STAIR CLIMBING MOTION CONTROL FOR PORTABLE MECHANISM

NURUL NARINA BINTI MOHAMAD HANAPI

BACHELOR OF MECHATRONICS ENGINEERING WITH HONORS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

STAIR CLIMBING MOTION CONTROL FOR PORTABLE MECHANISM

NURUL NARINA BINTI MOHAMAD HANAPI

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

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled "STAIR CLIMBING MOTION CONTROL FOR PORTABLE MECHANISM 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 :

APPROVAL

I hereby declare that I have checked this report entitled “STAIR CLIMBING MOTION CONTROL FOR PORTABLE MECHANISM” 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

I wish to give my sincere thanks to the Faculty of Electrical Engineering for providing facilities which required for accomplish this project. Besides, I am also took this opportunity to give my special gratitude to my friends and mostly my beloved parents by giving their encouragement, support, time and idea throughout various phases for the completion of this project.

ACKNOWLEDGEMENTS

First of all, I would like to praise be to Allah for the health and well-being which had been given to me in order to complete this writing of project thesis.

Next, I would like to take this opportunity to express my sincere acknowledgement to my supervisor, Prof. Madya Dr. Mariam Binti Md Ghazaly from Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka (UTeM) for giving assistance, suggestion and guidance towards the completion of this thesis. Without my supervisor, it would be difficult for me to accomplish this entire project.

In addition, special thanks are dedicated to my family, peers, members and the technician team from the Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka for their moral support in completing this degree.

Lastly, my sincere thanks are dedicated to everyone who have helped me directly or indirectly in completing this study.

ABSTRACT

A wheel-leg mobile robot has the capability to move on flat surfaces and climb the stairs. In this project, a Delta-Wheel stair climbing robot is a purpose to help elderly and disabled people. This is because elderly and disabled people facing difficulties in performing stairs climbing and moving on flat surfaces in daily activities as normal people. These people have required an assistant to help them to climb the stairs. However, the control system of the stair climbing is not a big problem but to maintain stability and smoothness of the stair climbing robot it will become more challenges The main objectives of this project is to design Fuzzy Logic Controller (FLC) by using MATLAB Toolbox in MATLAB software for stair climbing mechanism. The performance of the FLC will be evaluated in term of obstacle detection and obstacles avoidance. The FLC is design by using two inputs and two outputs. The two inputs are left sensor and right sensor that measure distance of obstacles from the robot. The ouputs are speed movement of left motor and right motor that driven by two DC motors. The Infrared sensor are placed at front corners of the robot. The IR sensors can detect up to 0 cm to 27 cm at 90 degrees. The sensors is required to detect the obstacles and it activates the controller to avoid the obstacles. If the robot able to avoid the obstacles, the system will read another sensor input for next process. Based on this signal, the FLC control the speed of the left motor and right motor respectively. The robot able to moving forward with fast speed of left motor and right motor when sensor not detected the obstacle at front of the robot. As the results, the robot able to move forward and reverse but fail to make a turn because the structure of the tires.

ABSTRAK

Robot mudah alih beroda mempunyai keupayaan untuk bergerak di atas permukaan yang rata dan memanjat tangga. Dalam projek ini, robot mendaki tangga yang diberi nama Roda Delta adalah bertujuan untuk membantu golongan warga tua dan golongan orang kurang upaya. Hal ini kerana, orang tua dan orang kurang upaya menghadapi kesukaran untuk aktiviti mendaki tangga dan bergerak diatas permukaan rata dalam kehidupan seharian seperti orang biasa. Golongan ini akan memerlukan seseorang untuk membantu mereka ketika mendaki tangga. Walau bagaimanapun, sistem kawalan pendakian tangga bukanlah satu masalah yang besar tetapi untuk mengekalkan kestabilan dan kelancaran robot ketika menaiki tangga, ia akan menjadi lebih mencabar. Objektif utama untuk projek ini adalah untuk mereka bentuk Pengawal Logik Kabur (PLK)" dengan menggunakan kotak peralatan MATLAB di dalam MATLAB untuk mekanisme pendakian tangga. Prestasi FLC akan dinilai dari segi keberkesanan untuk mengesan halangan dan menghindar halangan. PLK direka dengan menggunakan dua kemasukant dan dua pengeluaran. Kedua-dua kemasukan ialah penderia kiri dan penderia kanan yang mengukur jarak halangan dari robot. Pengeluaran adalah pergerakan kelajuan motor kiri dan motor kanan yang didorong oleh dua motor DC. Penderia inframerah diletakkan di sudut depan robot. Penderia IR boleh mengesan sehingga 0 cm hingga 27 cm pada 90 darjah. Penderia diperlukan untuk mengesan rintangan dan ia mengaktifkan pengawal untuk mengelakkan rintangan. Jika robot dapat mengelakkan rintangan, sistem akan membaca kemasukan penderia lain untuk proses seterusnya. Berdasarkan isyarat ini, PLK mengawal kelajuan motor kiri dan motor kanan masing-masing. Robot mampu bergerak ke hadapan dengan kelajuan pantas motor kiri dan motor kanan apabila penderia tidak mengesan halangan di hadapan robot. Secara keputusannya, robot dapat bergerak ke hadapan dan mengundur tetapi gagal untuk berpusing kerana struktur tayar.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

IR	- InfraRed
FLC	- Fuzzy Logic Controller
MISO	- Multiple Input Single Output
MIMO	- Multiple Input Multiple Output
US	- United States
PID	- Proportional Integral Derivative
LRF	- Laser range finder
ANN	- Artificial neural network
ANFIS	- Adaptive neuro fuzzy inference system
DC	- Direct current
PI	- Proportional Integral
PD	- Proportional Derivative
PB	- Positive Big
PS	- Positive Small
Z	- Zero
NB	- Negative Big
NS	- Negative Small
Kp	- Proportional term
Kd	- Derivative term
Ki	- Integral term
PWM	- Pulse Width Modulation
ZMF	- Z-Shaped Membership Functions
SIGMF	- Sigmoidal Membership Functions
SMF	- S-Shaped Membership Functions

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

A stairway is a kind of building structure and consider as an obstacle and challenging for vehicles and mobile robots when an emergency occurs such as disaster, fire, and earthquake, especially for wheelchair users. Elderly and disabled people have used the wheelchair in their daily life to move from one destination to another destination. Unfortunately, the number of disabled people are increasing day by day due to some issues such as road accident or disease as well as a number of wheelchair users. Elderly and disabled people are facing difficulties in performing stairs climbing in daily activities as normal people. Nowadays, there are various type of stair climbing mechanism are presented for wheelchair and the design are depending on adaptability on stair climbing and traverse obstacle or slope. Furthermore, the stair climbing robot also can be useful where the stair climbing robot can replace human assistant in the stair climbing operation for wheelchair users.

Generally, stair climbing robot having different locomotive mechanisms based on their performance such as wheeled, legged, tracked and hybrid locomotion. The wheeled locomotive mechanism having the simplest design and able to move faster on flat terrain and provided high energy efficiency. However, in the different unstructured environment their movement is very difficult because it depends on surroundings and different sizes of obstacles during mobility.

A robot with a legged locomotive mechanism capable to climb the stairs and traverse obstacle on different terrain and make it stable with that locomotive mechanisms. Design of legged robot is quite complex and not provide smooth motion on flat terrain compared to wheeled robots. The operation of the legged robots is considering as human legs during up and down stairs.

Furthermore, stair climbing robot with wheel-leg locomotive mechanism has better performance because it will able to moving on flat surfaces and climb the stairs properly. In this project, Delta-Wheel structure is used based on a wheel-leg mechanism in order to move on different structure environment by changing their locomotion based on requirement. Even though, the leg-wheel mechanisms able to adapt in many kind environments it also shows the complex structure and required control algorithm to perform the movement.

Then, Fuzzy Logic Controller (FLC) by using multiple input multiple output (MIMO) was designed in this stair climbing robot to control the complex system and improve the system performance in term of stability and smooth motion. All the system performance of the stair climbing robot are examined.

1.2 Motivation

In recent years, the researcher has involved the attention of people around the world by replacing the manual wheelchair with an automatic wheelchair which able to climb the stairs. The stair climbing robot can be used not only for stair climbing, but it also can be implemented on flat surfaces. Furthermore, the stair climbing robot also play an important role for disabled people in term of healthcare to make their daily life more comfortable.

According to the Disabilities Statistics Annual Report for United State (US) 2016 [1], there are 12.6% of the population in the US are disabled people. Every year the number of disabled people are increasing due to various issue from various age. Figure 1.1 shows the statistics of disabled people from 2008 to 2015 based on several types of questions during a community survey. One of the question from six disability question, it is difficult for disabled people with physically disabled to walking and climbing stairs in daily life.

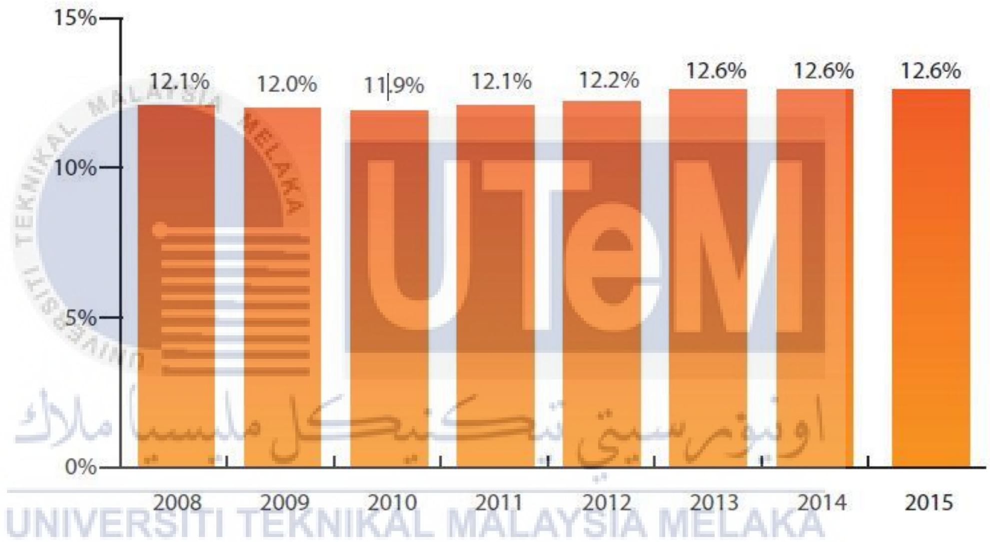

Figure 1.1: Percentage of People in the US with Disabilities, 2008-2015 [1]

On the other hand, the most disabled people being reported is ambulatory with 6.6% and closely followed by cognitive with 4.8% and independent living with 4.5%. Table 1.1 shows the most common type of disabilities in the United State from 2008 to 2015 in percentage

Table 1.1: People with Disabilities by Type and Year, 2008-2015 [1]

Type of disabilities	**Percentage (%)**							
	2008	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**	**2015**
Hearing	3.5	3.4	3.4	3.4	3.4	3.6	3.6	3.6
Vision	2.3	2.1	2.1	2.2	2.2	2.3	2.3	2.3
Cognitive	4.5	4.5	4.5	4.6	4.6	4.7	4.8	4.8
Ambulatory	6.4	6.4	6.4	6.5	6.5	6.6	6.7	6.6
Self-care	2.4	2.4	2.4	2.5	2.5	2.5	2.5	2.5
Independent living	4.3	4.2	4.3	4.4	4.4	4.4	4.5	4.5

So, the best solution in order to decrease the number of human assistant during climbing stairs for disabled people especially for physically disabled is by replacing the human assistant with a stair-climbing wheelchair. Nowadays, there are various type of wheelchairs, for example, manual wheelchair and automatic wheelchair. However, wheelchair out there is not affordable for poverty wheelchair users and some wheelchair are not suitable for other purposes. Furthermore, the stair climbing wheelchair that able to adapt with different unstructured of environment likes stairs, curb and zebra crossing give an advantage for wheelchair users in daily life.

1.3 Problem Statement

Stairs represent a dangerous component in building structure compared to slopes. These stairs are considering as an obstacle to the mobile robot to perform their movement, especially on stair climbing. Hence, stairs will become a major problem for disabled people and elderly who use the wheelchair in daily life.

Elderly and disabled people have required an assistant to help them to climb the stairs, however, this technique is not efficient in term of safety. For example, by carrying disabled or older people on backs, this technique is very good and less cost required but the risk to injure for both people is very high. On the other hand, it is not suitable for the long-term because it can injure the backbone. Another technique is by lifting a wheelchair user but it will require more assistant especially if the weight of a wheelchair user is more than the weight of assistant. Then, it will require from three to four assistant to help wheelchair users for ascending and descending stairs.

However, a suitable type of mechanisms will become a serious issue in order to design a stair climbing robot that able to move on flat surfaces and able to adapt in any kind of unstructured environment as well as traverse obstacle. A stair climbing robot is very useful for wheelchairs users which the stair climbing robots can be used instead of using the human assistant. The control system of the stair climbing is not a big problem but to maintain stability and smoothness of the stair climbing robot it will become more challenges. Hence, it required a good controller so that the stair climbing robot can maintain stability and provide smooth motion during movement.

1.4 Objectives

The objectives of this project are: -

i. To develop open loop test for obstacle detection, obstacle avoidance and stair climbing..
ii. To design Fuzzy Logic Controller for stair climbing mechanism.
iii. To analyze performance of the mechanism in terms of obstacles avoidance.

1.5 Scope and Limitations

The scope of this project are: -

i. The Fuzzy Logic Controller of the stair climbing robot is developed using Arduino.
ii. Infrared sensor is used for obstacle detection and it only able to detect from range 0 – 27 cm.
iii. The stair climbing robot has limitation about two-step during climb the stairs with the diameter of wheel 16 cm.
iv. The stair climbing robot is not designed for carrying the load during climb the stairs.
v. Developing an obstacle detection and obstacle avoidance system occur at the frontal part of the mechanism to assist a user to control the mechanism.

1.6 Thesis Outline

This project is focusing on the design of motion control for stair climbing robot for portable mechanism. This research is to help disabled people and elderly people when traveling. Chapter 1 is an introduction of the stair climbing robot and in these chapters also included why the stairs are very important especially for disabled people. It is also discussed how the wheels will react when traversing obstacles like stairs.

Chapter 2 described the literature review based on past research related to stair climbing robot. The most important thing in this chapter is the type of mechanism, type of sensors and type of controller. In order to design the motion control for a stair climbing robot, the right controller need to consider

Chapter 3 described in details the method used in this project. In this chapter, all of the method for every experiment and process to design Fuzzy Logic Controller was described. In addition, the list of component used in this project also stated in this chapter.

Chapter 4 described the result and discussion for the overall project. All the data from three open-loop experiment were discussed in details in this chapter. The result of the designing Fuzzy Logic Controller also described in this section

Chapter 5 described the conclusion and recommendation for the whole project. The overall conclusion of the whole project was discussed either it achieved objective or not. For the recommendation part, the improvement for this project was suggested based on the previous result in order to get a better result

CHAPTER 2

LITERATURE REVIEW

2.1 Overview of Stair Climbing

The stair represents the most dangerous building structure due to its challenging structure for a various vehicle and mobile robot, especially for wheelchair users. A few decades ago, there are many scientists and researcher are presented stair climbing mechanisms for disabled people. This idea is come out when a number of wheelchairs user become increase every year.

The main goal of developing this stair climbing robot is to provide disabled people with a smooth and safe motion during stair climbing and to be independent. There is various type of stair climbing which had been done by scientist and researcher such as a wheelchair. q, Zero Carrier, Shrimp, iBOT, Rocky, and TopChair [2].

2.2 Type of Mechanism

Generally, the design and functionality of the stair climbing robot are depending on their locomotion. There are three main type of locomotive mechanism for stair climbing which are wheeled, legged, tracked and hybrid locomotion [2][3].

2.2.1 Wheeled Mechanism

Wheeled mechanisms are the most popular locomotive mechanism used for mobility because with a simple design it easy to control and provide fast motion with better energy efficiency [4]. Generally, wheeled mechanisms can be classify based on the number of wheels used from two wheels up to five wheels or more depending on the requirement and situation. However, to make the robots stable three wheels is enough for static and moving condition. There are various type of mobile robot with a wheeled mechanism in a real application such as Spirit [4], ORF-L [5], and SHRIMP [6].

Spirit, as shown in Figure 2.1, is one example of the mobile robot based on a rocker-bogie mechanism with six-wheeled and all of the wheels able to adapt on the ground with several environments. A robot with these type of mechanism able to maintain their stability in order to reduce the tip-over. However, the wheel-type mobile robot is difficult to adapt to climbing operation likes steps and stairs. Some researcher in [7] is modified rocker-bogie mechanisms to increase the capability of a mobile robot during climbing.

Figure 2.1: Spirit Robot with Rocker Bogie Mechanism [4]

The iBOT as shown in Figure 2.2 is another type stair climbing based on wheeled locomotive mechanism and it has good performance in term of moving on flat surfaces. The iBOT is an automatic wheelchair consist of four wheels contact to ground and capable to move on flat terrain and climb the stairs without an assistant.

On the other hand, iBOT is controlled by using a computer and several actuators such as sensors and motor so that is able to move on stairs and on uneven terrain [8].

Figure 2.2: iBOT [8]

2.2.2 Legged Mechanism

The stair climbing robot with legged locomotive mechanism concept is suitable for step climbing. The design of the robot with legged mechanism is inspired to make it function more likely as human legs during go up and down stairs. The biped robot is one example of a stair climbing robot with legged locomotion. It was developed by Sugahara et al. by using Stewart Platforms is called WL-16RII (Waseda Leg – No. 16 Refine II) as shown in Figure 2.3 [9].

The robot is designed to fetch disabled people for stair climbing process. Then, the stair climbing robot with a legged locomotive mechanism capable to climb the stairs and was higher adaptability on unstructured environment or slope. The legged mechanism has the advantages of highly to climb the stairs with smooth motion. Quadruped and hexapod are two another example of legged robots which able to climb the stairs and traverse obstacle [10]. However, a legged robot has a complex structure and not provide smooth motion when moving on flat surfaces.

Figure 2.3: WL-16RII (Waseda Leg – No. 16 Refine II) [9]

2.2.3 Tracked Mechanism

Stair climbing robot with track-based which shows a good performance on stair climbing operation and high adaptability on uneven terrain. Commonly, the wheel of a tracked robot consists of two or more belt or track so that it can adapt with stairs structure. Lawn et al [11] are the first researcher introduced stair-climbing wheelchair called as with single track based mechanism as shown in Figure 2.4.

The design makes the stair climbing robot capable to adapt in changed environments such as stairs, curbs, and slope in order to assist elderly and disabled people to be independent. However, these robots will gives high pressure on the edges of stairs due to heavyweight and have low efficiency during stair climbing. Therefore, an efficiency of the robots will increase by an attached motorized wheel on the robot but at the same time, it will increase the complexity structure of the robot.

Figure 2.4: Wheelchair with Single Based Track [11].

Electric Power Wheelchair Stair Climbing (EPW-SC) as shown in Figure 2.5 is one example of stair climbing wheelchair with tracked mechanisms who developed by Tao et al [12]. This tracked robot capable to move on flat terrain and stair climbing. Therefore, tracked based stair-climbing wheelchair is required control system in order to control motion and receive a command to ensure it can perform during ascending and descending stairs with properly.

Other than that, these robots provide better performance in terms of comfortability for wheelchair users during stair climbing. However, track-based stair climbing is difficult to maintain stability at the beginning and end of the stairs. Then, a slip will happen in between track and edge of the stairs. These robots also have a complex design and the price is quite expensive [12].

Figure 2.5: Tracked-based EPW-SC [12].

2.2.4 Hybrid Mechanism

The development of stair climbing robot has become popular now. So the stair climbing robot can be presented with hybrid locomotion. Hybrid locomotion is combining advantages between two type of locomotive mechanism either leg-wheel or track-wheel and is able to adapt in any kind of unstructured environment as well as uneven terrain. The Yuan and S. Hirose have presented “Zero Carrier” stair climbing robot by using leg-wheel hybrid locomotion as shown in Figure 2.6 Zero Carrier is consisting of eight legs and the end of each leg are attached by wheels.

Since the stair climbing robot is designed by using the leg-wheel mechanism, the stair climbing robot is lightweight, compact, powerful and with special hybrid locomotion, it has two type of movement [13]. Firstly, a robot able to walking using legs and each of leg is attached by a wheel on the robot. The second type is the robot able to move on a flat surface in wheel mode and climb the stairs in leg mode depend on the requirement but generally, it will increase the complexity of actuation and becoming difficult to control. However, all of eight legs are control by the motor so that the legs become independently. A robot with wheel-leg locomotives has high efficiency on flat surfaces due to the wheel based and capable to climb the stair with legged based [13][14][15] . Besides that, there is various type of stair climbing robot using leg-wheel locomotion such, Roller Walker, Epi.q-1 and RT-Mover [16].

Figure 2.6: “Zero Carrier” [13].

Other than that, wheelchair. q is another type of stair climbing robot with hybrid locomotion developed by G. Quaglia W. Franco and M. Nisi [17]. As shown in Figure 2.7 below, it used the concept track-wheel locomotion that allowed the robot to move on flat terrain with higher effectiveness of wheeled locomotion and climb the stairs with the performance of a track. The track provided good enough stability during stair climbing with less efficiency. However, in order to rise the efficiency on the flat surface, the wheel of the robot can be controlled by using the motor but on the other hand, the complexity of the robot will increase. However, during stair climbing operation the robot facing difficulty to be stable at the beginning and the end of the stairs. Another researcher in [18][19] proposed the same concept of stair climbing wheelchair for their robots.

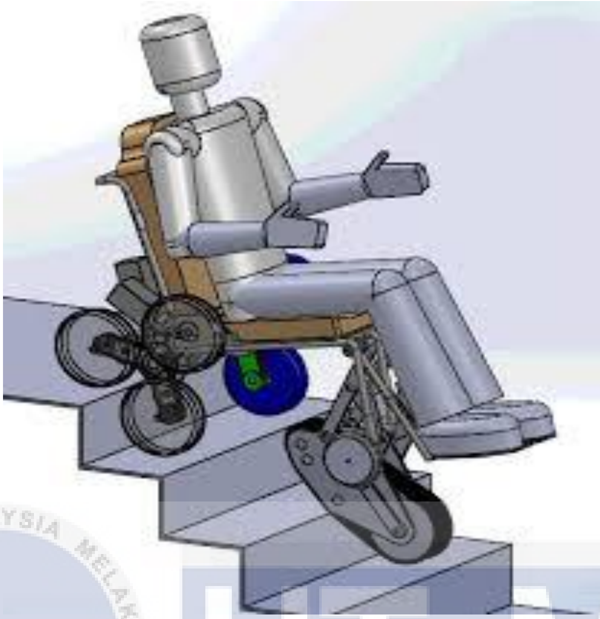

Figure 2.7: Wheelchair. q with track-wheel mechanism [17].

The comparison between wheeled mechanism, legged mechanism, tracked mechanism, and a hybrid mechanism was compared as in Table 2.1. The comparison of the different type of mechanism was compared based on energy efficiency, adaptability on flat surfaces and climb the stairs, control complexity, speed and capability to adapt to the unstructured environment. On flat surfaces, the wheeled mechanism has better performance, however, on the unstructured environment or uneven terrain, legged mechanism provide better performance compared to the wheeled mechanism as well as to climb the stairs.

Table 2.1: The Comparizson Between Wheeled Mechanism, Legged Mechanism, Tracked Mechanism and Hybrid Mechanism

Type of Mechanism	Advantages	Disadvantages
Wheeled	✓ Able to move on flat surfaces with fast motion [2][4][7] ✓ Good energy efficiency [4] ✓ Low control complexity [3]	× Quite difficult to passing stairs or high curbs [14][19] × Not suitable to move on soft terrain and uneven terrain [6]
Legged	✓ Capable of moving on flat surface and stair climbing [3][4][9] ✓ Capable to adapt in any kinds of the unstructured environment [2][4]	× High control complexity [3] [9] [18] × Have slow motion on flat surfaces [10] × High power consumption [11] × Low energy efficiency [12]
Tracked	✓ Capable to climb the stairs [18] ✓ Capable to move on uneven terrain [10] ✓ Low control complexity [18]	× Have slow motion on flat surfaces [12] × Has stability problem at the initial and final of the stairs [11] [12] [18] × Low energy efficiency [4] [18] [19]
Hybrid	✓ Very effective to move on flat surfaces and climb the stair [14][15] ✓ Capable to adapt on uneven terrain [3][14]	× High control complexity [13][15] × High energy efficiency [13]

2.3 Sensor of Stair Climbing Robot

The sensor is one of the devices required when design stair climbing robot. Since stairs are considering as an obstacle for a mobile robot, it becomes important to make sure mobile robots can perform their tasks. There are various type of sensor commonly used in stair climbing robots such as Infrared Sensor, Ultrasonic Sensor and Laser Range Finder (LRF) with different characteristic and functionality [20][21]. A few decades ago, infrared sensor and ultrasonic sensor are widely used in a mobile robot for obstacle detection and obstacle avoiding. For this reason, the requirement of choosing sensor is based on the capability to detect and avoid obstacles. Therefore, the accuracy of the sensor is most important when choosing the sensor.

Infrared Sensor is the most popular sensor used for mobile robots for obstacle detection and obstacle avoidance since it can detect obstacles with various distances depending on requirement. Infrared sensors are widely used in detecting motion due to cheap and high sensitivity to the obstacle. Wheeled mobile robot in [20] is using a GP2D12 distance measuring sensor of the infrared sensor to detect the obstacle. GP2D12 distance measuring sensor is lightweight, compact and high ability to receive the signal to detect the object. In this robot, the infrared sensor is implemented in order to avoid the obstacle when moving.

The ultrasonic sensor also one of the example sensor which is suitable for obstacle avoidance and obstacles detection. These type of sensor only can give the signal to the robot if any obstacle around the robot. However, the ultrasonic sensor cannot give the accurate position of the obstacles. Pham et al. [22] are used ultrasonic sensor in order to control the movement of elderly and disabled people when moving on the different structure of the environment

In this research, a stair climbing robot with the infrared sensor will introduce in order to detect and avoid obstacles. Since the infrared sensor can detect obstacles at the various distance, hence it is the best solution for stair climbing robot when moving.

2.4 Stair Climbing Control Technique

Research on controlling the stair climbing has gained a big attraction around the world because stair climbing robot can replace the manual wheelchair and user can be dependent without the human assistant. Due to nonlinearity characteristic, it makes the equation of motion control as a difficult task to derive, several types of control techniques applied a few years ago from researchers to control the stair climbing robot. The motion control system of the stair climbing robots is required the suitable controller to maintain stability and smoothness of the robot during stair climbing [23].

There are various type of controller can be used in stair climbing to control the motion such as Fuzzy Logic Controller, Proportional-Integral-Derivative (PID), and Artificial Neural Networks (ANN). Other than that, Adaptive Neuro-Fuzzy Inference Systems (ANFIS) is a new control technique which is implemented to control the position of the stair climbing robot.

2.4.1 Proportional Integral Derivative (PID) Controller

Proportional Integral Derivative (PID) is a type of conventional controller and widely used in industry. PID give advantages in term of simple to use compared to other controller. However, PID controller is required mathematical model when designing and for a complex system it will increase the number of input and output of the system [24]. In other words, complexity to design a controller for system become increase and not provide accuracy for the system. In stair climbing robot, PID is implemented to control the movement of motor [21]

There are three parameters are used in PID such as proportional term (Kp), an integral term (Ki) and derivative term (Kd). These parameters can be set accurately in order to get the best response performance. In this research, DC motor was used and the output of the controller was depended on the voltage of the DC motor used. However, the design of PID controller will be tuning by using two methods such as Ziegler-Nichols and Trial and Error Method [25].

In order to design a controller based on PID, there are several characteristics should be considered. The advantages of PID controller is the performance can be improving in term of transient response. However, the transient response will affect four characteristics likes rise time, overshoot, settling time and steady-state error for parameter Table 2.2 shows the result of improving performance by using a PID controller.

Table 2.2: Performance of PID parameter characteristic on closed loop system.

Closed Loop Response	Rise Time	Overshoot	Settling Time	Steady-State Error
Kp	Decrease	Rise	Remain	Decrease
Ki	Decrease	Rise	Rise	Degrade
Kd	Remain	Decrease	Decrease	Remain

2.4.2 Fuzzy Logic Controller

Recently, Fuzzy Logic Controller, Artificial Neural Networks and Adaptive Neuro-Fuzzy Inference Systems controller have been presented to solve the problems of stair climbing robot when using other controllers such as PID controller. Fuzzy Logic Controller (FLC) is one example of the unconventional controller and in order to design a controller mathematical model is not required [26].

Fuzzy set theory was introduced by Zadeh [27] and it act as a controller at the first time about a few decades ago. Then, fuzzy set theory was used in several field especially for control system. Until todays, there are many researchers used fuzzy logic controller in mobile robot as a controller.

Ahn et al. [28]are first person whose introduced fuzzy logic controller and open loop module was design in first time. In this research, fuzzy logic controller is quite difficult to design because it depends on several factor such as speed and parameter used. However, in order to increase the speed and decrease the parameter used, fixed control table was used to solved this problem [28].

Then, bhende et al. make the comparison between two type of fuzzy logic controller with one type of convetional controller in term of application [29]. The two type of fuzzy logic controller is Takagi-Sugeno (TS)-type and Mamadani-type and conventional controller is Proportional Integral (PI) controller. These three type of controller above was analyzed when it connected to a power filter to improve the performance of control system in term of power quality and reactive power compensation. Fuzzy logic controller is widely used because in order to design a controller, a mathematical modelling is not required.

Mamdani-type was used in stair climbing wheelchair by using multiple input single output (MISO) [29]. In order to design MISO in the system, the inaccuracy and change of inaccuracy need to be measured. In this research, the PD-fuzzy logic was applied to control the front wheel and rear wheel by design it using MATLAB software. Controller in this system is used to provide enough torque by maintaining the constancy and smooth motion of the robot. The smoothness of the robot can be control by using five rule of the Gaussian membership function. This five rule membership are Positive Big (PB), Positive Small (PS), Zero (Z), Negative Small (NS) And Negative Big (NB) [30] [23].

In [31] Fuzzy Logic Controller is used to controlling navigation of mobile robot for obstacle avoidance. This is because obstacle avoidance is commonly become a problem for the mobile robot to moving and it is difficult to make the decision when on the navigation path. Based on previous research, in order to design an unconventional controller, the membership function is required. Based on membership function, the mobile robot will able to make the decision for obstacle avoidance. Then, the performance was tested by using multiple mobile robots for effectiveness.

The unconventional controller such as Fuzzy Logic Controller has better performance in stair climbing robot compared to the conventional controller. These type of controller able to give smooth motion and stability for the robot during moving in many ways.

2.5 Summary

Based on past research, there are many types of mechanism and type of controller used for stair climbing robot has been reviewed. However, stair climbing robot with wheel-legged mechanism are widely used and this mechanism looks more interesting because it acts as a human leg. Furthermore, Fuzzy Logic Controller is chosen for this research because it offers many benefits for the robots especially for non-linearity and also has a simple structure. The proposed solution for stair climbing robot in order to design motion control were described in chapter 3.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the designing Fuzzy Logic Controller by using Multiple Input Multiple Output (MIMO) of the stair climbing robot will be discussed in details. The designing of stair climbing motion control by suing Fuzzy Logic Controller should consider several factors in order to achieve the objective. From the research, the robot with leg-wheel mechanism is considered the most suitable mechanism for moving on the flat surface and rough terrain. As mentioned previously in Chapter 1, the robot with legged-wheeled mechanism is combining the advantages of legged mechanism and wheeled mechanism.

Hence, the proposed of the stair climbing robot for this project is based on legged-wheeled mechanism. The wheeled mechanism has a smooth motion on the flat surface and legged mechanism able to adapt in any kinds of the environment such as stair climbing. The design for this project is based on Delta-Wheel robot. The Delta-Wheel with three wheels are arranged at each corner of triangular shape and the robot consists of twelve wheels. When moving on flat surfaces or slopes, all of the wheels contacts on the ground will rotate while when stair climbing four Delta Wheel will rotate.

Besides that, the methodology for obstacle detection experiment, obstacle avoidance experiment, and stair climbing experiment will be discussed in detail in this chapter. The obstacle detection experiment will be two test which is distance test and angle test. However, for obstacles avoidance experiment will have three different situations which are an obstacle at sensor 1, the obstacle at sensor 2 and obstacles at sensor 1 and sensor 2. The stair climbing experiment will be tested without a load.

3.2 Project Flowchart

The flowchart in Figure 3.1 shows the process for stair climbing robot for this research.

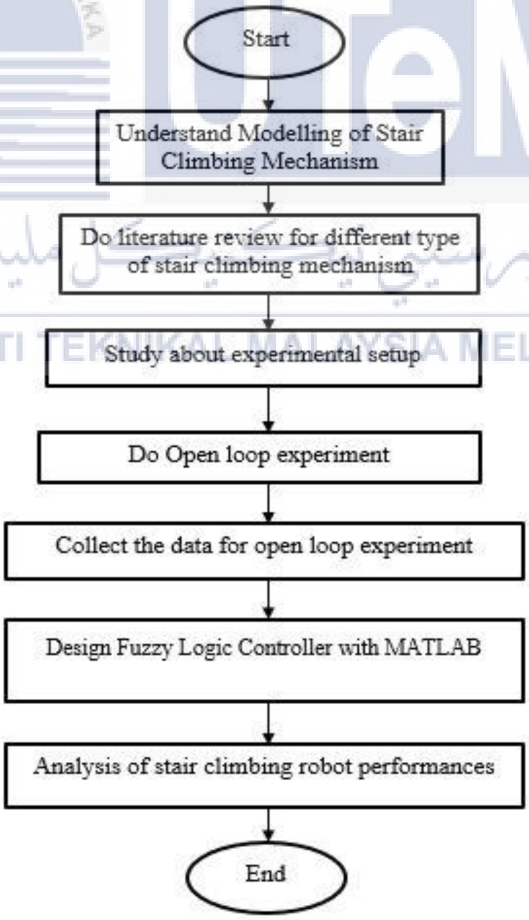

Figure 3.1: Flowchart of Project Development

3.3 System Overview of Stair Climbing System

The controller was used to control the movement of the stair climbing robot when moving on the flat surface or on different terrain. The stair climbing robot also has obstacle detection and obstacle avoidance system. The block diagram in Figure 3.2 shows a system overview of a stair climbing robot by using the Arduino. Then, the power supply 5V supplied to the Arduino. The Joystick is connected to the Arduino to give the signals to move on flat surfaces or stair climbing. Besides that, the Arduino also receives a signal from the infrared sensor which placed at the two front corners of the robot to sense the obstacles. The Arduino programming from the computer will give the command to the motor driver in the Pulse Width Modulation (PWM) form and then the motor driver is connected directly to the DC motor to control the direction and speed.

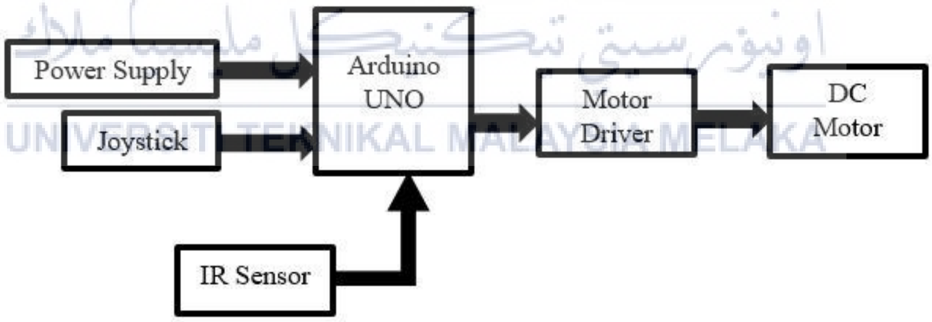

Figure 3.2: Block Diagram of Stair Climbing System

3.4 Stair Climbing Experimental Setup

The experiment is conducted by connecting stair climbing robot with Arduino. The stair climbing robot has two DC geared motor (XYD-16) at the back that connected with the rear Delta-Wheel mechanism. The movement of Delta-Wheel will be control by DC geared motor. Figure 3.3 shows the side view of the stair climbing robot and Figure 3.4 shows the top view of stair climbing robot.

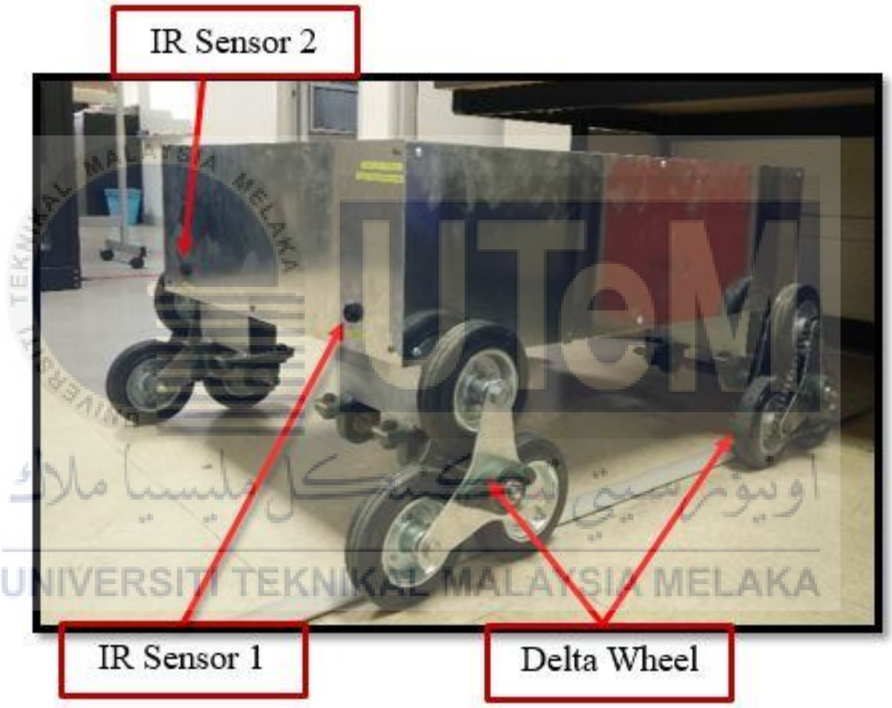

Figure 3.3: Side View of Stair Climbing Robot

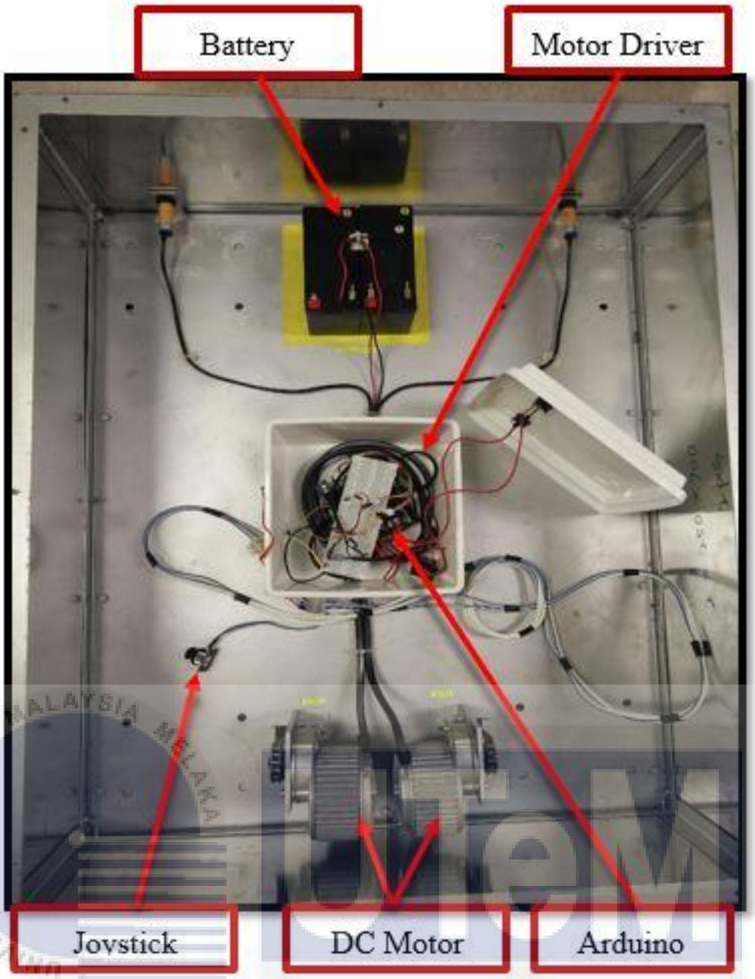

Figure 3.4: Top View of Stair Climbing Robot

3.4.1 Delta-Wheel

Delta Wheel as shown in Figure 3.5 is combining between the wheel and leg mechanism it called as a hybrid mechanism. In this project, four delta wheel will be implemented into a stair climbing robot. The basic operation of stair climbing robot with delta wheel is when moving on flat surfaces all of the wheels which contact on the ground will rotate otherwise when stair climbing or traverse obstacle four delta wheel will be rotated.

Figure 3.5: Delta-Wheel Mechanism

3.4.2 Arduino UNO

This stair climbing robot is used Arduino UNO as a controller due to its cheap and user-friendly. Arduino UNO in Figure 3.6 will be received a signal from joystick and IR sensor in order to control the robot.

Figure 3.6: Arduino UNO

3.4.3 DC Geared Motor

An actuator is one of the most important devices should be considered for stair climbing robot. DC motor is the most popular actuator used in designing a stair climbing robot. The DC motor usually provides high torque and low speed where it is suitable to move and support stair climbing robot during climb the stairs. The speed of the DC motor is controlled by using Pulse Width Modulation (PWM) method. However, to provide high torque and lower speed of the motor, DC geared motor is the most suitable actuator to overcome this problem. DC Geared motor (XYD-16) as shown in Figure 3.7 will be used in this project.

Figure 3.7: DC Geared Motor

3.4.4 Joystick

A joystick is an input device that use to control stair climbing robot by manually. An analog joystick as shown in Figure 3.8 will be employ to control the motion of stair climbing robot. This type of joystick is widely used to control the motion of mobile robots in any direction either left or right or vice versa. Design of joystick is flexible and it is easy to control compared to the keyboard. The output pin is located at the center of the joystick and the output voltage is based on stick position. This joystick will be used in stair climbing robot to control the motion of the robot in different direction. However, it only can function if receive the command from the microcontroller.

Figure 3.8: Joystick

3.4.5 Infrared Sensor

The sensor is one of the important equipment for stair climbing robot in order to detect an obstacle and avoid obstacles and provide safety for users. The accuracy of the sensor should be considered when choosing a sensor since there are many Infrared Sensor offers difference characteristic and functionality. In this project, two Digital Infrared Sensor with the number of model SN-E18-B03N1 as shown in Figure 3.9 will be used since it providing simplest structure, lightweight and consume less time to detect an object or obstacles.

The basic operation of Infrared Sensor (IR) is it contains LED and works by reflecting IR signal from an obstacle. These type of IR sensor can detect an obstacle from 0cm to 30cm with the 5V voltage supply. However, the distance of the obstacles can be detecting is fully dependent on the voltage supply. The length of distance can detect obstacle will increase as well as voltage supply increase. The sensitivity of SN-E18-B03N1 can be adjusted manually depend on the requirement of mobile robots. This IR sensor will place at frontal part of the robots and the signal when detecting an obstacle is sent to a controller.

Figure 3.9: Infrared Sensor

3.4.6 Motor Driver

The MDDS30 motor driver in Figure 3.10 was employed in this stair climbing robot to control 24V DC Geared Motor. In this project, Arduino UNO was used as a controller and it required an additional part to control the movement of the motor. Arduino Uno cannot be used to control DC motor due to not provide enough power supply. Arduino UNO only provided voltage supply from 5V up to 12V and the current from Arduino is insufficient to drive the DC motor.

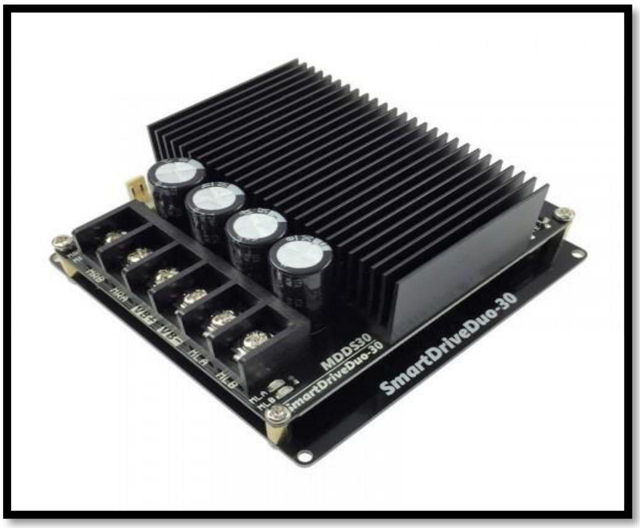

Figure 3.10: MDDS30 Motor Driver

3.5 Experiment of Stair Climbing

In order to design a controller for stair climbing robot, there are several experiments will be run for this project. This experiment is important to identify the functionality of every component used in this robot and the capability of the robot to adapt in different ways. Table 3.1 shows the list of experiment was run in this project.

Table 3.1: List of Experiments

No	List of Experiments
1	**Obstacle Detection Experiment** • Distance test 0-50 cm • Angle test (0-180° with interval 10°) at distance 10 cm, 20 cm and 30 cm.
2	**Obstacle Avoidance Experiment** • Obstacle at left side • Obstacle at right side • Obstacle at both side
3	**Stair Climbing Experiment Without Load**

3.5.1 Experiment 1: Obstacle Detection

Obstacle detection is important for the mobile robot when moving especially on the unstructured environment. Hence, obstacle detection experiment was conducted for stair climbing robot to analyze the effect of infrared sensor towards the obstacles. For this experiment, there is two test which is distance test at distance 0 cm to 50 cm and angle test from 0° to 180° at a different distance.

Obstacles detection distance test experiment is shown in Figure 3.11 for the top view and Figure 3.12 for the side view. Firstly, for distance test, IR sensor was placed at the height of 3 cm from the table. Then, 5V DC power supply will be supply to the sensor. The voltage at the signal wire will be measured when the obstacles at 90° from the distance 0 cm to 50 cm. The value of output voltage will record and repeat three times to get the average value.

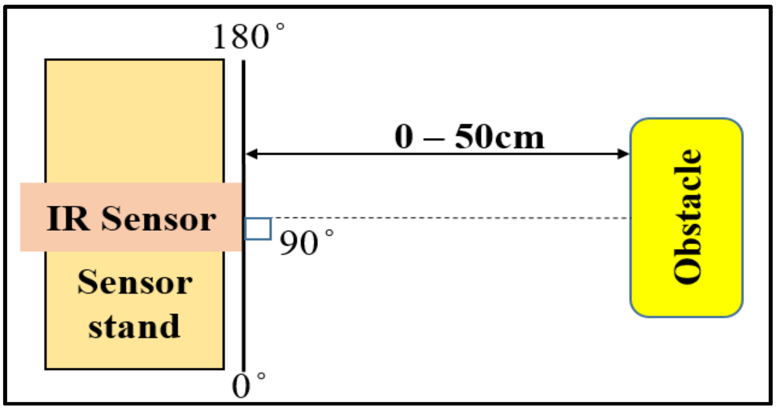

Figure 3.11: Top View of Obstacle Detection for Distance Test

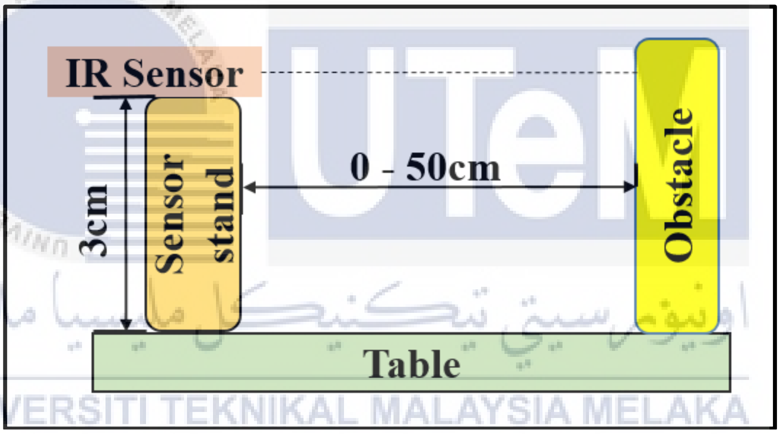

Figure 3.12: Side View of Obstacle Detection for Distance Test

Figure 3.13 and Figure 3.14 show the diagram setup from the top view and side view for angle test at distance 10 cm, 20 cm and 30 cm with the various angle. The IR sensor was placed at height 3cm from the table and the 5V DC power supply was supply to the sensor. The obstacles are moved from 0° to 180° at the fixed distance 10 cm and the output voltage was recorded. The experiment is repeated for distance 20 cm and 30 cm and the output voltage was recorded. The IR sensor give signal once it detects the obstacles.

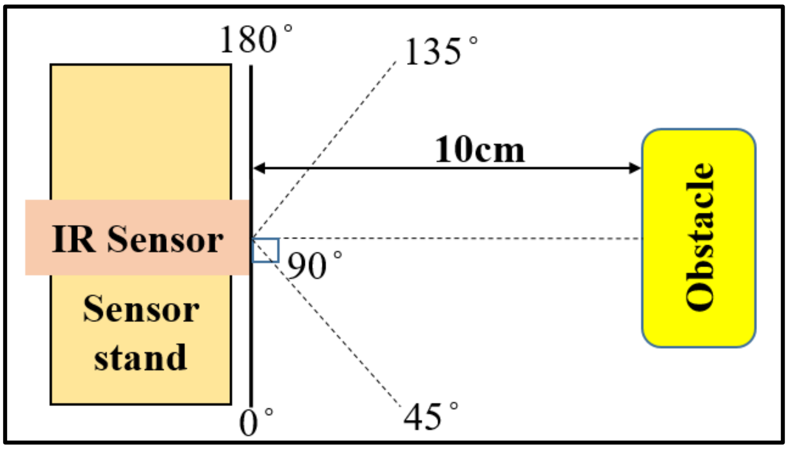

Figure 3.13: Top View of Obstacle Detection for Angle Test

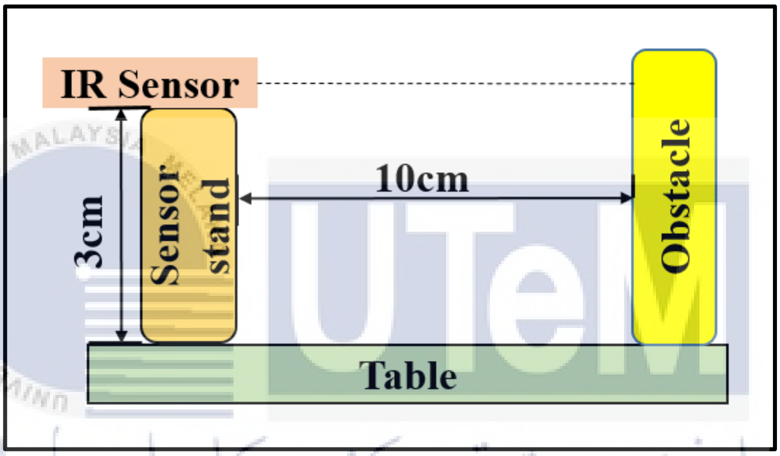

Figure 3.14: Side View of Obstacle Detection for Angle Test

The output voltage is measured in order to prove the working principle for the IR sensor in term of obstacle detection. The output voltage of IR sensors is directly proportional to the distance of obstacles detected. Hence, the result of the experiment can be analyzed.

3.5.2 Experiment 2: Obstacle Avoidance

Obstacle avoiding is the second experiment was conducted for stair climbing robot to analyze the effect of IR sensor towards the obstacles in term of avoiding. In this stair climbing robot, two IR sensor will be placed at left corner (S1) and right corner (S2) on the frontal part of the robot. Both of IR sensors are supplied by the 5V power supply and the signal from IR sensors is sent to the Arduino. Then, Arduino received the command in order to control the movement of the robot. The stair climbing robot able to avoid the obstacle when moving with three different situations which are obstacles at right, obstacles at left and obstacles at both sides.

Figure 3.15 shows the illustration for obstacles at left placed. In this experiment, there is two motor which is Motor 1 on the left side (M1) and Motor 2 on the right side (M2) was work accordingly based on IR sensors. The joystick be used to control the motion of the motor in four directions such as forward, reverse, left and right. When the stair climbing robot moves forward and left sensor (S1) sense an obstacle, the robot should be able to make a decision either to stop or turn right. The robot was set to avoid obstacles when it detected an obstacle at distance 20 cm at all different situation.

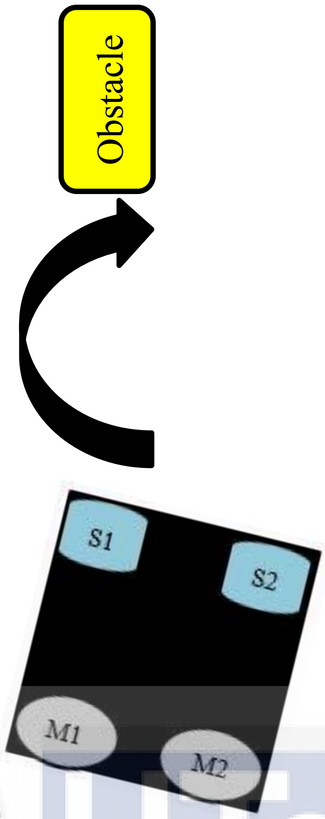

Figure 3.15: Obstacle Avoidance at Sensor 1

Next, Figure 3.16 shows the illustration when the obstacle at the right side which is at S2. When the stair climbing robot moved forward or in the other word joystick was in the forward direction and suddenly detected an obstacle, both motors were stopped. Then, when joystick was in reverse direction both motor will reverse to avoid the obstacles. However, if the joystick in right direction, the both will stop since the obstacle was detected by S2 and at this position, the stair climbing robot will be able to make a decision either to stop or turn left. Therefore, in another direction robot can move freely.

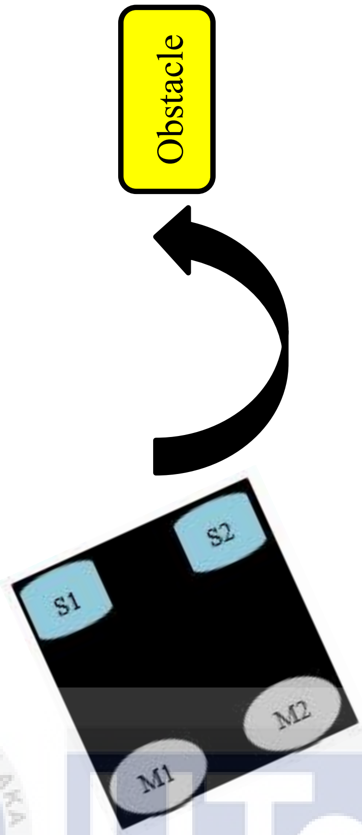

Figure 3.16: Obstacle Avoidance at Sensor 2

Then, the last situation is when obstacles were detected at both sides as shown in Figure 3.17. The joystick controls the robot to move forward and the both IR sensors were detected obstacles at S1 and S2. Then, the M1 and M2 will be stopped in any direction except in reverse condition.

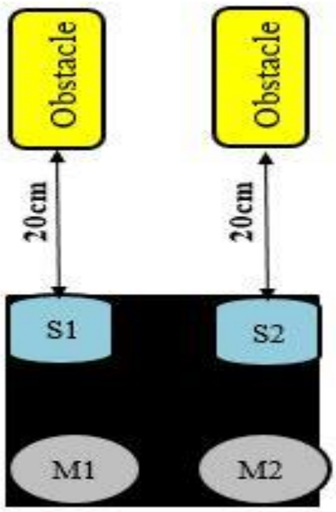

Figure 3.17: Obstacle Avoidance at Sensor 1 And Sensor 2

In this experiment, the condition for both motors was measured when the IR sensors detected an obstacle at any side. Then, when the IR sensors detected an obstacle at three different situations, the stair climbing robot cannot move forward and it should be not hitting obstacle. This experiment was being tested in order to achieve objective two based on scope number two.

3.5.3 Experiment 3: Stair Climbing

Stair climbing experiment was being the most challenging experiment for this project since it needs considering many parameters such as the height of the stair, width of the stairs, radius of wheel and radius of Delta-Wheel. This experiment is to tested and analyze the performance of Delta-Wheel mechanism in order to adapt with stairs structure. The delta-wheel has three wheels attached together with triangular shape and it looks like human legs when climbing the stair.

First thing first, the height and gradient of the stairs was measured and the stair climbing robot only limited to climb the stairs with the two-step. The experiment was test when gradient of the stairs is 10°. The joystick was pushed in forward to move the robot in the forward direction and will stop at front of the stairs. Figure 3.18 shows the illustration of the robot in front of the stairs. On flat surfaces, the wheel which contact on the ground will rotate, however, when climbing the stair, the wheel was changed to Delta-Wheel.

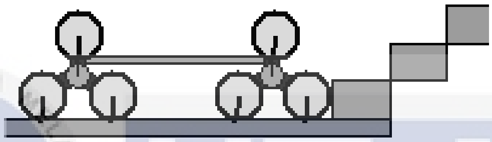

Figure 3.18: Stair Climbing at Stage 1

Noted that, when the robot start hit the stair surfaces, the revolution of the delta-wheel will start as shown in Figure 3.19 and Figure 3.20 and the timer will start to record the time taken for the robot to finish stair climbing task. In this stage, the DC geared motor provided high torque so that the robot can push itself to climb the stairs. The DC motor with high torques gives friction to the wheel to grip the surfaces of the stairs.

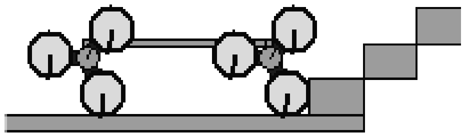

Figure 3.19: Stair Climbing at Stage 2

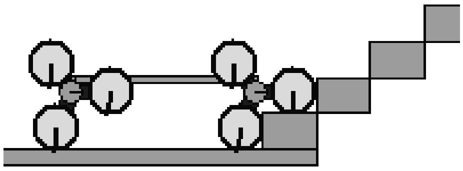

Figure 3.20: Stair Climbing at Stage 3

When the robot reached the first stair, the delta-wheel will rotate again to climb another stair as shown in Figure 3.21 and Figure 3.22. The delta-wheel robot will keep following the same step as previously until it reached second stairs.

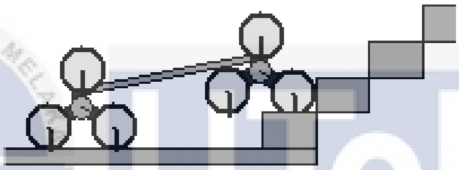

Figure 3.21: Stair Climbing at Stage 4

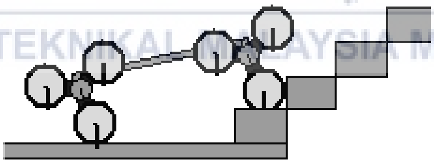

Figure 3.22: Stair Climbing at Stage 5

Once it reached the second stairs, the timer will be stop and time taken is recorded. The experiment will be repeated three times for repeatability. Figure 3.23 shows the position of the robot when it reached the second stairs.

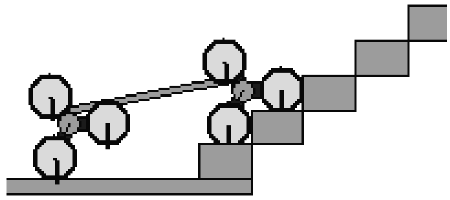

Figure 3.23: Stair Climbing at Stage 6

The time taken for the stair climbing robot to finished second stairs was recorded and the experiment was repeated when gradient of stairs is 20°. This is to identify how fast the robot can climb the stairs at different of gradient.

3.6 Fuzzy Logic Controller Design

A Fuzzy Logic Controller (FLC) is a controller based on fuzzy set theory and fuzzy logic and widely used in mobile robot because it can handle a certain level of imprecision and uncertainty. In this research, the FLC was chosen as a controller to control the motion of the stair climbing robot in term of obstacle avoidance. For this controller, there are two input were used which are left sensor(S1) and right sensor(S2) for detection of obstacles and then the output of the controller was left motor speed movement and right motor speed movement that measured in Pulse Width Modulation (PWM) for motion of the robot. Therefore, this system has two inputs and two outputs.

MATLAB Toolbox was used to design FLC that consist of several libraries can be used in order to easily design, testing and simulate the Fuzzy Logic Controller. In addition, there are membership functions, graphical user interfaces (GUI) in this MATLAB Toolbox, so that fuzzy inference system can be easily modified with that structure. Based on Figure 3.24 below, the figure has shown the block diagram of

Fuzzy Inference System. All of the procedure to design a FLC is discussed in this segment.

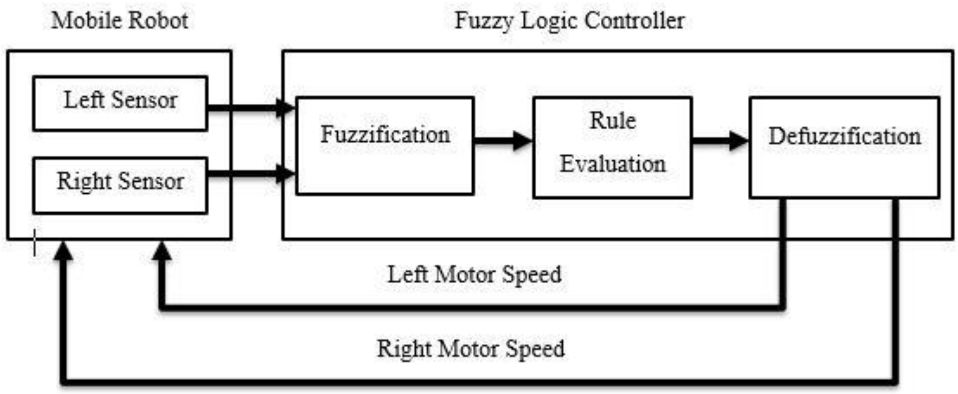

Figure 3.24: Block Diagram of Fuzzy Controller

3.6.1 Fuzzification

Fuzzification is a method used to generate the output to input into fuzzy sets theory. In fuzzification, there is the various classification of the graph can be applied such as trapezoidal, Gaussian and triangular membership function. However, trapezoidal and triangular, and bell-shaped is widely used because it is simple compared to others. In this research, the input of FLC is by getting data of distance of obstacles between the robot from the left sensor and the right sensor located at the front corner of the robot. This input is defined by using two different variables which are Near and Far and the range is set from 0 (minimum) to 50 (maximum) for both sensors and it design by using Z-Shaped and Sigmoidal membership functions. This is because both sensors can have covered the distance up until 27 cm and more than 27 cm is considered as far. Then, the output of FLC also defined as the same with the input which uses two different variables for left motor speed movement and right motor speed movement. The two variables are Reverse and Forward and the range is set from

0 (minimum) and 100 (maximum) that measured in PWM and this variable are designed by using Z-Shaped and S-Shaped membership functions.

3.6.2 Fuzzy Rule Evaluation

Fuzzy Rule is design to control the movement of the robot and the system consists of two input and two output and each of this has two variable of the membership function. Therefore, four possible rules has been designed for this system to control the movement of the robot. Table 3.2 and Table 3.3 below shows the possible fuzzy rules used for right motor speed and right motor speed. All of these rules are the relative between input and output. Therefore, the overall rule of the system has shown Table 3.4.

Table 3.2: Fuzzy Rule for Left Motor Speed Movement

Left Motor Speed Movement		Right Sensor	
		Near	Far
Left Sensor	Near	Reverse	Forward
	Far	Reverse	Forward

Table 3.3: Fuzzy Rule for Right Motor Speed Movement

Right Motor Speed Movement		Right Sensor	
		Near	Far
Left Sensor	Near	Reverse	Reverse
	Far	Forward	Forward

Table 3.4: Rule for Fuzzy Logic System

No of Rules	Left Sensor (S1)	Right Sensor (S2)	Left Motor Speed	Right Motor Speed
1	Near	Near	Reverse	Reverse
2	Near	Far	Forward	Reverse
3	Far	Near	Reverse	Forward
4	Far	Far	Forward	Forward

Based on the feedback from the sensors, the fuzzy rule of the controller was triggered. The rules above are used to trigger the outputs of the fuzzy system that form from a combination of the speed of the motor. The output is representing the left motor and right motor where speed is measured in PWM.

In this stair climbing robot, the different number of membership function need to consider. There are two membership function for inputs are choosing for both left sensor and the right sensor which are “Near” and “Far”. These membership functions come from Z-Shaped and Sigmoidal types of membership function. The rules of fuzzy controls were works by triggered the robot by using sensors.

The outputs of the fuzzy controls are defines as left motor and right motor respectively. Every output consists of three types of membership function that can be described as “Reverse” and “Forward” for both motor. Therefore, the characteristics of the input and outputs were described in Table 3.5,

Table 3.6, Table 3.7 and Table 3.8 below and the values that appear was represented the x-axis of membership functions.

Table 3.5: Characteristics for Left Sensor

Term	Range of Membership Function (cm)	Type of Membership Function
Near	20: 27	Z-Shaped (zmf)
Far	5 : 27	Sigmoidal (sigmf)

Table 3.6: Characteristics for Right Sensor

Term	Range of Membership Function (cm)	Type of Membership Function
Near	20: 27	Z-Shaped (zmf)
Far	5 : 27	Sigmoidal (sigmf)

Table 3.7: Characteristics for Left Motor Speed Movement

Term	Range of Membership Function (PWM)
Reverse	0 : 40
Forward	50 : 100

Table 3.8: Characteristics for Right Motor Speed Movement

Term	Range of Membership Function (PWM)
Reverse	0 : 40
Forward	50 : 100

Two inputs and two outputs was used in designing Fuzzy Logic Controller. Therefore, four number of rules was created by using FIS as shown in Figure 3.25.

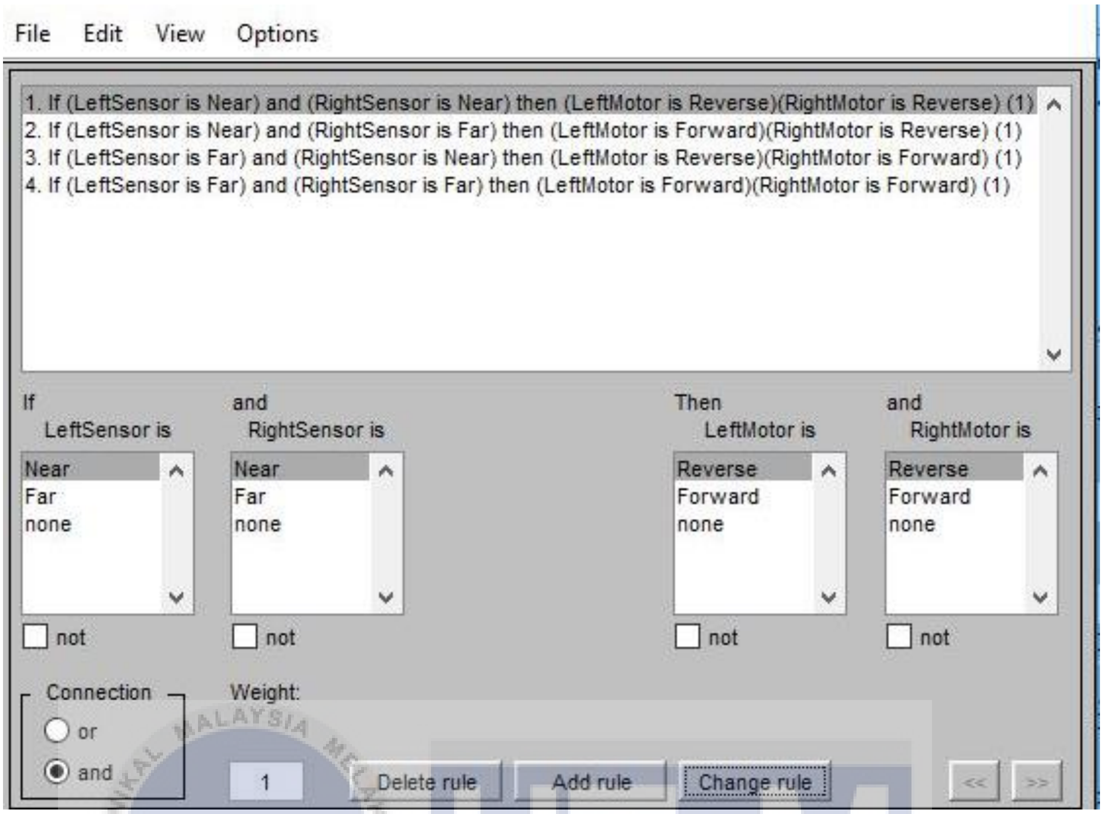

Figure 3.25: Rule Viewer

The rules that created was shown as below:

1. If *(Left Sensor is Near and Right Sensor is Near)* Then (*Left Motor is Reverse and Right Motor is Reverse)*
2. If *(Left Sensor is Near and Right Sensor is Far)* Then *(Left Motor is Forward and Right Motor is Reverse)*
3. If *(Left Sensor is Far and Right Sensor is Near)* Then *(Left Motor is Reverse and Right Motor is Forward)*
4. If *(Left Sensor is Far and Right Sensor is Far)* Then *(Left Motor is Forward and Right Motor is Forward)*

3.6.3 Defuzzification

Defuzzification is used to exchange output of fuzzy with one single crisp by using fuzzy set. In Fuzzy Logic Controller, the defuzzification was applied in a control system. Defuzzification process are widely used in many application as a control action especially to actuate the control plant. There are the various method of defuzzification was introduced such as centroid method, max membership method, weight average method, and mean-max membership method.

However, the centroid method was chosen compared to others because this method is most suitable for the mobile robot by generating the center of the area for the system. In order to evaluate the IR sensor configuration, several condition of the inputs need to considered. For example, when left sensor detect the obstacles at near condition, right sensor automatically not detect the obstacles. Figure 3.26 shows the centroid of method for configuration of the left motor and right motor when receive signal from both IR sensors.

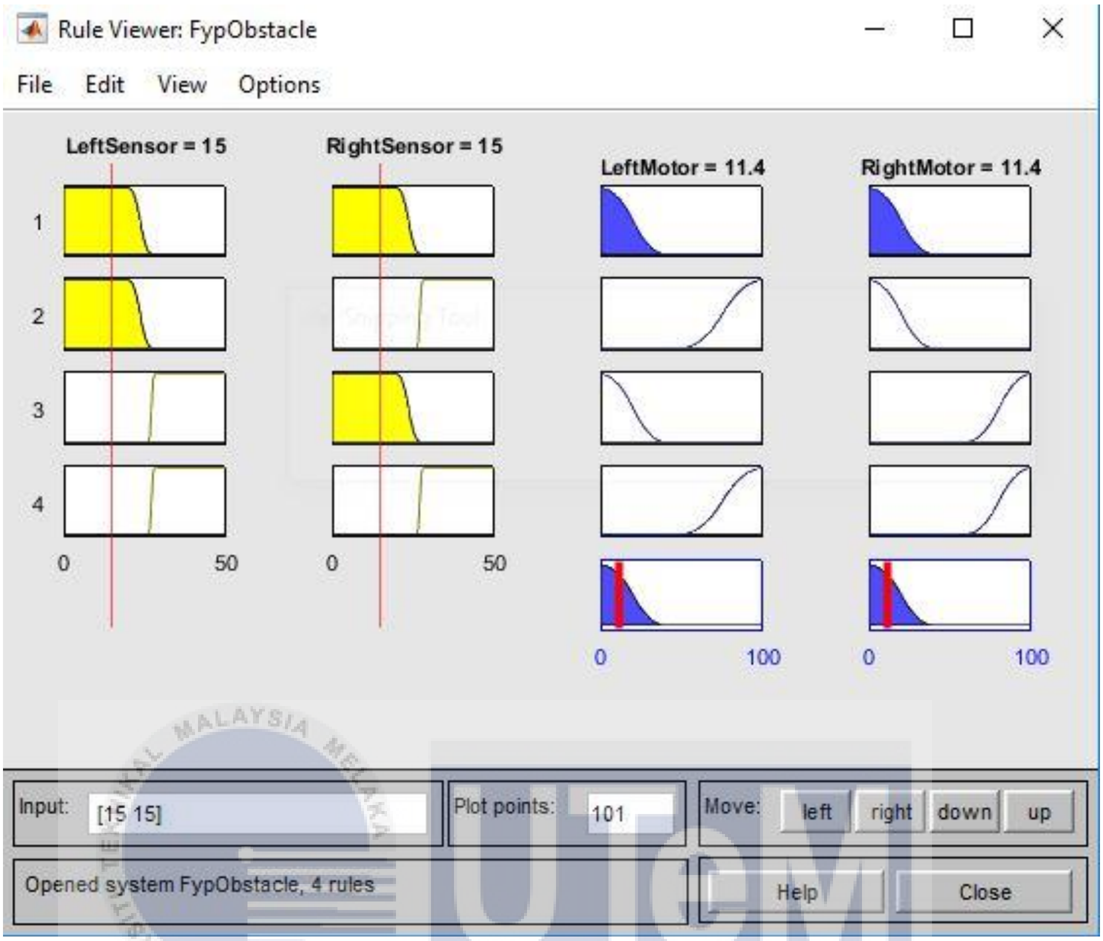

Figure 3.26: Centroid Method

3.7 Summary

At the end of this section, the expriment setup for every experiment has presented. All of the procedure and equipment used for each experiment was stated clearly. The stair climbing robot was design based on legged-wheeled mechanism and the component implemented in this robot are 24V DC Motor, Arduino UNO, IR Sensors, 24V Battery, Joystick and MDDS30 Motor Driver. In this Chapter, the controller used to control the motion was determined. Therefore, Fuzzy Logic Controller was design and implement in this robot. The method to design FLC was explain in details. The results after implemented controller was discussed in next chapter.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

In this chapter, the result for the whole project are discussed for both open loop test and closed-loop test that have been done. Three experiments which are obstacle detection, obstacle avoidance, and stair climbing already done. In addition, this chapter also presents the designing of Fuzzy Logic Controller. All of the results are discussed in this chapter.

4.2 Open Loop Experiment

Open loop experiment is the experiment that has been done by testing the movement of the robot without controller which is no feedback to the system. The result for all expriment was discussed in sub section below.

4.2.1 Obstacle Detection

The reading of output voltage for obstacle detection distance test experiment without a controller at distance from 0 cm to 50 cm with interval 3cm at 90 degrees as shown in Table 4.1. The data are collected from three times repeatability and the average of the output voltage were calculated. Hence, the standard deviation was calculated to obtain the error value. However, the value of the output voltage for three times repeatability was the same value. Then the standard deviation value becomes 0.

Based on Figure 4.1, the graph shows output voltage versus distance of the obstacle from 0 cm to 50 cm at 90 degrees. The IR sensor only detects an obstacle at distance from 0 cm to 27 cm and gives the output voltage 0.62V. At distance 28 cm or more, the IR sensors cannot detect any obstacle with output voltage 4.97V.

Table 4.1: Distance Test from 0 cm to 50 cm with interval 3 cm at 90 degrees.

Output Voltage (V) / Distance of the Obstacles (cm)	**Repeatability**				
	1	**2**	**3**	**Average**	**Standard Deviation**
0	0.62	0.62	0.62	0.62	0.00
3	0.62	0.62	0.62	0.62	0.00
6	0.62	0.62	0.62	0.62	0.00
9	0.62	0.62	0.62	0.62	0.00
12	0.62	0.62	0.62	0.62	0.00
15	0.62	0.62	0.62	0.62	0.00
18	0.62	0.62	0.62	0.62	0.00
24	0.62	0.62	0.62	0.62	0.00
27	0.62	0.62	0.62	0.62	0.00
30	4.87	4.87	4.87	4.87	0.00
33	4.87	4.87	4.87	4.87	0.00
36	4.87	4.87	4.87	4.87	0.00
39	4.87	4.87	4.87	4.87	0.00
42	4.87	4.87	4.87	4.87	0.00
45	4.87	4.87	4.87	4.87	0.00
48	4.87	4.87	4.87	4.87	0.00
51	4.87	4.87	4.87	4.87	0.00

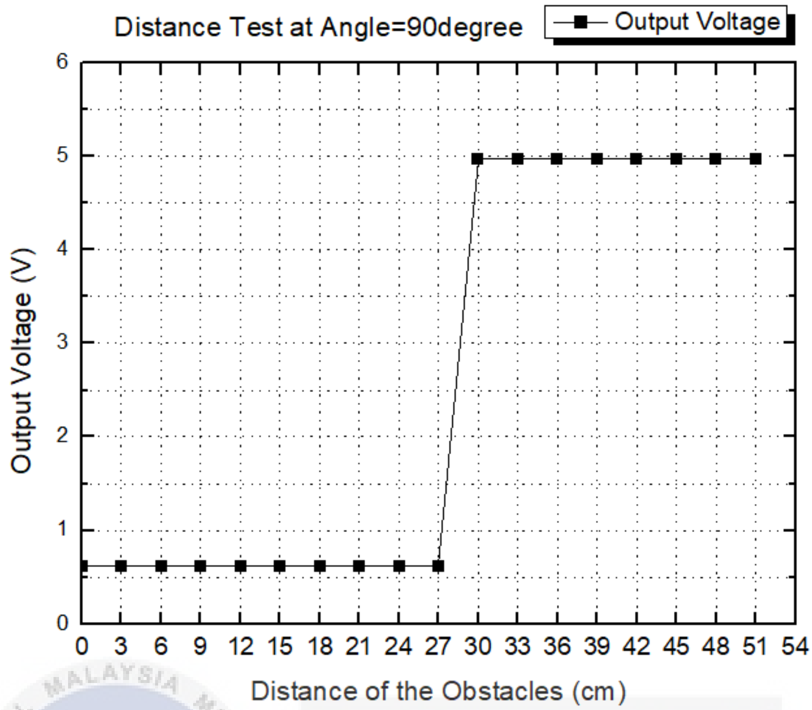

Figure 4.1: The Graph Output Voltage versus Distance of the Obstacle

The output voltage of obstacle detection angle test at distance 10 cm, 20 cm and 30 cm was shown in Table 4.2 with the angle from 0 degrees to 180 degrees. The graph for angle test with the angle from 0 degrees up to 180 degrees at distance 10 cm, 20 cm and 30 cm was shown in Figure 4.2. Based on the graph, there are three different colour which are the black colour for distance 10 cm, red colour for distance 20 cm and green colour for 30 cm. The obstacles at 10 cm only can be detected by IR sensors from angle 80 degrees to 100 degrees. Then, at 20 cm the IR sensors only detect the obstacle at 90-degree angle. However, at 30 cm the IR sensors cannot detect any obstacles at any angles. From the experiment at distance 16 cm or more, IR sensors cannot detect any obstacles except at 90-degree angle.

Table 4.2: The Output Voltage at distance 10 cm, 20 cm and 30 cm with various angle.

Output Voltage (V) / Angle of the obstacles (degrees)	At distance 10cm	At distance 20cm	At distance 30cm
0	4.97	4.97	4.97
10	4.97	4.97	4.97
20	4.97	4.97	4.97
30	4.97	4.97	4.97
40	4.97	4.97	4.97
50	4.97	4.97	4.97
60	4.97	4.97	4.97
70	4.97	4.97	4.97
80	0.62	4.97	4.97
90	0.62	0.62	4.97
100	0.62	4.97	4.97
110	4.97	4.97	4.97
120	4.97	4.97	4.97
130	4.97	4.97	4.97
140	4.97	4.97	4.97
150	4.97	4.97	4.97
160	4.97	4.97	4.97
170	4.97	4.97	4.97
180	4.97	4.97	4.97

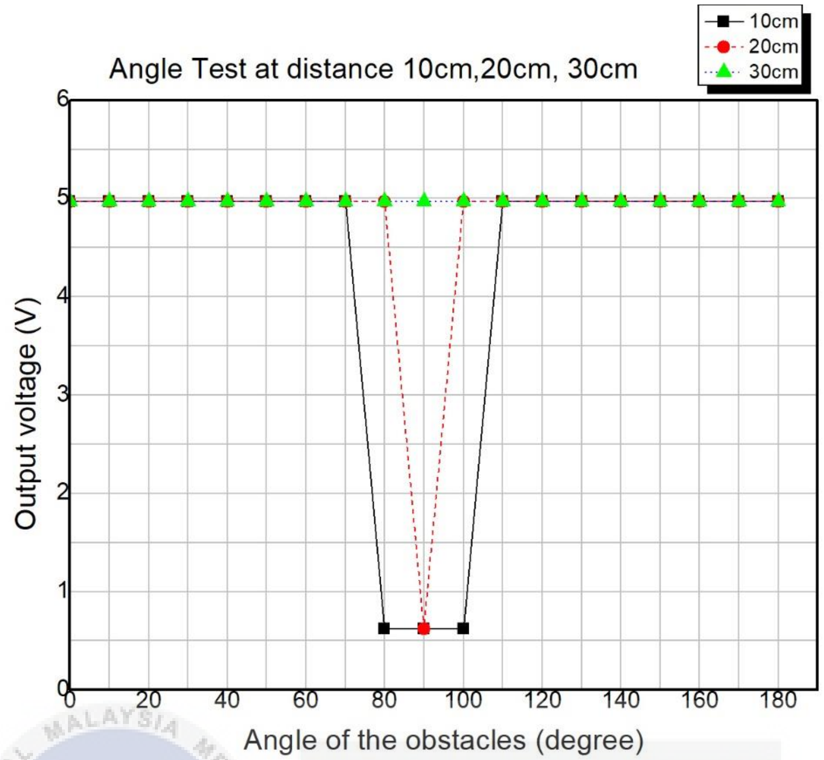

Figure 4.2: The Graph Output Voltage versus Angle of the obstacles

Therefore, based on the graph in Figure 4.1 and graph in Figure 4.2, the value of output voltage when IR sensors detect an obstacle is 0.62V. However, when IR sensors cannot detect an obstacle, the output voltage is 4.97V. This is because the IR sensors work by returning voltage directly proportional to the distance of the obstacles detected. The IR sensors give better performance at a shorter distance with a larger change in voltage. Hence, the output voltage increase, as well as the distance of obstacle, detected increase.

4.2.2 Obstacle Avoidance

Obstacles avoidance experiment are being tested to evaluated the performance of the sensor of the stair climbing robot to avoid the obstacles at three different situations which are when obstacle at left (S1), obstacle at right (S2) and obstacles at both sides (S1 and S2) at distance 20 cm with different joystick position. The open loop experiment has been done by evaluated the motion of the Motor 1 (M1) and Motor 2 (M2) and ability of the robot to make the decision when avoid the obstacles. The data collected are presented in the table below.

Table 4.3 below shows the result of obstacle avoidance that has been tested with different position of joystick when obstacle at the left side (S1) at distance 20cm. When an obstacle at the left side, it means that only sensor 1 able to detected obstacle while sensor 2 not detected obstacle. From the data collected, when the joystick is pushed in a forward direction, M1 rotate in the clockwise direction and M2 rotate in the anti-clockwise direction. Actually, this direction makes the robot to move forward and once sensor 1 detected an obstacle, both of the motor were stopped from move forward. When the joystick is pushed in reverse position, M1 rotate anti-clockwise and M2 also rotate anti-clockwise that produce reverse condition and the robot reverse accordingly to prevent from hitting obstacles. In addition, when the joystick is pushed in the left position, M1 was rotate anti-clockwise and M2 rotate clockwise and it was stop immediately to prevent the robot from turning left once the sensor detected an obstacle at left side. However, when the joystick is pushed in the right direction, M1 rotate in a clockwise direction and M2 rotate in an anti-clockwise direction where this motion actually makes the forward motion, so that it prevents the robot from hitting the obstacle.

Therefore, the stair climbing robot is able to make a decision when the joystick is pushed in a different position when obstacle at left side. The robot is able to avoid an obstacle from turning left and moving forward but the robot can move freely in reverse and right condition. Furthermore, the robot is save from hitting obstacles at different joystick position.

Table 4.3: Obstacle at Left Side

Joystick Position	Motor 1	Motor 2	Hitting Obstacle
Forward	Stop	Stop	No
Reverse	Reverse	Reverse	No
Left	Stop	Stop	No
Right	Forward	Reverse	No

Based on Table 4.4 below, it shown the result of the robot when obstacle at the right side (S2) by evaluated with four different joystick position. When the joystick is pushed in a forward position, M1 rotate clockwise and M2 rotate anti-clockwise and this condition makes the robot to stop from moving forward. Next, the position of the joystick was changed to reverse position and M1 was rotate anti-clockwise and M2 rotate in a clockwise direction and the robot to move in reverse motion to avoid the obstacle. Then, the joystick was pushed in left position and direction of both motor was evaluated where M1 rotate in anti-clockwise and M2 rotate in clockwise where this condition makes the robot to turning left. Lastly, when the joystick is pushed in the right position, M1 and M2 was stopped and it prevent from turning right once it detected obstacle at the right side.

At this situation, the stair climbing robot was able to prevent itself from move forward or turning right when obstacle at the right side. However, the robot able to move in another direction such as reverse position and turning left.

Table 4.4: Obstacle at Right Side

Joystick Position	Motor 1	Motor 2	Hitting Obstacle
Forward	Stop	Stop	No
Reverse	Reverse	Reverse	No
Left	Reverse	Forward	No
Right	Stop	Stop	No

Last but not least, obstacles avoidance when obstacle at both side is the last situations that evaluated for obstacle avoidance. Table 4.5was shown the configuration of the robot when observed at different joystick position. When the joystick is pushed in a forward position, M1 rotate clockwise and M2 rotate anti-clockwise and this motion makes the robot move forward. Therefore, both motors were automatically stopped when sensor 1 and sensor 2 detected an obstacle at both sides. When the joystick is pushed in reverse position, M1 rotate anti-clockwise and M2 rotates clockwise and this makes the robot to reverse motion. Then, when the joystick is changed to the left position, M1 rotate anti-clockwise and M2 rotate clockwise and the robot was stopped or in other words it avoiding from turning left. Furthermore, when the joystick is pushed in the right position, M1 rotate clockwise and M2 rotate anti-clockwise and both motors were stopped from turning right. The

At the end of the experiment, the performance of the robot has been done analyzed. When the joystick is pushed in four different positions, the robot able to make the decision in order to avoid obstacle at both side and the robot is saved from hitting an obstacle. The robot is cannot move forward, turning right or turning left when obstacle at both sides but it can move reverse.

Table 4.5: Obstacle at Both Side

Joystick Position	Motor 1	Motor 2	Hitting Obstacle
Forward	Stop	Stop	No
Reverse	Reverse	Reverse	No
Left	Stop	Stop	No
Right	Stop	Stop	No

From the Table 4.3, Table 4.4 and Table 4.5 above, can conclude that the stair climbing robot is able to make a decision when sensor detect the obstacles at three different situations which are when obstacles at the left side, obstacle at right side and obstacles at both sides. When sensor 1 sense obstacle, the robot cannot move forward or turning left but it can move reverse and turn right. However, when sensor 2 sense obstacle at the right side, the robot cannot move forward and turn right while it can move in reverse and turn left. Furthermore, when both sensor sense obstacles at both side, it only allowed the robot to move reverse while in other direction robot are unable to moving forward, turning left and turning right.

Therefore, the open loop experiment for obstacles avoidance has been done by observed the configuration of the robot when the joystick is in forward, reverse, left and right position based on sensor readings. In addition, M1 and M2 are work effectively as well as the robot detected the obstacles. Obstacle avoidance is very important for the mobile robot to overcome unexpected obstacles.

4.2.3 Stair Climbing

The stair climbing experiment without load was analyzed and discussed based on the result. The robot has been tested in order to measure how long the robot takes the time to finish the climbing process. However, the data for the height and width of the stairs was collected first before doing the experiment. Therefore, the process of the stair climbing was depended on stair dimension which is (17.5cm x 25cm). The experiment was measured in two different gradients (10° and 20°) from the stairs to verify the time taken of the robot take to finish the climbing process.

In order to get a better result, the number of steps which is two-step remain the same for every gradient of the stairs. This is actually to make sure there are no issues when getting the data for the climbing process. Unfortunately, there is a problem when running the experiment and noticed that the robot cannot climb the stairs due to some problems. Therefore, Table 4.6 shows the result of the robot cannot climb the stairs where is no time taken is recorded.

Table 4.6: Time Taken for Stair Climbing

Gradient	Time taken				Success
	1	2	3	Average	
10	0.00	0.00	0.00	0.00	NO
20	0.00	0.00	0.00	0.00	NO

The problem of the robot cannot climb the stairs was identified. The problem is delta wheel mechanism cannot function very well that affect the robot unable to climb the stairs as shown in Figure 4.3.

Figure 4.3: The problem of Delta-Wheel frame

However, these problems happened are actually because of the slippage happened on the wheels. In addition, DC motor has provided enough torque to push the robot because this type of motor comes with high torque motor but the wheel was wear and tear as shown in Figure 4.4 below. Therefore, the wheels cannot have provided enough friction to push itself to move forward and start climbing the stairs. Lack of friction makes the wheels cannot grip the floor and it also makes the wheels keep rotating at the same place.

Figure 4.4: Wear and Tear of the wheel robot

4.3 Fuzzy Logic Controller

In this research, Fuzzy Logic Controller has been done designed. The FLC is implement to navigate the motion of stair climbing robot in term of obstacle avoidance. Based on chapter 3, the method to design fuzzy logic controller for obstacle avoidance has discussed. The controller is design by using Multiple-Input Multiple-Output (MIMO) by using Mamdani method.

The fuzzy input and output should obey the relationship between distance and motion of the robot. Based on Figure 4.5, it shown the result of the Graphical User Interfaces for the system. There are two input which are left sensor and right sensor and for the output are left motor and right motor.

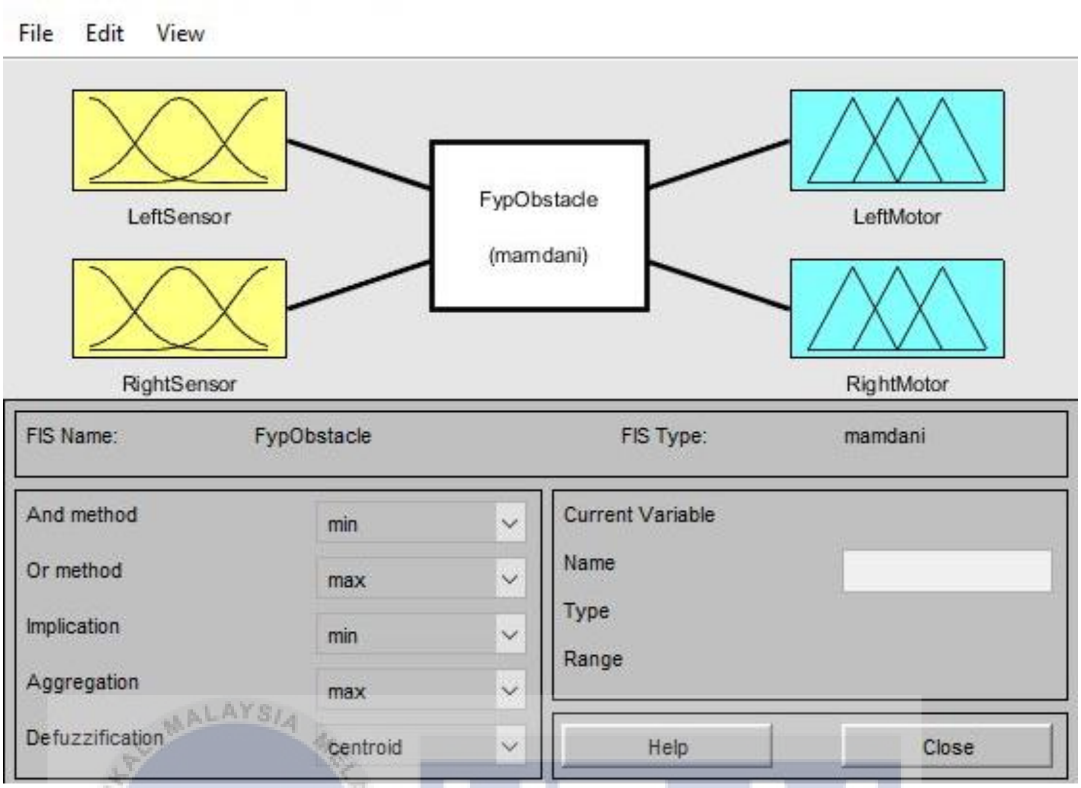

Figure 4.5: Fuzzy Logic Controller Graphical Users Interfaces

After Fuzzy Logic Controller Graphical Users Interfaces has designed, the membership function of each variable was constructed. Figure 4.6 and Figure 4.7 shown the result of the Fuzzy Inference System (FIS) of input membership function for both sensors. The range of both sensor is the distance between obstacles and the robot and it has two type of membership functions. The membership function used are combination between Z-Shaped and Sigmoidal and the intersection point between this two membership functions was called "Threshold". The Z-Shaped (zmf) of first membership function is chosen for "Near" because the robot needed to make decision to turn or reverse when it detect obstacle less than threshold value which is 27 cm. The Sigmoidal (sigmf) is chosen for membership function "Far" because the robot no need to turn or reverse when the distance more than threshold value because it was considered far. The range of every membership need to tune manually until getting the better performance after tested on the robot.

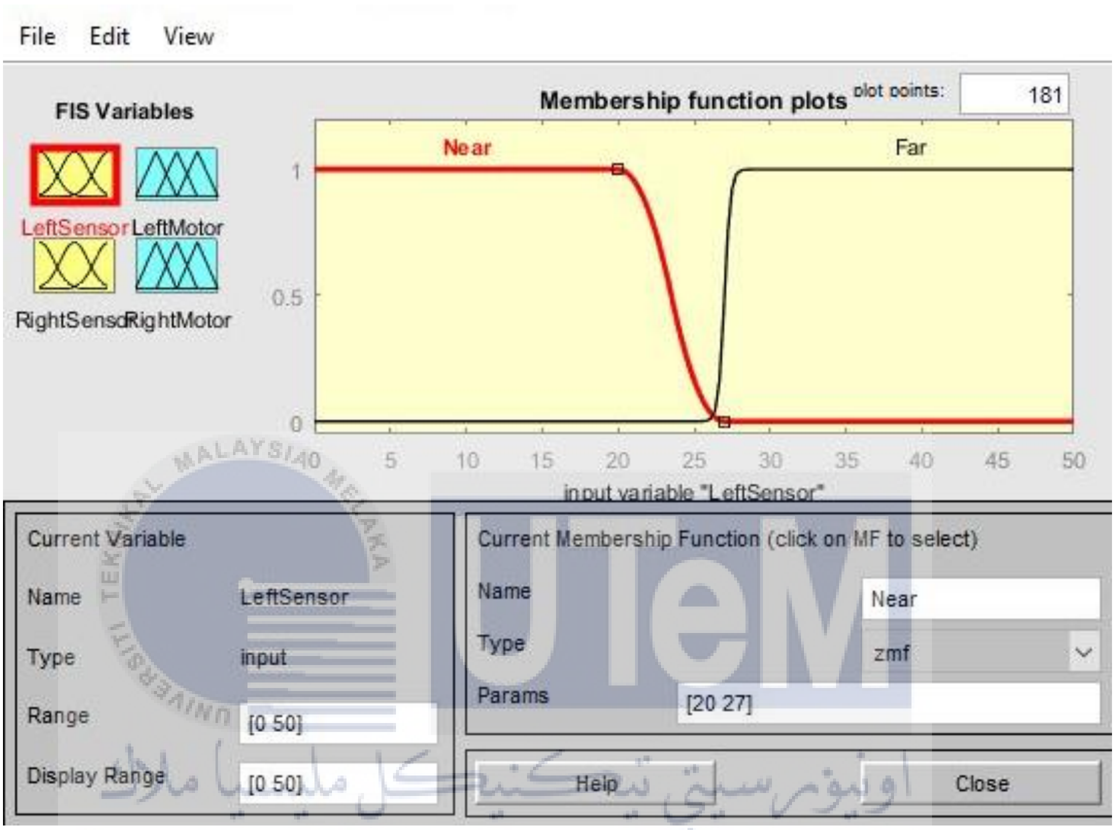

Figure 4.6: Membership Function of Left Sensor

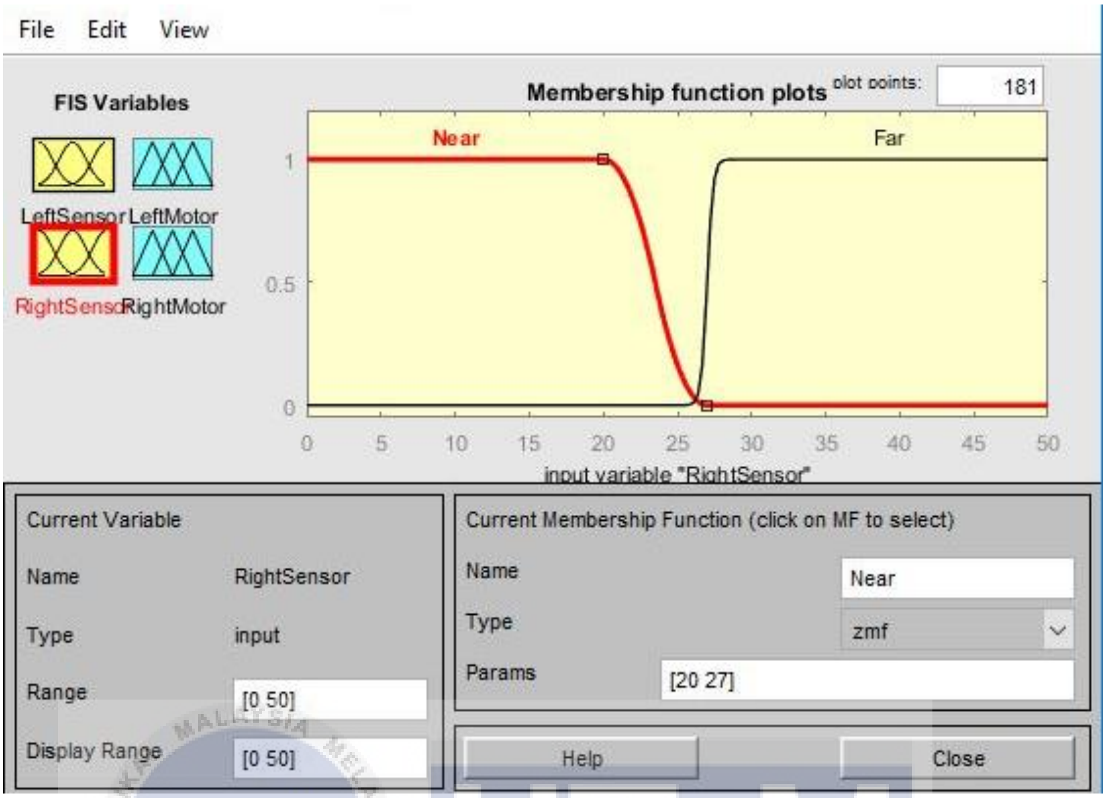

Figure 4.7: Membership Function of Right Sensor

Then, the membership functions of the output have constructed. The output of the FLC was measured in PWM or in other word it is the speed when control the motion of the robot. The output consists of two type of membership functions which are reverse and forward. Figure 4.8 and Figure 4.9 below shows the results output of the membership function for both motor. The output is to analyze reacting of the both motor when detected the obstacles. Membership functions of the reverse movement was designed by using Z-Shaped type of membership functions. The Z-Shaped is chosen for the reverse movement because for the mobile robot, the robot required to decrease the speed based on fuzzy logic rules. The membership functions for forward movement is designed using S-Shaped because when robot moving forward, it required to increase the speed.

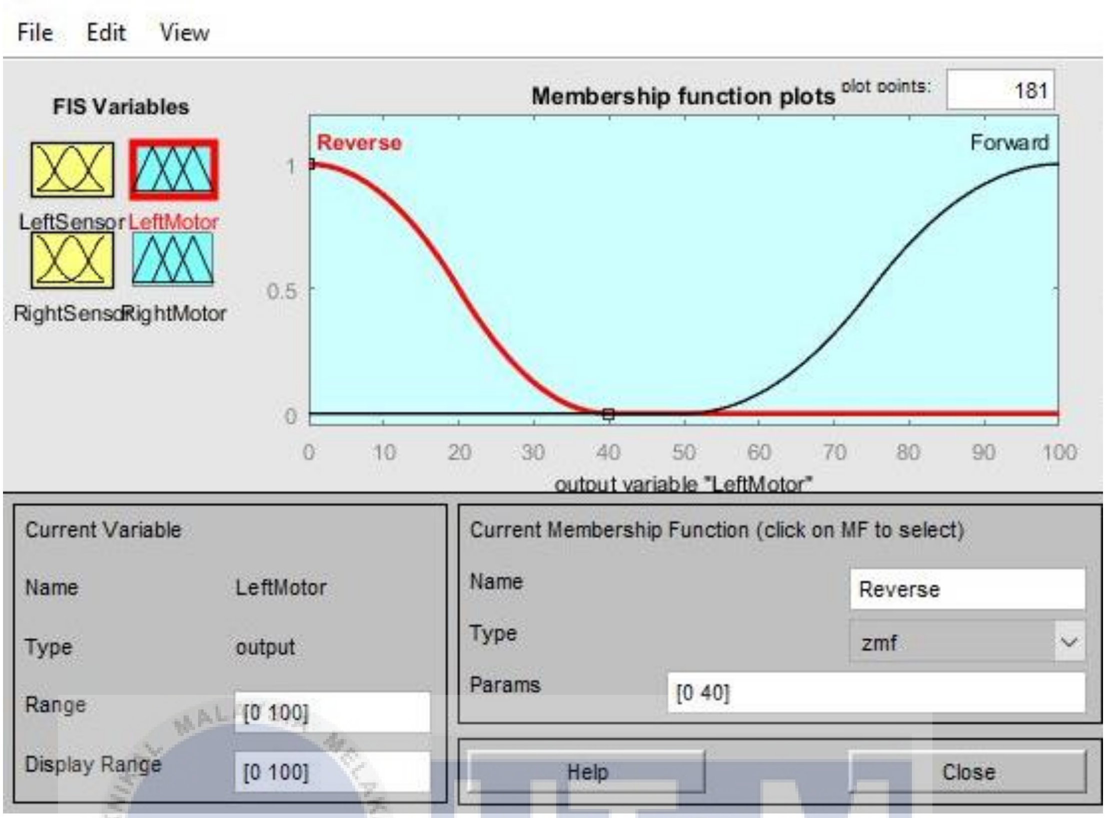

Figure 4.8: Membership function of Left Motor Speed

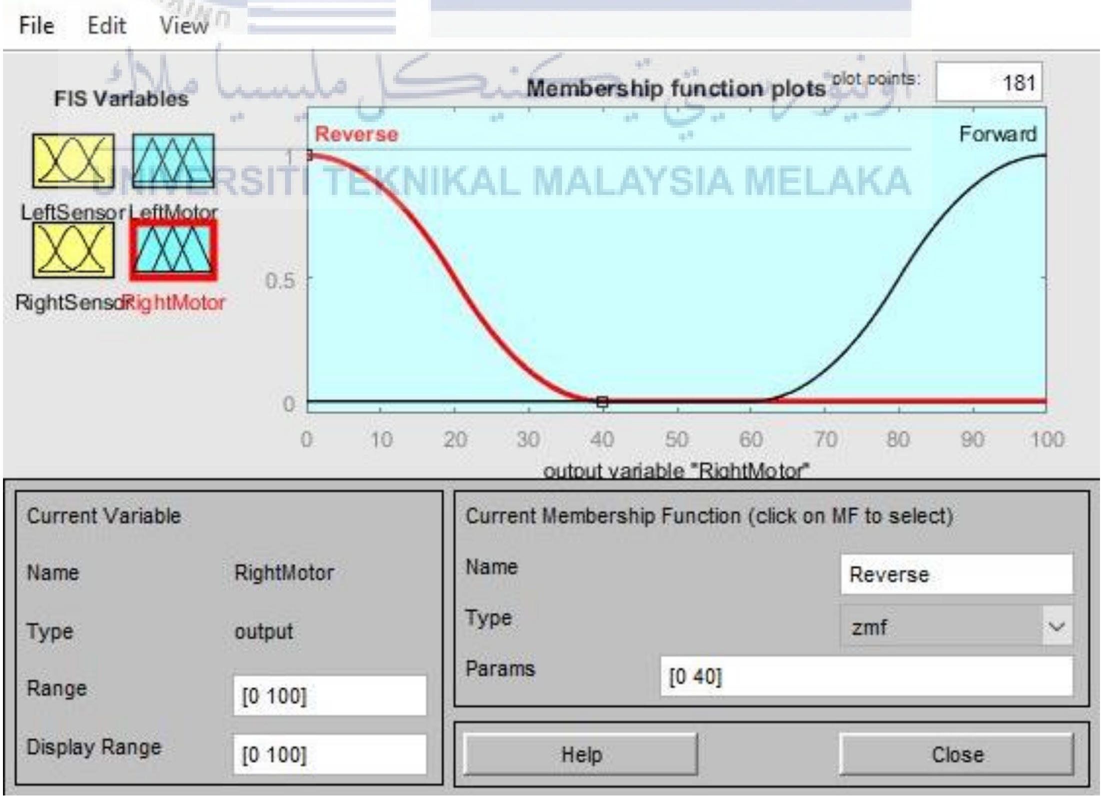

Figure 4.9: Membership Function of Right Motor Speed

Based on Figure 4.10 below, the result of the rule editor was shown. The movement of the robot are based on this simple rule of Fuzzy Logic by using If – Then Rules.

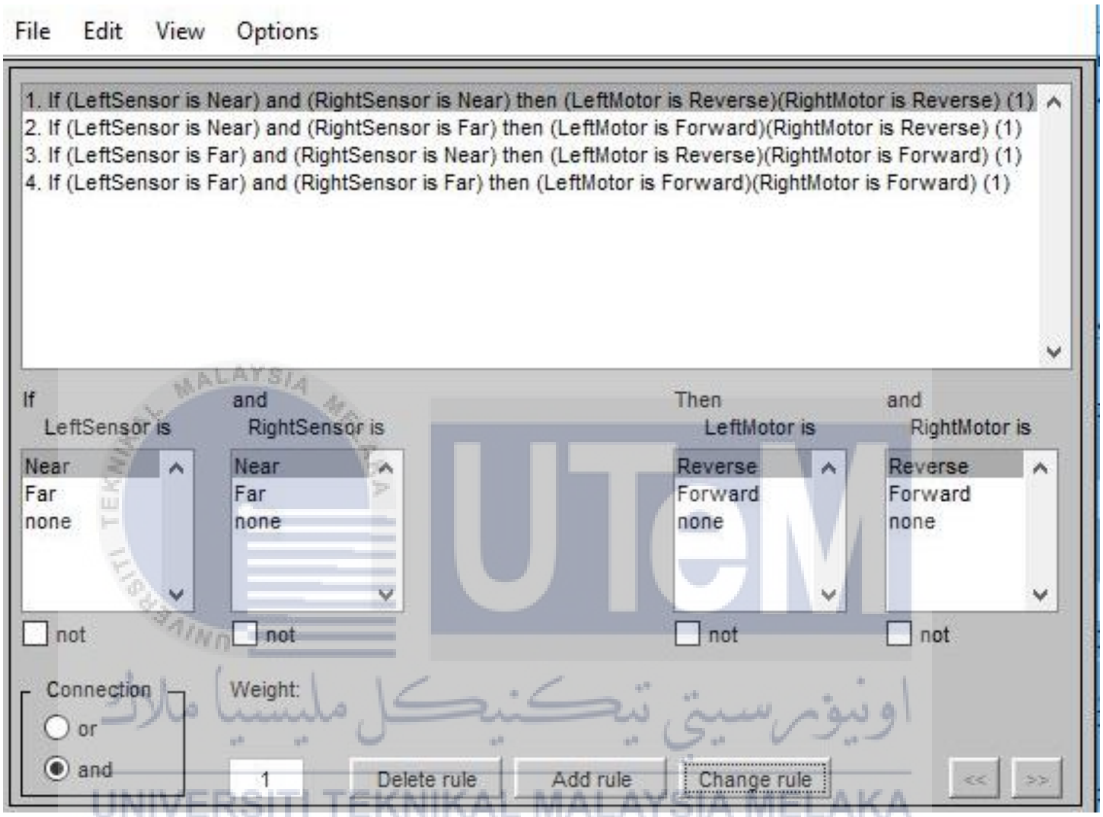

Figure 4.10: Rule Editor

Figure 4.11 and Figure 4.12 below shows the result of Surface Viewer in three axis graph which are x-axis (left sensor), y-axis (right sensor) and z-axis (left motor or right motor). The result of surface viewer was depending on the value of inputs. Different value of the both input will effect the output value as well as surface viewer condition. The surface viewer for both motor is the same because the parameters used for both motor are same.

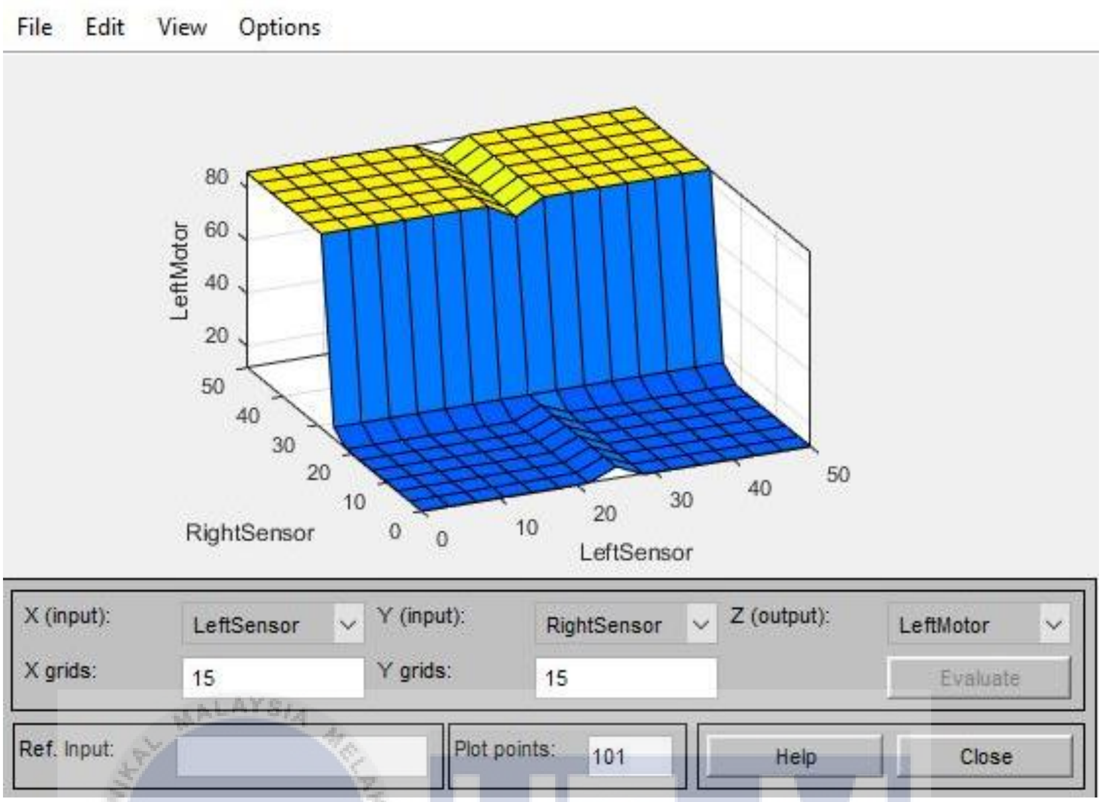

Figure 4.11: Surface Viewer of Left Motor

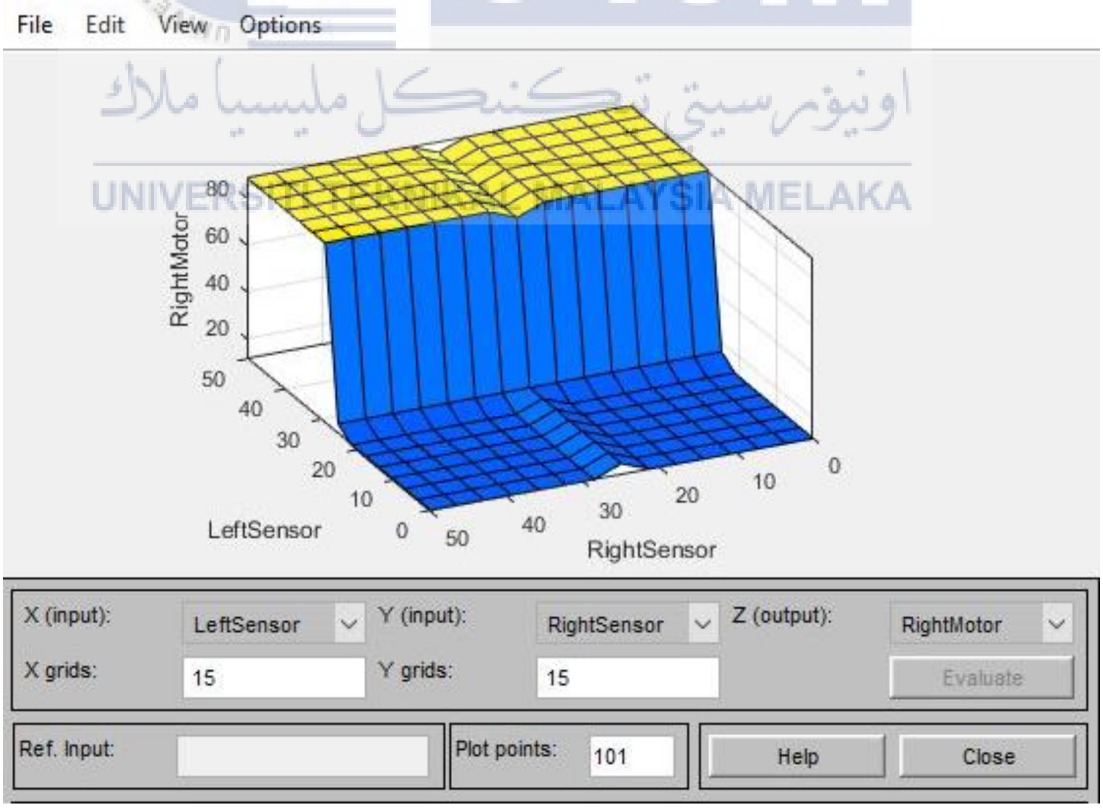

Figure 4.12: Surface Viewer of Right Motor

4.4 Robot System

The performace of the robot has been obeserved in real time for controller feedback. The result of the robot system is discuss in sub section below.

4.4.1 Actual Response of the Robot

The program of Fuzzy Logic Controller rules was created and it converted into Arduino program. This section was discussed about actual response of the robot after implemented FLC. The push button is pressed to start the robot and it read the sensor distance towards the obstacles. Robot received several data signal from the sensors and it was implement into Fuzzy Logic rules. Then, robot was works according to the fuzzy rules and the motion of the robot are depending on the output.

Figure shown the experiment environment of obstacle avoidance. When push button is pressed and automatically robot move forward. If sensor read the obstacles is far, the left motor and right motor moving forward in fast condition until it approaching the obstacles. At this condition, the both motor move reverse in slow motion when both sensors detect obstacles. After the reverse process is complete, the sensor read again for the next process. If left sensor detected obstacles and right sensor not detected any obstacles, then robot not able to turn because of the structure of the tires.

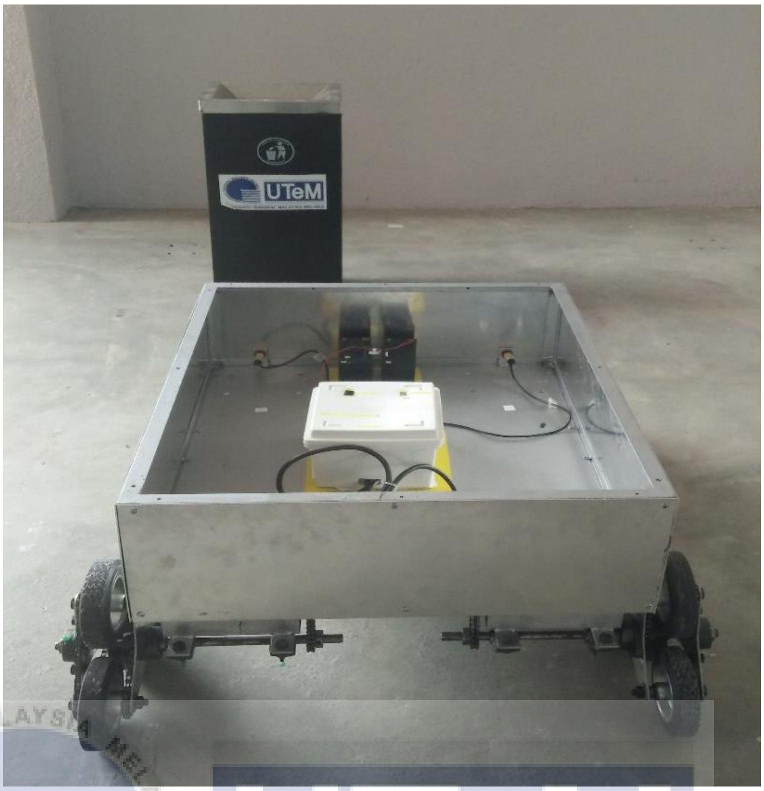

Figure 4.13: Experiment environment of obstacle avoidance

4.4.2 Behavior of the robot during avoiding obstacles

The robot has two delta-wheel powered by separate DC motors and the sensor are placed at front corners of the robot. The sensors are to detect the obstacle at left and right side. The distance between the robot and obstacles act as repulsive forces for avoiding the obstacles. The behavior of the robot was analyzed during avoiding the obstacles based on fuzzy rules and show the behavior of the robot when it moves on straight motion. However, the robot facing quite difficulty to make turning process because of the structure of tires. This issues were analyzed when the robot changed it motion to turning left or turning right.

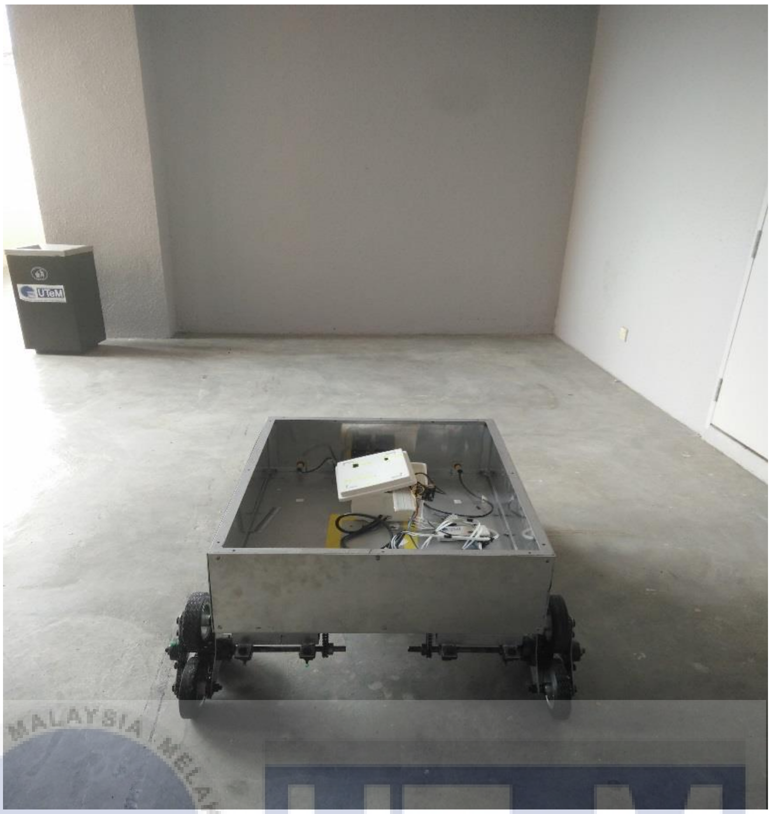

Figure 4.14: Robot moving forward when sensor is far

Figure 4.15: Robot moving reverse when both sensor is near

4.5 Summary

At the end of this chapter, all of the result for experiment was discussed. Obstacle detection experiment were divided into two test which are distance test and angle test. The IR sensor able detected the obstacles to control the robot by given different output voltage. For obstacle avoidance experiment, stair climbing robot is able to make a decision when sensor detect the obstacles at three different situations. Stair climbing experiment is not succeed because the robot unable to climb the stair due to some issues. Lastly, the Fuzzy Logic Controller have been design and implemented into the system. Then, the robot able to work accordingly to Fuzzy Logic system by applied rules that created.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The first objective of this Final Year Project is to develop open loop test for obstacle detection, obstacles avoidance and stair climbing. From previous result discussed in chapter 4, the experiment have been conducted and all the data of the experiment was collected and constructed in the table. Therefore, objective one is achieved.

The second objective is to design Fuzzy Logic Controller (FLC) for stair climbing mechanism in order to traverse obstacle or movement in uneven terrain. The Fuzzy Logic Controller is design by using MATLAB Toolbox for Multiple-Input Multiple-Output (MIMO). All the step and result of membership function have been discussed in Chapter 3 and Chapter 4. Therefore, objective two is successfully achieved.

The third objective is to analyze the performance of stair climbing mechanism in term of avoiding obstacles. From the second objective, the Fuzzy Logic Controller have been designed. Hence, the FLC was implemented into the system to tested the performance obstacle avoidance when traverse obstacles. The robot have been successfully avoid the obstacles by translate the sensor measurement to the DC motor and the result have been discussed in chapter 4. Therefore, third objective is achieved.

As a conclusion, all of the objective for this project have been successfully achieved. The future work of this project was discussed in next subtopic in order to make improvement of the system.

5.2 Recommendation

Every project or task for mobile robot can be improved to make it works more better than before. For this project, there are several suggestion can be used to improved the movement of the robot.

First suggestion is to change the material of the tires used. Based on open loop test stair climbing, the robot is used solid rubber tires and it look like not enough effective. It is because the solid rubber tires mostly have higher rolling resistance and it give roughest ride over bumps. Therefore, solid rubber tires can be replaced with air filled wheelchair tires because it give smooth motion especially when traverse on uneven terrain.

Other than that, the improvement of the robot structure can be make in order to give smooth movement when turning process. The attachement of the wheels can be design flexibility so that the robot can turning effectively. Besides that, the delta-wheel frame can be improved by making it more bigger in terms of radius. This improvement can give more clearance when climbing the stairs.

The Fuzzy Logic Controller with ultrasonic sensor or with more IR sensors can be suggested for this project in order to make it have better performance when traverse obstacles. By using ultraonic sensors, the system can be detect the obstacles up to 400cm.

REFERENCES

[1] L. Kraus, “2016 Disability Statistics Anual Report,” pp. 1–48, 2017.

[2] L. Bruzzone and G. Quaglia, “Review article: Locomotion systems for ground mobile robots in unstructured environments,” *Mech. Sci.*, vol. 3, no. 2, pp. 49–62, 2012.

[3] G. Quaglia and M. Nisi, “Design of a self-leveling cam mechanism for a stair climbing wheelchair,” *Mech. Mach. Theory*, vol. 112, no. January, pp. 84–104, 2017.

[4] D. Kim, H. Hong, H. S. Kim, and J. Kim, “Optimal design and kinetic analysis of a stair-climbing mobile robot with rocker-bogie mechanism,” *Mech. Mach. Theory*, vol. 50, pp. 90–108, 2012.

[5] B. Chen, R. Wang, Y. Jia, L. Guo, and L. Yang, “Design of a high performance suspension for lunar rover based on evolution,” *Acta Astronaut.*, vol. 64, no. 9–10, pp. 925–934, 2009.

[6] R. Siegwart, P. Lamon, T. Estier, M. Lauria, and R. Piguet, “Innovative design for wheeled locomotion in rough terrain,” *Rob. Auton. Syst.*, vol. 40, no. 2–3, pp. 151–162, 2002.

[7] M. Sato, A. Kanda, and K. Ishii, “Simultaneous Optimization of a Wheeled Mobile Robot Structure and a Control Parameter,” *Simulation*, pp. 230–235, 2009.

[8] A. Gonzalez, E. Ottaviano, and M. Ceccarelli, “On the kinematic functionality of a four-bar based mechanism for guiding wheels in climbing steps and obstacles,” *Mech. Mach. Theory*, vol. 44, no. 8, pp. 1507–1523, 2009.

[9] Y. Sugahara *et al.*, “Walking up and down stairs carrying a human by a biped

locomotor with parallel mechanism," *2005 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, no. 16, pp. 3425–3430, 2005.

[10] E. Z. Moore, D. Campbell, F. Grimminger, and M. Buehler, "Reliable Stair Climbing in the Simple Hexapod ' RHex ,'" no. September 2014, 2002.

[11] M. J. Lawn, T. Sakai, M. Kuroiwa, and T. Ishimatsu, "Development and practical application of a stairclimbing wheelchair in Nagasaki," 1999.

[12] W. Tao, J. Xu, and T. Liu, "Electric-powered wheelchair with stair-climbing ability," no. August, pp. 1–13, 2017.

[13] Jianjun Yuan and S. Hirose, "Research on leg-wheel hybrid stair-climbing robot, Zero Carrier," pp. 654–659, 2005.

[14] H. Wang, L. He, Q. Li, W. Zhang, D. Zhang, and P. Xu, "Research on a kind of leg-wheel stair-climbing wheelchair," *2014 IEEE Int. Conf. Mechatronics Autom. IEEE ICMA 2014*, pp. 2101–2105, 2014.

[15] T. S. Wheelchair, I. Stair, D. Measurement, and D. Sensors, "Young-bong Bang\ Chang-hyuk Lee2, Je-hong Yoo\ Kyung-min Lee3 and In-su Kim3," pp. 1788–1791, 2011.

[16] S. Nakajima, "Development of four-wheel-type mobile robot for rough terrain and verification of its fundamental capability of moving on rough terrain," *2008 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2008*, pp. 1968–1973, 2008.

[17] G. Quaglia, W. Franco, and M. Nisi, "Politecnico di Torino Porto Institutional Repository [Proceeding] Evolution of Wheelchair.q, a Stair-climbing Wheelchair Evolution of Wheelchair.q, a Stair-climbing Wheelchair," no. 2015, 2015.

[18] S. Yu, T. Wang, Y. Wang, D. Zhi, C. Yao, and Z. Wang, "A Tip-Over and Slippage Stability Criterion for Stair- Climbing of a Wheelchair Robot with

Variable Geometry Single Tracked Mechanism," no. June, pp. 88–93, 2012.

[19] S. Yu, T. Wang, X. Li, C. Yao, Z. Wang, and D. Zhi, "Configuration and Tip-Over Stability Analysis for Stair-Climbing of a New-Style Wheelchair Robot," pp. 1387–1392, 2010.

[20] A. A. Nwe, W. P. Aung, and Y. M. Myint, "Software Implementation of Obstacle Detection and Avoidance System for Wheeled Mobile Robot," vol. 88704, pp. 572–577, 2008.

[21] K. A. Mahadhir, C. Y. Low, H. Hamli, A. Jaffar, E. Salleh, and M. S. Bahari, "Stair Climbing of a Track-Driven Mobile Robot with Flipper Arm," *Appl. Mech. Mater.*, vol. 393, pp. 586–591, 2013.

[22] V. T. Pham, Q. Qiu, A. A. P. Wai, and J. Biswas, "Application of ultrasonic sensors in a smart environment," *Pervasive Mob. Comput.*, vol. 3, no. 2, pp. 180–207, 2007.

[23] S. R. I. Confere, A. N. K. Nasir, and M. O. Tokhi, "D-Fuzzy Control of a Stai ir Climb bing Whe Has," vol. 4, pp. 18–25, 2013.

[24] M. M. Sabir and J. A. Khan, "Optimal Design of PID Controller for the Speed Control of DC Motor by Using Metaheuristic Techniques," *Adv. Artif. Neural Syst.*, vol. 2014, pp. 1–8, 2014.

[25] S. Temich and D. Grzechca, "ScienceDirect A fuzzy logic controller to seat comfort correction fuzzy logic controller to seat comfort correction for A fuzzy logic controller to seat comfort correction for manually manually propelled propelled wheelchair study A fuzzy logic controller," *IFAC-PapersOnLine*, vol. 49, no. 25, pp. 55–60, 2016.

[26] E. Mihankhah, A. Kalantari, E. Aboosaeedan, H. D. Taghirad, and S. A. A. Moosavian, "Autonomous staircase detection and stair climbing for a tracked

mobile robot using fuzzy controller," *2008 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2008*, pp. 1980–1985, 2008.

[27] Q. Li, P. Wang, and E. S. Lee, "R-Fuzzy Sets," *Comput. Math. with Appl.*, vol. 31, no. 2, pp. 49–61, 1996.

[28] S. Ahn, Y. H. Kim, W. H. Kwon, and C. Information, "Design of," pp. 598–602, 1996.

[29] C. N. Bhende, S. Mishra, and S. K. Jain, "TS-fuzzy-controlled active power filter for load compensation," *IEEE Trans. Power Deliv.*, vol. 21, no. 3, pp. 1459–1465, 2006.

[30] M. Wang, Y. Kung, and Y. Tu, "Fuzzy Logic Control Design for a Stair-Climbing Robot," vol. 11, no. 3, pp. 174–182, 2009.

[31] B. K. Patle, D. R. K. Parhi, A. Jagadeesh, and S. K. Kashyap, "Application of probability to enhance the performance of fuzzy based mobile robot navigation," *Appl. Soft Comput. J.*, vol. 75, pp. 265–283, 2019.

APPENDICES

APPENDIX A PROJECT GANTT CHART FYP1 And FYP2

Table A: The Gantt Chart for Final Year Project 1 (FYP 1)

Project Activities	2018													
	SEPTEMBER				OCTOBER				NOVEMBER				DECEMBER	
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Weekly Progress Meeting	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Research of Stair Climbing Robot	■	■	■	■	■	■	■	■	■	■	■	■		
Study About Experimental Setup			■	■	■	■	■							
Develop Obstacle Detection Experiment						■	■	■	■					
Data Collection and Analysis							■	■	■					
Report Writing					■	■	■	■	■	■	■	■		
FYP 1 Seminar											■			
Edit Final Report													■	
Submission Final Report														■

Table B: The Gantt Chart for Final Year Project 2 (FYP 2)

Project Activities	2019													
	FEBRUARY		MARCH				APRIL				MAY			
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Weekly Progress Meeting	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Continue Research from FYP 1	■	■	■	■	■	■	■	■	■	■	■	■	■	
Literature Review for Controller	■	■	■											
Design Fuzzy Logic Controller		■	■	■	■									
Develop Fuzzy Logic Controller for Stair Climbing Robot					■	■	■	■						
Data Collection and Analysis						■	■	■						
Discussion and Suggestion for Improvement									■	■				
Report Writing		■	■	■	■	■	■	■	■	■	■	■	■	
FYP 2 Seminar												■		
Submission Final Report														■

APPENDIX B SOURCE CODE FOR IR SENSORS

```
#define SENSOR 2 // define pin 2 for sensor
#define ACTION 9 // define pin 9 as for ACTION
void setup() {
  Serial.begin(9600);// setup Serial Monitor to display information
  pinMode(SENSOR, INPUT_PULLUP);// define pin as Input  sensor
  pinMode(ACTION, OUTPUT);// define pin as OUTPUT for ACTION
}
void loop() {
  int L =digitalRead(SENSOR);// read the sensor
      if(L == 0){
    Serial.println(" Obstacle detected");
    digitalWrite(ACTION,HIGH);// send signal
      }
else{
     Serial.println("  === All clear");
     digitalWrite(ACTION,LOW);// turn the relay OFF
      }
  delay(500);
}
```

APPENDIX C SOURCE CODE FOR OBSTACLE AVOIDANCE

```
// Motor A

int enA = 5;
int in1 = 4;

// Motor B

int enB = 6;
int in3 = 7;

// Joystick Input

int joyVert = A0; // Vertical
int joyHorz = A1; // Horizontal

// SENSOR INPUT

const int sensorfront1 = 8;
const int sensorfront2 = 9;
const int sensorback1 = 10;
const int sensorback2 = 11;

// Motor Speed Values - Start at zero

int MotorSpeed1 = 0;
int MotorSpeed2 = 0;

// Joystick Values - Start at 512 (middle position)

int joyposVert = 512;
int joyposHorz = 512;


void setup()

{
  // Set all the motor control pins to outputs

 pinMode(enA, OUTPUT);
 pinMode(enB, OUTPUT);
 pinMode(in1, OUTPUT);
 pinMode(in3, OUTPUT);
```

```
// SET ALL THE SENSOR PINS TO INPUTS

 pinMode(sensorfront1, INPUT);
 pinMode(sensorfront2, INPUT);
 pinMode(sensorback1, INPUT);
 pinMode(sensorback2, INPUT);

// Start with motors disabled and direction forward

 // Motor A

 digitalWrite(enA, LOW);
 digitalWrite(in1, HIGH);


 // Motor B

 digitalWrite(enB, LOW);
 digitalWrite(in3, HIGH);


}

void loop()
{



  // Read the Joystick X and Y positions

 joyposVert = analogRead(joyVert);
 joyposHorz = analogRead(joyHorz);

 // Determine if this is a forward or backward motion
 // Do this by reading the Verticle Value
 // Apply results to MotorSpeed and to Direction

 if (digitalRead(8) == LOW  && digitalRead(9) == LOW)
 { // if BOTH the switches read HIGH

    motorstop ();
  // delay(1000);
    controllbackward ();
  }
 if (digitalRead(8) == LOW  && digitalRead(9) == HIGH)
 { // // obstacle at LEFT turn RIGHT

    motorstop ();
   // delay(1000);
```

```
    controllright ();
    controllbackward ();

  }

  if (digitalRead(8) == HIGH  && digitalRead(9) == LOW)
  { // obstacle at right turn left


    motorstop ();
    //delay(1000);
    controllbackward ();
    controllleft ();

  }

  if (digitalRead(8) == HIGH  && digitalRead(9) == HIGH)
  {
   controllforward ();
   controllbackward ();
   controllright ();
   controllleft ();
  }

/*
   if (digitalRead(10) == LOW  && digitalRead(11) == LOW)
  { // if BOTH the switches read HIGH

    motorstop ();
    //delay(1000);
    controllforward ();
  }
  if (digitalRead(10) == LOW  && digitalRead(11) == HIGH)
  { // // obstacle at LEFT turn RIGHT

    motorstop ();
    //delay(1000);
    controllright ();
    controllforward ();
  }
  if (digitalRead(10) == HIGH  && digitalRead(11) == LOW)
  { // obstacle at right turn left
    motorstop ();
    //delay(1000);
    controllforward ();
    controllleft ();
  }
  if (digitalRead(10) == HIGH  && digitalRead(11) == HIGH)
  {
   controllforward ();
```

```
    controllbackward ();
    controllright ();
    controllleft ();
  }
*/
}
void controllbackward ()
 {
  if (joyposVert > 564)
  {
    // This is backward
    // Set Motor A backward
    digitalWrite(in1, HIGH);
    // Set Motor B backward
    digitalWrite(in3, LOW);
     //Determine Motor Speeds
     MotorSpeed1 = ((joyposVert-564)*80)/(1023-564);
     MotorSpeed2 = ((joyposVert-564)*80)/(1023-564);

   }
    // Set the motor speeds
  analogWrite(enA, MotorSpeed1);
  analogWrite(enB, MotorSpeed2);
 }

void controllforward ()
{
  if (joyposVert < 460)
  {
    // This is forward

    // Set Motor A backward

    digitalWrite(in1, LOW);

    // Set Motor B forward

    digitalWrite(in3, HIGH);
    //Determine Motor Speeds

    // As we are going forwards we need to reverse readings
    joyposVert = joyposVert - 460; // This produces a negative number
    joyposVert = joyposVert * -1;  // Make the number positive
    MotorSpeed1 = (joyposVert *80)/(460);
    MotorSpeed2 = (joyposVert *80)/(460);
  }
  else
  {
    // This is Stopped
     MotorSpeed1 = 0;
```

```
      MotorSpeed2 = 0;
    }

    // Set the motor speeds
  analogWrite(enA, MotorSpeed1);
  analogWrite(enB, MotorSpeed2);
}

void controllright ()
{
  if (joyposHorz > 564)
  {
    // This is RIGHT
    // Set Motor A RIGHT
    digitalWrite(in1, LOW);
     // Set Motor B RIGHT
    digitalWrite(in3, LOW);
     //Determine Motor Speeds
    MotorSpeed1 = ((joyposHorz-564)*80)/(1023-564);
    MotorSpeed2 = ((joyposHorz-564)*80)/(1023-564);
  }
  // Set the motor speeds
  analogWrite(enA, MotorSpeed1);
  analogWrite(enB, MotorSpeed2);
}


void controllleft ()
{
  if (joyposHorz < 460)
  {
    // This is left
     // Set Motor A left
    digitalWrite(in1, HIGH);
    // Set Motor B left
    digitalWrite(in3, HIGH);
    //Determine Motor Speeds
    // As we are going LEFT we need to reverse readings
    joyposHorz = joyposHorz - 460; // This produces a negative number
    joyposHorz = joyposHorz * -1;  // Make the number positive
    MotorSpeed1 = (joyposHorz *80)/(460);
    MotorSpeed2 = (joyposHorz *80)/(460);
    }
  // Set the motor speeds
  analogWrite(enA, MotorSpeed1);
  analogWrite(enB, MotorSpeed2);
}
void motorstop ()
{
   MotorSpeed1 = 0;
```

```
  MotorSpeed2 = 0;
   // Set the motor speeds
  analogWrite(enA, MotorSpeed1);
  analogWrite(enB, MotorSpeed2);
}
```

APPENDIX D SOURCE CODE FOR FUZZY LOGIC CONTROLLER

```
#include "fis_header.h"
// Motor A
int enA = 5;
int in1 = 4;
// Motor B
int enB = 6;
int in3 = 7;

// SENSOR INPUT
const int sensorfront1 = 8;
const int sensorfront2 = 9;
const int sensorback1 = 10;
const int sensorback2 = 11;

//--- Motor variable
int p1; // left motor speed in pwm pulses 0-80
int p2; // right motor speed in pwm pulses 0-80

// 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 = 2;
// Number of rules to the fuzzy inference system
const int fis_gcR = 7;

FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
void setup()
{
   // initialize the Analog pins for input.
   // Pin mode for Input: LeftSensor
   pinMode(0 , INPUT);
   // Pin mode for Input: RightSensor
   pinMode(1 , INPUT);
   // initialize the Analog pins for output.
   // Pin mode for Output: LeftMotor
   pinMode(2 , OUTPUT);
   // Pin mode for Output: RightMotor
   pinMode(3 , OUTPUT);
}
void loop()
{
  SensorRead(Sensorfront1, SensorFront2);
   // Read Input: LeftSensor
```

```
   g_fisInput[0] = Sensorfront1; // Sensorfront1
   // Read Input: RightSensor
   g_fisInput[1] = Sensorfront2; // Sensorfront2

   g_fisOutput[0] = 0;
   if (Sensorfront1>=25)
   {
   fis_evaluate();
   //--- Set output value: LeftMotor
   p1 =  g_fisOutput[0]; // p1
   //--- Set output value: RightMotor
   p2 =  g_fisOutput[0]; // p2
   if (p1>
   fis_evaluate();
   // Set output vlaue: LeftMotor
   analogWrite(2 , g_fisOutput[0]);
   // Set output vlaue: RightMotor
   analogWrite(3 , g_fisOutput[1]);
}
//***********************************************************************
****
// Support functions for Fuzzy Inference System
//***********************************************************************
****
// Triangular Member Function
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)
{
   FIS_TYPE a = p[0], b = p[1], c = p[2];
   FIS_TYPE t1 = (x - a) / (b - a);
   FIS_TYPE t2 = (c - x) / (c - b);
   if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
   if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
   if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
   t1 = min(t1, t2);
   return (FIS_TYPE) max(t1, 0);
}
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_trimf
};
// Count of member function for each Input
int fis_gIMFCount[] = { 2, 2 };
// Count of member function for each Output
int fis_gOMFCount[] = { 3, 3 };
// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { -20, 0, 25 };
FIS_TYPE fis_gMFI0Coeff2[] = { 15, 50, 60 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2 };
FIS_TYPE fis_gMFI1Coeff1[] = { -20, 0, 25 };
FIS_TYPE fis_gMFI1Coeff2[] = { 15, 50, 60 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2 };
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff };

// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { -24, 0, 50 };
FIS_TYPE fis_gMFO0Coeff2[] = { 0, 40, 80 };
FIS_TYPE fis_gMFO0Coeff3[] = { 30, 80, 150 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2,
fis_gMFO0Coeff3 };
FIS_TYPE fis_gMFO1Coeff1[] = { -30, 0, 50 };
FIS_TYPE fis_gMFO1Coeff2[] = { 0, 40, 80 };
FIS_TYPE fis_gMFO1Coeff3[] = { 30, 80, 150 };
FIS_TYPE* fis_gMFO1Coeff[] = { fis_gMFO1Coeff1, fis_gMFO1Coeff2,
fis_gMFO1Coeff3 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff, fis_gMFO1Coeff };
// Input membership function set
int fis_gMFI0[] = { 0, 0 };
int fis_gMFI1[] = { 0, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1};
// Output membership function set
int fis_gMFO0[] = { 0, 0, 0 };
int fis_gMFO1[] = { 0, 0, 0 };
```

```
int* fis_gMFO[] = { fis_gMFO0, fis_gMFO1};
// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1 };
// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1 };

// Rule Inputs
int fis_gRI0[] = { 1, 1 };
int fis_gRI1[] = { 1, 1 };
int fis_gRI2[] = { 1, 2 };
int fis_gRI3[] = { 1, 2 };
int fis_gRI4[] = { 2, 1 };
int fis_gRI5[] = { 2, 1 };
int fis_gRI6[] = { 2, 2 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4, fis_gRI5,
fis_gRI6 };
// Rule Outputs
int fis_gRO0[] = { 1, 1 };
int fis_gRO1[] = { 2, 2 };
int fis_gRO2[] = { 1, 1 };
int fis_gRO3[] = { 3, 2 };
int fis_gRO4[] = { 1, 1 };
int fis_gRO5[] = { 2, 3 };
int fis_gRO6[] = { 3, 3 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5,
fis_gRO6 };
// Input range Min
FIS_TYPE fis_gIMin[] = { 0, 0 };
// Input range Max
FIS_TYPE fis_gIMax[] = { 50, 50 };

// Output range Min
FIS_TYPE fis_gOMin[] = { 0, 0 };
// Output range Max
FIS_TYPE fis_gOMax[] = { 80, 80 };
//*******************************************************************
****
// 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_RESOLUSION - 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_RESOLUSION; ++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 };
    FIS_TYPE fuzzyInput1[] = { 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0 };
    FIS_TYPE fuzzyOutput1[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, fuzzyOutput1, };
    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);
        }
    }
}
```