152
21	49.55	106	132	6.461
22	49.76	100	133	6.77
23	49.36	108	131	7.079
24	49.53	100	133	7.388
25	49.43	100	133	7.697
26	49.83	103	133	8.006
27	52.99	101	137	8.314
28	53.68	105	137	8.624
29	56.89	105	136	8.933
30	65.2	102	130	9.242
31	77.49	93	123	9.55
32	83.27	80	89	9.857
33	83.9	90	122	10.165
34	83.8	91	122	10.473
35	83.88	91	122	10.781

36	83.97	91	122	11.089
37	83.98	91	122	11.398
38	84	91	122	11.706
39	83.96	91	122	12.014
40	84.03	90	122	12.322
41	84.12	91	122	12.632
42	84.17	91	122	12.94
43	84.13	91	122	13.248
44	84.15	91	122	13.556
45	84.21	91	122	13.864
46	84.25	91	122	14.173
47	84.34	90	122	14.481

APPENDIX M MAIN PROGRAMMING CODE FOR MOTION CONTROL

```

#include <Servo.h>
#include "L298N.h"
#include "HCSR04.h"
#include "ADXL345.h"
#include "ITG3200.h"
#include "fis_header.h"

Motor ML(6,7,8);
Motor MR(13,12,11);
HCSR04 UsSen(9,10);

Servo servo;
//---IMU Raw Variable
double Ax, Ay, Az;
float Gx, Gy, Gz, LGx, LGy;
unsigned long lastTime;
float dt;

//--- Filtered Variable

float AngleX, AngleY, AngleZ;
float LastAngleX, LastAngleY, LastAngleZ;

//---- HCSR04 variable
int dist;
long pulse;

//--- Motor variable
int p; // motor speed in pwm pulses 0-255

```

```
// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 1;
// Number of rules to the fuzzy inference system
const int fis_gcR = 15;
```

```
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
```

```
void setup()
```

```
{
  ADXL.init();
  ITG.init();
```

```
  servo.attach(3);
  servo.write(80);
```

```
  Serial.begin(9600);
```

```
  LastAngleX = 0.0;
  LastAngleY = 0.0;
  LastAngleZ = 0.0;
  lastTime = millis();
  LGy=90;
```

```
}
```

```
void loop()
```

```
{
  AngleRead(AngleX, AngleY, AngleZ);
  UsSen.scan(dist,pulse);
```

```
  //---Update Fuzzy Logic input/output data
  g_fisInput[0] = AngleX;
  g_fisInput[1] = dist;
  g_fisOutput[0] = 0;
```

```
  if (dist>=60)
```

```
  {
    fis_evaluate();
    p = g_fisOutput[0]; // Set output value: Speed
    if(p>=255)
    {
      p=255;
    }
    else if(p<=80)
```

```

    {
        p=80;
    }
}
else if(dist<60)
{
    p=0;
    //do LIDAR scanning subroutine
}

ML.CCW(p);
MR.CCW(p);

Serial.print(AngleX);
Serial.print("\t");
Serial.print(dist);
Serial.print("\t");
Serial.println(p);
}

void AngleRead(float &X, float &Y, float &Z)
{
    ADXL.scan(Ax, Ay, Az);
    ITG.scan(Gx,Gy,Gz,dt,lastTime);

    X = 0.8*(LastAngleX+Gx)+0.2*(Ax);
    Y = 0.8*(LastAngleY+Gy)+0.2*(Ay);
    Z = 0.8*(LastAngleZ+Gz)+0.2*(Az);

    LastAngleX = X;
    LastAngleY = Y;
    LastAngleZ = Z;
}

//*****
// Support functions for Fuzzy Inference System
//*****
// Z-shaped Member Function
FIS_TYPE fis_zmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1];
    FIS_TYPE m = ((a + b) / 2.0);
    FIS_TYPE t = (b - a);
    if (x <= a) return (FIS_TYPE) 1;
    if (x <= m)

```

```

{
    t = (x - a) / t;
    return (FIS_TYPE) (1.0 - (2.0 * t * t));
}
if (x <= b)
{
    t = (b - x) / t;
    return (FIS_TYPE) (1.0 - (2.0 * t * t));
}
return (FIS_TYPE) 0;
}

```

// Trapezoidal Member Function

FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)

```

{
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];
    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d - x) / (d - c)) : 0)));
    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x - a) / (b - a)) : 0)));
    return (FIS_TYPE) min(t1, t2);
}

```

// S-Shaped membership function

FIS_TYPE fis_smf(FIS_TYPE x, FIS_TYPE* p)

```

{
    FIS_TYPE a = p[0], b = p[1];
    FIS_TYPE m = ((a + b) / 2.0);
    FIS_TYPE t = (b - a);
    if (a >= b) return (FIS_TYPE) (x >= m);
    if (x <= a) return (FIS_TYPE) 0;
    if (x <= m)
    {
        t = (x - a) / t;
        return (FIS_TYPE) (2.0 * t * t);
    }
    if (x <= b)
    {
        t = (b - x) / t;
        return (FIS_TYPE) (1.0 - (2.0 * t * t));
    }
    return (FIS_TYPE) 1;
}

```

// Generalized Bell Member Function

FIS_TYPE fis_gbellmf(FIS_TYPE x, FIS_TYPE* p)

```

{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t = (x - c) / a;

```

```

    if ((t == 0) && (b == 0)) return (FIS_TYPE) 0.5;
    if ((t == 0) && (b < 0)) return (FIS_TYPE) 0;
    return (FIS_TYPE) (1.0 / (1.0 + pow(t, b)));
}

FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return min(a, b);
}

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size, _FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;

    if (size == 0) return ret;
    if (size == 1) return array[0];

    ret = array[0];
    for (i = 1; i < size; i++)
    {
        ret = (*pfnOp)(ret, array[i]);
    }

    return ret;
}

//*****
// Data for Fuzzy Inference System
//*****
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =
{
    fis_zmf, fis_trapmf, fis_smf, fis_gbellmf
};

// Count of member function for each Input
int fis_gIMFCount[] = { 5, 3 };

// Count of member function for each Output
int fis_gOMFCount[] = { 3 };

```

```

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { 20, 90 };
FIS_TYPE fis_gMFI0Coeff2[] = { 60, 80, 100, 120 };
FIS_TYPE fis_gMFI0Coeff3[] = { 90, 160 };
FIS_TYPE fis_gMFI0Coeff4[] = { 30, 50, 70, 90 };
FIS_TYPE fis_gMFI0Coeff5[] = { 90, 110, 130, 150 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2,
fis_gMFI0Coeff3, fis_gMFI0Coeff4, fis_gMFI0Coeff5 };
FIS_TYPE fis_gMFI1Coeff1[] = { 20, 85 };
FIS_TYPE fis_gMFI1Coeff2[] = { 31.26, 3.278, 85 };
FIS_TYPE fis_gMFI1Coeff3[] = { 85, 150 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2, fis_gMFI1Coeff3
};
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff };

// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { 80, 167.5 };
FIS_TYPE fis_gMFO0Coeff2[] = { 40, 2, 167.5 };
FIS_TYPE fis_gMFO0Coeff3[] = { 167.5, 255 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2,
fis_gMFO0Coeff3 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff };

// Input membership function set
int fis_gMFI0[] = { 0, 1, 2, 1, 1 };
int fis_gMFI1[] = { 0, 3, 2 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1 };

// Output membership function set
int fis_gMFO0[] = { 0, 3, 2 };
int* fis_gMFO[] = { fis_gMFO0 };

// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Inputs
int fis_gRI0[] = { 2, 3 };
int fis_gRI1[] = { 4, 3 };
int fis_gRI2[] = { 5, 3 };
int fis_gRI3[] = { 3, 3 };
int fis_gRI4[] = { 1, 3 };
int fis_gRI5[] = { 2, 2 };
int fis_gRI6[] = { 4, 2 };
int fis_gRI7[] = { 5, 2 };
int fis_gRI8[] = { 1, 2 };

```

```

int fis_gRI9[] = { 3, 2 };
int fis_gRI10[] = { 2, 1 };
int fis_gRI11[] = { 4, 1 };
int fis_gRI12[] = { 5, 1 };
int fis_gRI13[] = { 1, 1 };
int fis_gRI14[] = { 3, 1 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4, fis_gRI5, fis_gRI6,
fis_gRI7, fis_gRI8, fis_gRI9, fis_gRI10, fis_gRI11, fis_gRI12, fis_gRI13, fis_gRI14 };

```

```
// Rule Outputs
```

```

int fis_gRO0[] = { 3 };
int fis_gRO1[] = { 2 };
int fis_gRO2[] = { 2 };
int fis_gRO3[] = { 1 };
int fis_gRO4[] = { 1 };
int fis_gRO5[] = { 2 };
int fis_gRO6[] = { 2 };
int fis_gRO7[] = { 2 };
int fis_gRO8[] = { 1 };
int fis_gRO9[] = { 1 };
int fis_gRO10[] = { 1 };
int fis_gRO11[] = { 1 };
int fis_gRO12[] = { 1 };
int fis_gRO13[] = { 1 };
int fis_gRO14[] = { 1 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5,
fis_gRO6, fis_gRO7, fis_gRO8, fis_gRO9, fis_gRO10, fis_gRO11, fis_gRO12,
fis_gRO13, fis_gRO14 };

```

```
// Input range Min
```

```
FIS_TYPE fis_gIMin[] = { 0, 0 };
```

```
// Input range Max
```

```
FIS_TYPE fis_gIMax[] = { 180, 250 };
```

```
// Output range Min
```

```
FIS_TYPE fis_gOMin[] = { 20 };
```

```
// Output range Max
```

```
FIS_TYPE fis_gOMax[] = { 300 };
```

```
/**

```

```
// Data dependent support functions for Fuzzy Inference System
```

```
*/

```

```
FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x, int o)
```

```
{
    FIS_TYPE mfOut;
    int r;
```

```

for (r = 0; r < fis_gcR; ++r)
{
    int index = fis_gRO[r][o];
    if (index > 0)
    {
        index = index - 1;
        mfOut = (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
    }
    else if (index < 0)
    {
        index = -index - 1;
        mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
    }
    else
    {
        mfOut = 0;
    }

    fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);
}
return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);
}

FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet, int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) / (FIS_RESOLUTION - 1);
    FIS_TYPE area = 0;
    FIS_TYPE momentum = 0;
    FIS_TYPE dist, slice;
    int i;

    // calculate the area under the curve formed by the MF outputs
    for (i = 0; i < FIS_RESOLUTION; ++i){
        dist = fis_gOMin[o] + (step * i);
        slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
        area += slice;
        momentum += slice*dist;
    }

    return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) : (momentum / area));
}

//*****
// Fuzzy Inference System
//*****
void fis_evaluate()
{

```

```

FIS_TYPE fuzzyInput0[] = { 0, 0, 0, 0, 0 };
FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };
FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };
FIS_TYPE fuzzyOutput0[] = { 0, 0, 0 };
FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, };
FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
FIS_TYPE sW = 0;

// Transforming input to fuzzy Input
int i, j, r, o;
for (i = 0; i < fis_gcI; ++i)
{
    for (j = 0; j < fis_gIMFCount[i]; ++j)
    {
        fuzzyInput[i][j] =
            (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i], fis_gMFICoeff[i][j]);
    }
}

int index = 0;
for (r = 0; r < fis_gcR; ++r)
{
    if (fis_gRType[r] == 1)
    {
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
    else
    {
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
        }
    }
}

```

```

        else
            fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
        }
    }

    fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
    sW += fuzzyFires[r];
}

if (sW == 0)
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
    }
}
else
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet, o);
    }
}
}

```

APPENDIX N HEADER FILE FOR FLC INFERENCE SYSTEM

```

#define FIS_TYPE float
#define FIS_RESOLUTION 101
#define FIS_MIN -3.4028235E+38
#define FIS_MAX 3.4028235E+38
typedef FIS_TYPE(*_FIS_MF)(FIS_TYPE, FIS_TYPE*);
typedef FIS_TYPE(*_FIS_ARR_OP)(FIS_TYPE, FIS_TYPE);
typedef FIS_TYPE(*_FIS_ARR)(FIS_TYPE*, int, _FIS_ARR_OP);

```

APPENDIX O SOURCE CODE FOR ACCELEROMETER

```

#include "Arduino.h"
#include "I2C.h"
#include "ADXL345.h"

void ADXL::init()
{

```

```

I2C.init();

//--- Configure Accelerometer ADXL345
I2C.write(ADXAddress, BW_Rate, 0x0A); //Set Sampling rate to 100Hz
I2C.write(ADXAddress, Power_Register, 8); //Enabling measuring mode
delay(10);
}

void ADXL::scan(double &X, double &Y, double &Z)
{
int xRaw, yRaw, zRaw;
double Rx, Ry, Rz, Racc, AccScale = 256.0;

xRaw = I2C.read(ADXAddress, ACC_XOUT_H)<<8; // X-axis
xRaw |= I2C.read(ADXAddress, ACC_XOUT_L);
yRaw = I2C.read(ADXAddress, ACC_YOUT_H)<<8; // Y-Axis
yRaw |= I2C.read(ADXAddress, ACC_YOUT_L);
zRaw = I2C.read(ADXAddress, ACC_ZOUT_H)<<8; // Z-Axis
zRaw |= I2C.read(ADXAddress, ACC_ZOUT_L);

//---convert to g
Rx = xRaw/AccScale;
Ry = yRaw/AccScale;
Rz = zRaw/AccScale;

//---find angle between resultant and axes
Racc = sqrt(pow(Rx,2) + pow(Ry,2) + pow(Rz,2));
X = (acos(Rx/Racc)*180)/3.14;
Y = (acos(Ry/Racc)*180)/3.14;
Z = (acos(Rz/Racc)*180)/3.14;
}

```

APPENDIX P HEADER FILE FOR ACCELEROMETER

```

#ifndef ADXL345_h
#define ADXL345_h
#include "Arduino.h"

//--- Accelerometer Register Addresses
#define ADXAddress 0x53 // Sensor address
#define BW_Rate 0x2C // bandwidth rate table 7 datasheet
#define Power_Register 0x2D // power control
#define ACC_XOUT_L 0x32 // Xout low
#define ACC_XOUT_H 0x33 // Xout high
#define ACC_YOUT_L 0x34 // Yout low
#define ACC_YOUT_H 0x35 // Yout high

```

```
#define ACC_ZOUT_L 0x36 // Zout low
#define ACC_ZOUT_H 0x37 // Zout high
```

```
class ADXL
{
public:
    void init();
    void scan(double &X, double &Y, double &Z);

private:

};

extern ADXL ADXL;

#endif
```

APPENDIX Q SOURCE CODE FOR ULTRASONIC

```
#include "Arduino.h"
#include "HCSR04.h"
```

```
HCSR04::HCSR04(int trig, int echo)
{
    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
    _trig = trig;
    _echo = echo;
}
```

```
void HCSR04::scan(int &dist, long &pulse)
{
    digitalWrite(_trig, LOW);
    delayMicroseconds(2);
    digitalWrite(_trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(_trig, LOW);
    pulse = pulseIn(_echo, HIGH);
    dist= pulse*0.034/2;
}
```

APPENDIX R HEADER FILE FOR ULTRASONIC

```

#ifndef HCSR04_h
#define HCSR04_h

#include "Arduino.h"

class HCSR04
{
public:
    HCSR04(int trig, int echo);
    void scan(int &dist, long &pulse);

private:
    int _trig;
    int _echo;
};

#endif

```

APPENDIX S SOURCE CODE FOR I2C



```

#include "I2C.h"
#include "Arduino.h"

#include <Wire.h>

void I2C::init()
{
    Wire.begin();
}

void I2C::write(char address, char registerAddress, char data)
{
    Wire.beginTransmission(address);
    Wire.write(registerAddress);
    Wire.write(data);
    Wire.endTransmission();
}

unsigned char I2C::read(char address, char registerAddress)
{
    unsigned char data=0;
    Wire.beginTransmission(address);
    Wire.write(registerAddress);
    Wire.endTransmission();
    Wire.beginTransmission(address);
    Wire.requestFrom(address, 1);
}

```

```

if(Wire.available())

{
  data = Wire.read();
}

Wire.endTransmission();
return data;
}

```

APPENDIX T HEADER FILE FOR I2C

```

#ifndef I2C_h
#define I2C_h

#include "Arduino.h"

class I2C
{
public:
  void init();
  void write(char address, char registerAddress, char data);
  unsigned char read(char address, char registerAddress);
};

extern I2C I2C;

#endif

```



APPENDIX U SOURCE CODE FOR GYROSCOPE

```

#include "Arduino.h"
#include "I2C.h"
#include "ITG3200.h"

void ITG::init()
{
  I2C.init();
  //--- Configure Accelerometer ADXL345
  char DLPF_CFG = 1;
  char DLPF_FS_SEL = 3<<3;
}

```

```

I2C.write(ITGAddress, DLPF_FS, (DLPF_FS_SEL|DLPF_CFG)); //Set the gyroscope
Full Scale Selection, Low Pass Filter and Internal Rate Sampling Configuration
I2C.write(ITGAddress, SMPLRT_DIV, 9); //Set the sample rate divider to 100 hz
}

```

```

void ITG::scan(float &Gx, float &Gy, float &Gz, float &dt, unsigned long &IT)

```

```

{
int Gx_Raw, Gy_Raw, Gz_Raw;
float RxGyro, RyGyro, RzGyro, GyroFactor = 14.375;
unsigned long now;

//--- Gyroscope Reading
Gx_Raw =I2C.read(ITGAddress, GYRO_XOUT_H)<<8; // X-axis
Gx_Raw |=I2C.read(ITGAddress, GYRO_XOUT_L);
Gy_Raw =I2C.read(ITGAddress, GYRO_YOUT_H)<<8; // y-axis
Gy_Raw |=I2C.read(ITGAddress, GYRO_YOUT_L);
Gz_Raw =I2C.read(ITGAddress, GYRO_ZOUT_H)<<8; // z-axis
Gz_Raw |=I2C.read(ITGAddress, GYRO_ZOUT_L);

```

```

now = millis();
dt = now - IT;
IT = now;

```

```

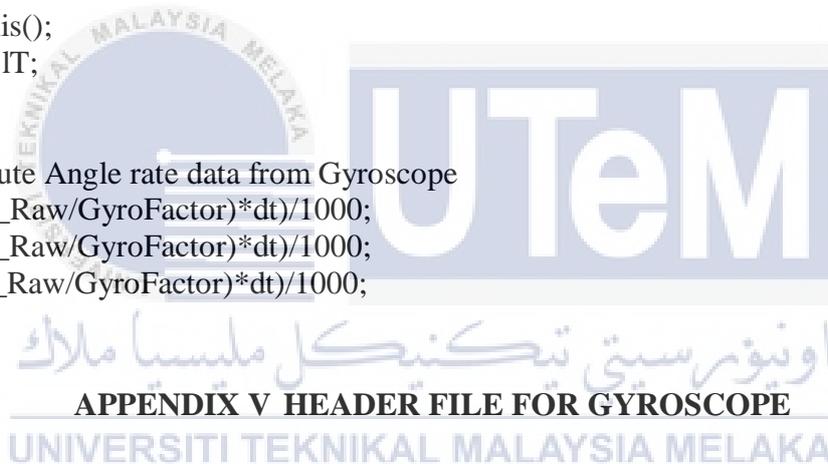
//--- Compute Angle rate data from Gyroscope
Gx = ((Gx_Raw/GyroFactor)*dt)/1000;
Gy = ((Gy_Raw/GyroFactor)*dt)/1000;
Gz = ((Gz_Raw/GyroFactor)*dt)/1000;

```

```

}

```



APPENDIX V HEADER FILE FOR GYROSCOPE

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

```

#ifndef ITG3200_h
#define ITG3200_h
#include "Arduino.h"

```

```

//--- Gyroscope ITG 3200 Register Addresses

```

```

#define ITGAddress 0x68
#define SMPLRT_DIV 0x15
#define DLPF_FS 0x16
#define GYRO_XOUT_H 0x1D
#define GYRO_XOUT_L 0x1E
#define GYRO_YOUT_H 0x1F
#define GYRO_YOUT_L 0x20
#define GYRO_ZOUT_H 0x21
#define GYRO_ZOUT_L 0x22

```

```

class ITG

```

```

{
public:
    void init();
    void scan(float &Gx, float &Gy, float &Gz, float &dt, unsigned long &IT);

private:

};

extern ITG ITG;

#endif

```

APPENDIX W SOURCE CODE FOR MOTOR CONTROLLER

```

#include "Arduino.h"
#include "L298N.h"

Motor::Motor(int IN1,int IN2,int PWM)
{
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(PWM, OUTPUT);
    _in1 = IN1;
    _in2 = IN2;
    _pwm = PWM;
}

void Motor::CW(int Speed)
{
    digitalWrite(_in1, HIGH);
    digitalWrite(_in2, LOW);
    analogWrite(_pwm, Speed);
}

void Motor::CCW(int Speed)
{
    digitalWrite(_in1, LOW);
    digitalWrite(_in2, HIGH);
    analogWrite(_pwm, Speed);
}

```

APPENDIX X HEADER FILE FOR MOTOR CONTROLLER

```
#ifndef L298N_h
#define L298N_h

#include "Arduino.h"

class Motor
{
public:
    Motor(int IN1,int IN2,int PWM);
    void CW(int Speed);
    void CCW(int Speed);
private:
    i
t_in1;
    int _in2;
    int _pwm;
};

#endif
```

