Faculty Of Electrical Engineering
Universiti Teknikal Malaysia Melaka

OBSTACLE AVOIDANCE IN INDOOR ENVIRONMENT USING SENSOR FUSION FOR MOBILE ROBOT

SYAZA NASUHA BINTI ZULLKIFLEE

Bachelor Degree of Mechatronics Engineering

2017

“ I hereby declare that I have read through this report entitle “ Obstacle Avoidance In Indoor Environment Using Sensor Fusion For Mobile Robot“ and found that it has comply the partial fulfilment for awarding the degree of Bachelor of Electrical Engineering (Mechatronic)”

Signature :………………………………………………………….

Supervisor’s Name : …NUR MAISARAH BT MOHD SOBRAN……..

Date : ….15/06/2017………………………………………..

OBSTACLE AVOIDANCE IN INDOOR ENVIRONMENT USING SENSOR FUSION FOR MOBILE ROBOT

SYAZA NASUHA BINTI ZULKIFLEE

A report submitted in partial fulfilment of the requirements for the degree of bachelor of Mechatronic Engineering

Faculty of Electrical Engineering
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

I declare that this report entitle *"Obstacle Avoidance In Indoor Environment Using Sensor Fusion For Mobile Robot"* is the result of my own research except as cited in the reference. The report has not been accepted for any degree and is not concurrently submitted in candidate of any other degree.

Signature : ……………………………………………….

Name : SYAZA NASUHA BINTI ZULKIFLEE

Date : 15/06/2017

To My Supervisor and Family

ACKNOWLEDGEMENT

Upon the completion of this report, there are many works involved that need the help of various people. They have guided me thoroughly upon understanding the concept of the project, the essence of knowledge provided will not be at waste. Particularly, I would like to express my biggest thanks to my main project supervisor, Miss Nur Maisarah Binti Mohd Sobran of her guidance for the project, suggesting ideas, help in understanding project purpose and concept and thoroughly guide me upon this whole project. Without her guidance, the project might unable to be implementing well and may not be able to succeed.

I am also very thankful to UTeM administration especially FKE administration for the funding of material of the project, helping much in providing the subject of the project.

Besides that, thanks to my fellow course mate that help giving friendly advice, suggestions, for me to complete the system well and help me on understanding some of the confusing part of the project.

ABSTRACT

Obstacle avoidance for mobile robot is one of the important ability to detect the presence of obstacle. Each of them are equipped with sensor such as ultrasonic sensor used to observe the surrounding environment. However to avoid the obstacle well, the distance between obstacle detected and sensor need to be measured considering the range of detection for each sensor. Furthermore, the position of the sensor on robot also influence the accuracy of obstacle detection. Three ultrasonic sensors are used to detect the obstacle in this research. The fuzzy rules incorporating Mamdani technique is implemented with three inputs which are the three ultrasonic sensors and two outputs which are left and right DC motor of the mobile robot's wheels which developed 27 fuzzy rules for controlling the robot's movement. The proposed methodology, which includes the obstacle avoidance based on fuzzy logic fusion, has been implemented and tested in real time experiments. Six situations have been presented using static obstacles and mobile robot while the robot is avoiding obstacles in different sizes. The success rate of avoiding static obstacles in all situations is at least 85%. The obstacle avoidance module is the most successful (with 100% success rate) when obstacle is at the left side of the robot and when obstacles are in front of the robot with small distance apart.

ABSTRAK

Pengelakan halangan untuk robot mudah alih adalah salah satu kebolehan yang penting untuk mengesan kewujudan halangan. Setiap daripada mereka dilengkapi dengan sensor seperti ultrasonik sensor yang digunakan untuk melihat persekitaran. Walau bagaimanapun untuk mengelak halangan dengan baik, jarak antara halangan yang dikesan dan sensor perlu diukur berdasarkan julat pengesanan untuk setiap sensor. Tambahan pula, kedudukan sensor pada robot juga mempengaruhi ketepatan dalam mengesan halangan. Tiga sensor ultrasonik digunakan untuk mengesan halangan dalam kajian ini. Peraturan fuzzy yang menggunakan teknik Mamdani menggunakan tiga input iaitu tiga sensor ultrasonik dan dua output iaitu DC motor kiri dan kanan pada tayar robot mudah alih menghasilkan 27 peraturan fuzzy untuk mengawal pergerakan robot. Kaedah yang dicadangkan, termasuklah pengelakan halangan berdasarkan gabungan logik fuzzy, telah dilaksanakan dan diuji dalam eksperimen sebenar. Enam situasi telah diperkenalkan menggunakan objek statik dan robot mudah alih sementara robot tersebut mengelak halangan yang mempunyai pelbagai saiz. Purata kejayaan pengelakan halangan statik dalam kesemua situasi adalah sekurang-kurangnya 85%. Modul pengelakan halangan yang paling berjaya (dengan purata kejayaan 100%) adalah apabila halangan berada di sebelah kiri robot dan apabila halangan di hadapan robot dengan jarak berjauhan yang kecil.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
DC	Direct Current
FLC	Fuzzy Logic Controller
IA	Intelligent Automation
IR	Infrared
MKR	Muratec Keio Robot
PWM	Pulse Width Modulation
USB	Universal Serial Bus

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Nowadays, there are many types of robots that are used in human environment. The robots have an ability to blend with dynamic and unknown environment such as an environment with the human movement, the place that surrounding with appliances and others without human involvement. There are also robots that are used in various applications, such as hospital, indoor security patrols, and materials handling in ware house.

One of the ability of those robots is obstacles avoidance. The obstacles avoidance ability is widely used for outdoor and indoor environment. It can be used in the variety area that have many object such as furniture, table, book shelf, the walking human, the moving object and many more.

A working example of the applications of obstacle avoidance ability is a MKR (Muratec Keio Robot), an autonomous omni-directional mobile transfer robot system for hospital applications. This robot used to transfer luggage, important specimens and other materials while maintaining a safe obstacle and collision avoidance that realizes a safe movement technology. The robot can distinguish people from others obstacles with human detection algorithm. The robot evades to people more safely by considering its relative position and velocity with respect to them. [1]

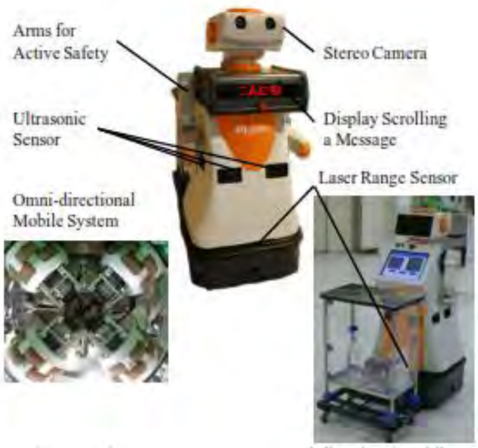


Figure 1.1 : MKR-003 *(Muratec Keio Robot) [1]

Besides that, obstacle avoidance ability is also popular in security system. For an example, a Intelligent Automation Robot (IA Robot) was developed with intelligent security architecture. In order to navigate IA Robot to complete mission with obstacle avoidance system and security system by using IR sensors, and USB Web-Camera installed in IA Robot. The IA Robot used to navigating and patrolling around the surrounding freely, and will avoid the obstacles if it meets the uncertain obstacles including static and dynamic obstacles. [2]

The obstacle avoidance ability on board a rescue robot which is a kind of surveillance robot that has been designed for the purpose of rescue people. The robot is able to avoid the obstacle and will stop to in front of the victim (human) to recognize the victim by using face recognition system. This robot used neural network algorithm as controller of the robot [3]

1.2 Problem Statement

In realizing the obstacle avoidance in mobile robot, the robot must poses the ability to detect obstacle. Sensory construction and capability in measuring accurate distance between mobile robot and the obstacles play a critical role in detection part since it will determine on how much the robot should react when facing various size of static objects.

The expected response from mobile robot after facing certain objects might be varies based on the situation that the mobile robot encounters. The response will be reflected by turning right, left or stop. To make an autonomous response of any value of sensor input and situation occurrences, a controller must be embedded. A desirable controller should be able to decide on course of action in tackling obstacles en-route.

1.3 Objectives

The objectives of this project are :

1. To asses a sensory construction of mobile robot in measuring accuracy of obstacle avoidance detection
2. To develop a fuzzy logic control algorithm for obstacle avoidance of mobile robot.
3. To evaluate the performance of the fuzzy logic controller that has been developed in mobile robot.

1.4 Scope

The scope of study which is needed for the completion of the project involved the following criteria:

i. The obstacles are placed within 250 cm radius from initial position of the robot and the data of three sonar sensor and speed of two DC motor are recorded while the robot is moving to avoid the obstacle.

ii. The algorithm of Fuzzy Logic Controller should be able to response to six different situations; when the obstacle at left side of the robot, when the obstacle at right side of the robot, when the obstacle is in front of the robot, when two obstacles that far apart in front of the robot, when two obstacles near of each other in front of the robot and when the robot need to do a u-turn instead of moving forward.

iii. The obstacle avoidance performance in the mobile robot will be tested using a Static or non-movement object such as boxes and the a results will be recorded for analysis.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter covers on the applications of mobile robot and the summary and discussion of section. On the applications of mobile robot, it covers briefly on obstacle avoidance in indoor environment. Next, the type of distance sensors, the type motors, and the type of controllers are covered in obstacle avoidance in indoor environment. The different types of distance sensor and motor used for obstacle avoidance and their construction are also include in the type of distance sensor and type of motor sections. Other than that, different types of controller are reviewed in order to control the mobile robot to avoid the obstacle.

2.2 The Applications of Mobile Robot

A mobile robot is an automatic machine that has an ability to move around in any given environment. There are many mobile robot used in various applications depending on the environment whether indoor environment or outdoor environment. The example of the usage of mobile robot in indoor environment is indoor security patrols, materials handling in the warehouse [4], and hospitals [1]. Meanwhile, the mobile robot also can be used in outdoor environment such as manufacturing industry, and hostile environment (nuclear power, submarine, space) [5]. In this research the application of mobile robot is focus on indoor environment for obstacle avoidance purpose.

2.2.1 Obstacle Avoidance in Indoor Environment

Obstacle avoidance technology is a very popular in the field of the ground mobile robot, and also one of the main embodiment of the intelligent robot. The mobile robot must has the ability to plan motion and to navigate autonomously avoiding any type of obstacles in indoor environment [6].

In this research, an efficient obstacle avoidance mobile robot was determined based on good components such as distance sensor, motor and controller after making a wise comparison.

2.2.1.1. Types of Distance Sensor

Many researchers have used sensor fusion to fuse data from various types of sensors, which improved the decision making process of routing the mobile robot. Each sensor has its own capability and accuracy, whereas integrating multiple sensors enhances the overall performance and detection of obstacles [7]. There are many types of distance sensor act as input for robot and used to avoid the obstacle such as ultrasonic sensor, IR sensor, laser sensor and a camera sensor.

The ultrasonic sensor is used in order to detect the accurate obstacle position. The true distance is measured by the time period from an ultrasonic pulse emission to the tip of the return pulse reception. In the obstacle position detection, the ultrasonic stereo sensor need to measures the distance and also the direction in order to detect the position of the obstacles since it is indicated the polar coordinate. [8]

For a robot with ability of obstacle avoidance in unknown environments, it usually requires a sensor such as a laser range finder to detect obstacles. However, even though laser range finder can obtain information around the environment, its cost is high and difficult to be widely utilized [9].

Infrared (IR) distance sensors is one of the hotspot and widely used as it is small size, easy to use, low cost and acceptable accuracy to used in robot that can avoid collision and obstacle [5][4][7]. But, the range of detection for IR sensor is limited and not suitable for a large scale of detection environment and to solve that problem, more IR sensor are needed to overcame the limited range of detection. One of the example of robot that used IR sensor to avoid avoidance is the two differential wheel robot (E-puck robot) that is used in this paper is equipped with eight infrared sensors (distance sensors) at surrounding of robot body to detect the obstacles [7].

The camera sensor that is used in avoiding obstacle robot is a range finder type of camera which allows obtaining distance in meters between the camera and the obstacle from the OpenGL context of the camera [7]. The camera sensor is suitable used with vision based controller to detected the obstacle.

Eventhough IR sensor is more popular than ultrasonic sensor and widely used in mobile robot, ultrasonic is more suitable used in this project for obstacle detection as ultrasonic sensor has wider range of detection compared to IR sensor. The wider range of detection give advantages in obstacle avoidance detection as the sensor can detection obstacle well even in complex environment.

2.2.1.2. Types of Controller

The capability of mobile robot to navigate autonomously has improved tremendously due to the improvement of various path planning and obstacle avoidance algorithms developed by recent researchers [10]. There are many types of controller that used to control the robot in avoiding obstacle such as Fuzzy Logic Controller, Artificial Neural Network and Vision-based Controller.

The fuzzy logic method has broadly used as one of effective means in unknown and complex industrial environments. In many research results, a fuzzy logic method has usually implemented for improving the efficiency of obstacle avoidance and path planning of mobile robot at unknown environments. Most fuzzy logic methods have a complex rule table for achieving different control objectives [11]. A fuzzy logic reactive navigation approach is described in which uses fuzzy rules to enable the mobile robot in reaching the destination by avoiding obstacles in its way [12]. A fuzzy logic reactive navigation approach is described in which using fuzzy rules to enable the mobile robot in reaching the destination by avoiding obstacle. The more the input variables are or the more detailed the fuzzy sets are , the faster the fuzzy rule's repository expands and the slower the fuzzy controller responds [13].

Artificial Neural Network (ANN) was introduced to make the mobile robot more intelligent. The nonlinear system had been used in the mobile robot is one of the concept of Multi Layer Perceptron. The robot is allowed to learn complex task such as avoiding obstacle in dynamic environment with layers of hidden unit from the network. The ANN will be used to train the performance of the robot that corresponds to the sensors on the robot. This approach focuses on the ability of the robot to move and avoid the obstacles in different of complexities of the environment. [11]

The vision-based obstacle avoidance technique is an appearance-based technique which classifies the input colour image from a monocular vision camera into defined classes such as ground, walls and obstacles. The robot is able to use the information from these image to localize itself in the environment. Vision sensors provide detailed information of the environment which can be used for navigation of mobile robots to follow the path and avoid the obstacles [14].

In this project, fuzzy logic controller is used as control system to control the movement of robot in avoiding obstacle. The fuzzy logic controller is one of the popular method in controlling mobile robot as this method is easy to developed based on input and output of the system. The vision-based obstacle avoidance technique is not suitable used in this project as the camera sensor is not available on the mobile robot. The artificial neural network (ANN) is suitable used in complex environment such as avoiding dynamic obstacle. However, this project only focus on detect and avoid static obstacles in simple environment, thus the ANN method is not suitable to used in this project.

2.3 Conclusion Based on Literature

As a conclusion from the literature review, the obstacle avoidance mobile robot for this project use the ultrasonic sensor to detect the obstacle, fuzzy logic controller used to control the movement of robot in order to avoid the obstacle detected and the DC motor as driving system for the robot to avoid the obstacle using pulse width modulation (PWM) method. In this project, the sensing response from the three units of ultrasonic sensor act as the input of the system. The fuzzy logic controller is act as control system for the robot. The DC motor at left and right side of the robot as the output of the system. With this configuration, the system should be able increase the accuracy to sense the obstacle and able to prevent the mobile robot from collide with the obstacles.

CHAPTER 3

METHODOLOGY

3.1 Obstacle Avoidance Mobile Robot System

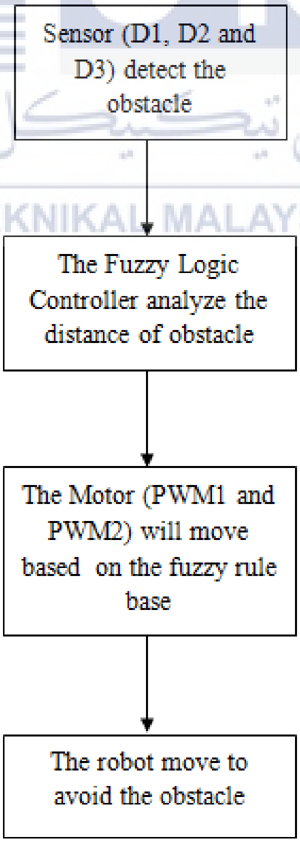


Figure 3.1 : The flowchart of obstacle avoidance mobile robot system

The mobile robot is attached with a sensory unit which is three unit of ultrasonic sensor. Each of the sensor is attached at different position at the body of mobile robot. The obstacle distance for right side of the mobile robot is represented as “D1”, “D2” for obstacle distance for front side and “D3” for obstacle distance for left side of the mobile robot. All the distance reading for “D1”, “D2” and “D3” from the sensor will act as the input to the fuzzy logic control and be analyze with respect to the membership functions and rule base of the fuzzy controller that have be designed. Meanwhile, the output for fuzzy logic control is the speed of both DC motor which is “PWM1” for the left motor and “PWM2” for the right motor. The PWM of both motors will control the mobile robot based on the fuzzy rule base that have been set. The mobile robot will be test its performance to avoid the obstacles in different type of situation.

3.2 Obstacle Avoidance Mobile Robot Kinematic

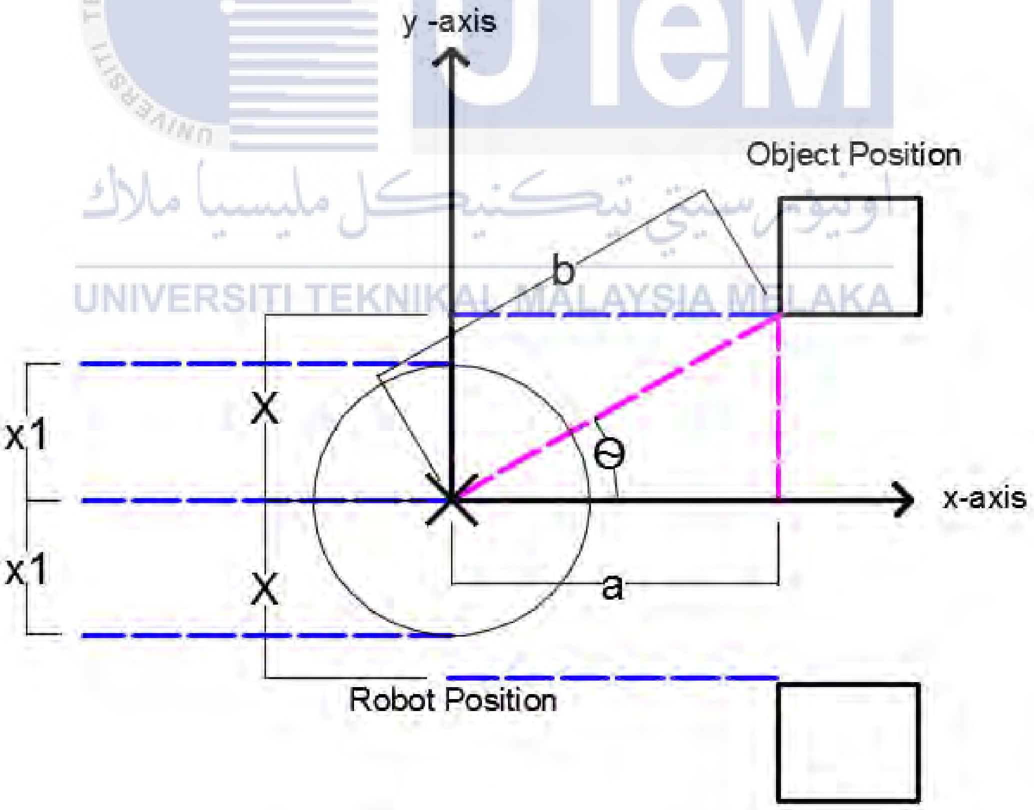


Figure 3.2 : Graphical plan based on robot and object positions

Figure 3.1 shows the kinematic model for obstacle avoidance mobile robot based on robot position and object detection position. The Pythagorean Theorem and Trigonometric method is used in order to determine the kinematic of the robot.

By referring to Figure 3.2 that using Pythagorean Theorem to solving kinematics :

$$\mathbf{X = [\ (b^2 - a^2)]^{1/2}} \qquad \mathbf{(3.1)}$$

Where;

X = the distance between the centre of the robot's body and the object detected (y-axis)

b = the displacement between the centre of the robot's body and the object detected

a = the distance between the centre of the robot's body and the object detected (x-axis)

By deriving the equation (3.1), the equations to determine angle between the centre of the robot's body and the object detected, ϴ is obtained. The ϴ will represent with the PWM of the robot.

Based on Figure 3.2, the robot placed between two objects and will move forward through the two objects. The robot will response based on;

If

x1 < X ; the robot can pass through the two objects,

If

x1 > X ; the robot can't pass through the two objects. The robot will change direction to the more spacious area.

3.2.1 Data Logging for Sensor

Ultrasonic sonar sensor was chosen in this project as it is suitable for the mobile robot to detected the obstacles. Ultrasonic sensor consist of one transmitter and one receiver. When the mobile robot approach the obstacle, a sound will emit by the speaker to the surroundings and the sound has been emitted hit the obstacles then the sound echo of the sound emitted will return back to and receive by the microphone. The sound emitted takes time to travel the wall distance. From the time travel the distance travel by emitted sound can be identified. [15]

The echo of the sound emitted is a distance between the ultrasonic sensors (D1, D2 and D3) and the obstacle detected. The formula below is used to calculate the distance of the obstacle.

In centimetres (cm);

Obstacle Distance = Pulse Distance / 58

In Inch;

Obstacle Distance = Pulse Distance / 148

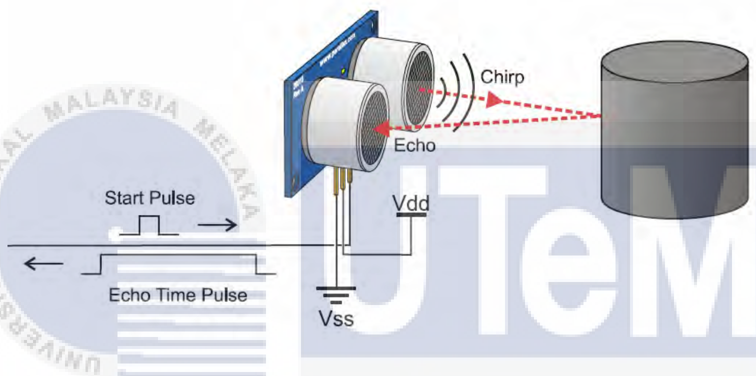


Figure 3.3 : The Working Principle of Ultrasonic Sensor [15]

3.2.2 Evaluate The Accuracy of The Sensor

When all the data of the each sensor have been obtained, the accuracy of each sensor will be identify by using the formula as be shown below:

$$Error = \left|\frac{\sum(\text{Expected Distance} - \text{Actual Distance})}{\sum \text{Expected Value}}\right| \quad (3.2)$$

$$Accuracy\ of\ sensor\ (\%) = (1 - error) \times 100\% \quad (3.3)$$

In this experiment, the accuracy of each sensor (D1, D2 and D3) can be measured by calculated the error using equation (3.2) for the reading of the sensor that obtained from the experiment. The actual distance is the average distance of the obstacle detected by sensor .Next, to find the percentage for accuracy of sensor, the equation (3.3) is used.

3.3 Control Block Design of Obstacle Avoidance Mobile Robot

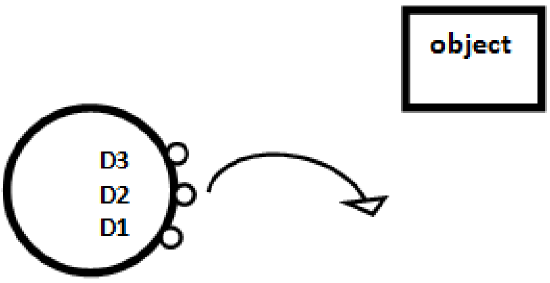
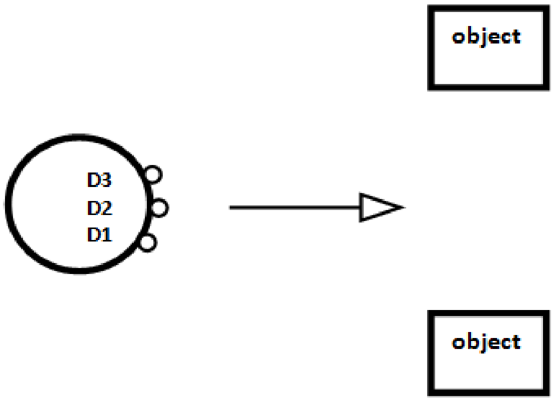
In this project, the ultrasonic sonar sensors (D1, D2 and D3) are used as the input to detect the obstacles distances for the mobile robot. The Fuzzy Logic Controller (FLC) will act as control system that control the movement of the mobile robot. Meanwhile, the output for this mobile robot is the speed of the motors. There are two outputs which is left motor (PWM1) and right motor (PWM2). Thus, this system consists of three inputs, one controller and two outputs. Figure 3.4 shows the control block diagram of obstacle avoidance mobile robot system.

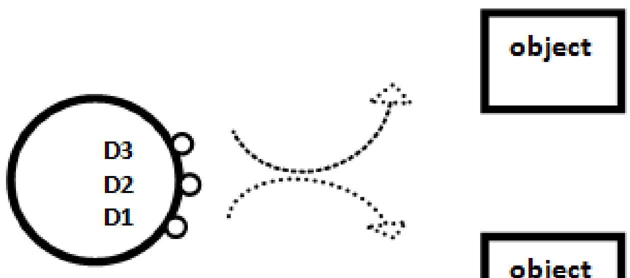
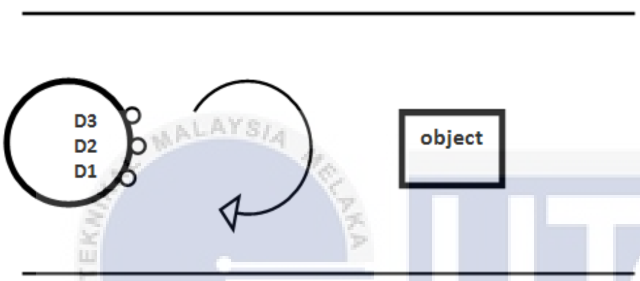
Figure 3.4 : Control Block Diagram of Obstacle Avoidance Mobile Robot System

3.4 Responses of Mobile Robot on Obstacle Detected

When the robot detect any object within the detection range of the ultrasonic sensor, the DC motor at left and right sides will rotate and the robot will avoid that object. There are different situations for the robot to avoid the object detected.

Table 3.1 : The Robot’s Responses in Different Situations

<table>
<tr><th>No.</th><th>Situations</th><th>Robot’s Response</th></tr>
<tr><td>1</td><td>
</td><td>❖ The object is at left side of the robot at the spacious area.
❖ The robot turns to the right side to avoid the object.</td></tr>
<tr><td>2</td><td>
</td><td>❖ The object is at right side of the robot at the spacious area
❖ The robot turns to the left side to avoid the object</td></tr>
<tr><td>3</td><td>
</td><td>❖ The object is in front of the robot at the spacious area
❖ The robot will turns whether to the left side or right side to avoid the object</td></tr>
<tr><td>4</td><td>
</td><td>❖ There are two objects in front of the robot that far apart at the spacious area
❖ The robot will adjust the direction and move pass through between two objects</td></tr>
</table>

5		❖ There are two objects in front of the robot that near of each other at the spacious area ❖ The robot is not moving forward but change the direction whether to the left or right side that has more spaces.
6		❖ The object is in front of the robot at small area (between two walls). ❖ The robot is not moving forward but do a u-turn to change the direction

3.5 Fuzzy Logic Controller

The fuzzy logic controller (FLC) provide a method of transforming control based on expert knowledge into an automatic control strategy. Fuzzy control is convenient for handling a complex analysis using conventional quantitative techniques or when the available sources of information provide qualitative, approximate, or uncertain data. Figure 3.5 shows the Fuzzy Logic Controller is consists of three steps:

1. Fuzzification: Change the control inputs into an easily used information that the inference mechanism can activate and apply rules.
2. Rule base: A set of IF-Then rules that consists of fuzzy logic quantification from the expert's linguistic description that is controllable.
3. Defuzzification: This converts the conclusions of the interface mechanism into actual inputs for the process.

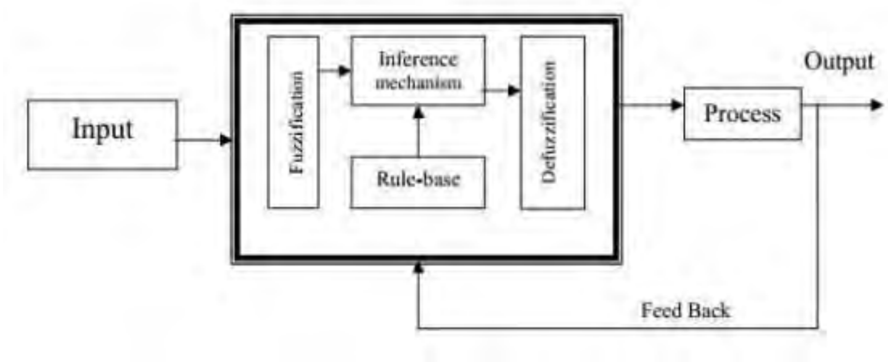


Figure 3.5 : Fuzzy Logic Controller

3.5.1 Parameter Involved in Obstacle Avoidance Mobile Robot

There are two parameter involved in obstacle avoidance mobile robot which are input parameter and output parameter. For input parameter is referring to distance of the obstacle detected by the ultrasonic sensor (D1, D2 and D3). Meanwhile, for the output parameter is refers to the speed of DC motor (PWM1 and PWM2) and the direction of robot to avoid the obstacle.

Table 3.2 : Parameter involved in obstacle avoidance mobile robot

Input parameter	Output parameter
1. **Ultrasonic Sensor** – Data occur when the obstacle is detected. The distance of the obstacle will be analyzed in fuzzy logic controller.	1. **DC motor** – The speed of the motor is determined based on the pulse with module of motor (PWM). The direction of the robot is determined after being analyzed by fuzzy logic controller.

3.5.2 Fuzzy Logic approach in Obstacle Avoidance Mobile Robot

The fuzzy logic is one of method that used to control the movement of the robot in order to avoid the obstacle. In this research, ultrasonic sensor is used as input while DC motor used as the output. The fuzzy logic controller will analyzed the data of distance of the obstacle detected by sensor based on range of distance set in fuzzy membership function. The data distance obtained from the sensor will classify whether the distance is in range for Near, Medium or Far parameters. With this analysis given to the fuzzy controller, the output of controller is used to rectify the speed of DC motor whether Stop, Very Slow, Slow or Fast. The direction of robot's movement is determined by fuzzy rule based in order to avoid the obstacle.

3.5.2.1 Block Diagram of Fuzzy Based Obstacle Avoidance

Ultrasonic sensor is used to detect and measure the distance value between the mobile robot and surrounding objects (obstacles). The distance that measured then will be used to build the fuzzy membership function by using MATLAB software. This acquired data produced from the sensors shows that there are an obstacle near the robot, so that it must change its direction and motor speed in order to avoid that obstacle. Figure 3.6 below show the block diagram of fuzzy based obstacle avoidance system.

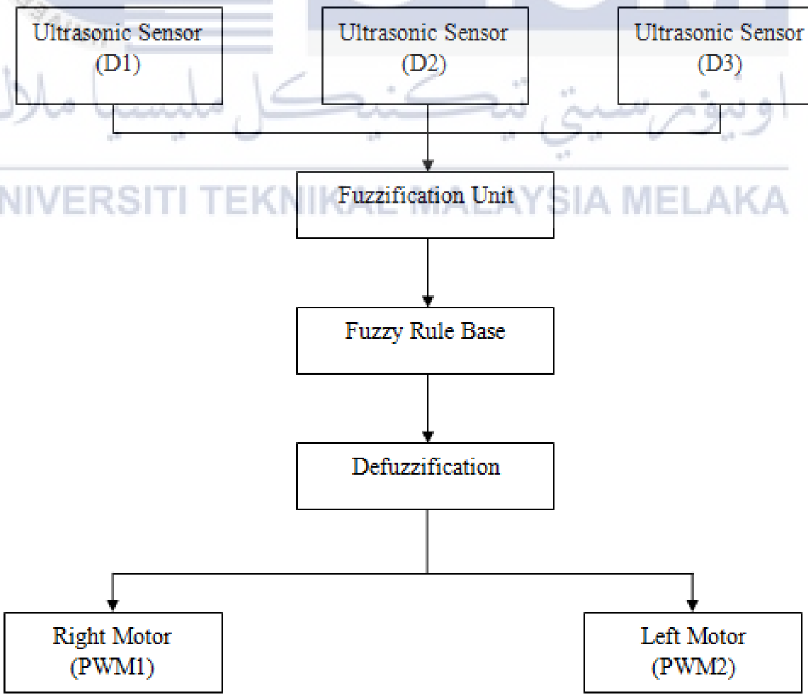


Figure 3.6 : Block Diagram of Fuzzy Based Obstacle Avoidance System

3.5.2.2 Input and Output of the Fuzzy Logic Controller

Table 3.3 shows the input and output variable that will be implement to the FIS editor in Fuzzy Logic Controller as parameter variable. These variable is define to developed a system from obstacle avoiding ability and implemented it in mobile robot. By using fuzzy logic controller, there are two inputs and two outputs as variable in controlling the system.

Table 3.3 : Input and Output variable of fuzzy logic control system

Input Variable	Output Variable
1. The detection of the obstacle within 250 cm radius of the range of detection of the sensor. 2. The different distances from obstacle detected by three sensors	1. The speed of the DC motor with the body of robot moves to avoid the obstacle 2. The direction of the robot define as the suitable speed when avoiding the obstacle

As for the robot moves to avoid the obstacle, it is depend on the speed of both DC motor at right and left side of the robot. If the speed at left motor is slow and the right motor is fast, the robot will turn left. Meanwhile, when the speed at left motor is fast while at right motor is slow, the robot will turn right. However, if both motor have same speed, it is identically the robot moving in straight line. The direction of the motor is determined by fuzzy rule base that have been set.

3.5.3 Fuzzification Unit

The mobile robot consists of three ultrasonic sensors used for measuring the distances which is right obstacle distance (D1), front obstacle distance (D2) and left obstacle distance (D3). The fuzzy logic controller will control the output which are the speed of DC motors that control by pulse width modulation (PWM) at right and left DC motors.

3.5.3.1 Membership function

In this fuzzy tool, it provides a function for analyzing, designing and simulating a system. Other specification is the input and output in term of membership function. The input can be set to some various inputs if many inputs are involved.

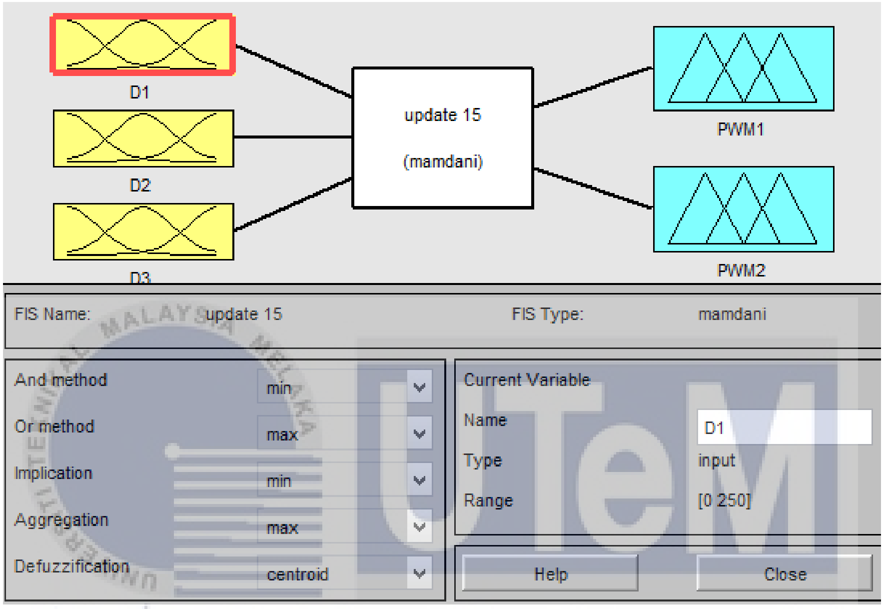


Figure 3.7 : General of Membership Function used for fuzzy controller

From the Figure 3.7 show the general of membership function of this project. This membership function can be defined as a mapping of point that associates each member linguistic variables with its grades of membership in the set of linguistic values. There is few types of membership functions that most commonly used in practice are triangles, trapezoids, bell curves, Gaussian and sigmoidal to assign membership values or functions to fuzzy variable.

In the obstacle avoidance mobile robot system is considered three derivative input and two derivative output. These three inputs is assign with distance of the obstacle detected by the ultrasonic sensor (D1, D2 and D3) while speed of left DC motor (PWM1) and speed of right motor (PWM2) as the output of the fuzzy logic membership functions. In this project, most suitable type of fuzzy is by using Ebrahim-Mamdani because it is for fuzzy algorithms for complex systems and decision processes.

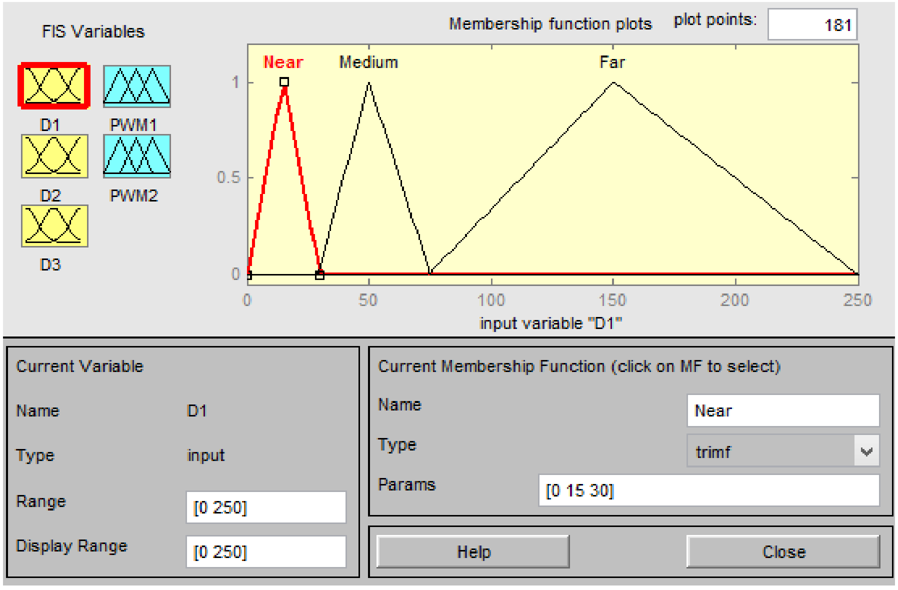


Figure 3.8 : Membership function for distance of the obstacle detected by sensor (D1) normalized input in the range of 0 to 250

In Figure 3.8 indicate a three membership function with three different conditions. This distance of obstacle detected by sensor (D1) variable is assign as first input which consists of Near, Medium and Far with the difference range set up of parameter for Near (0 15 30), medium (29 50 75) and Far (74 150 250) respectively. This parameter with same the same range also applied at D2 as second input and D3 as third input.

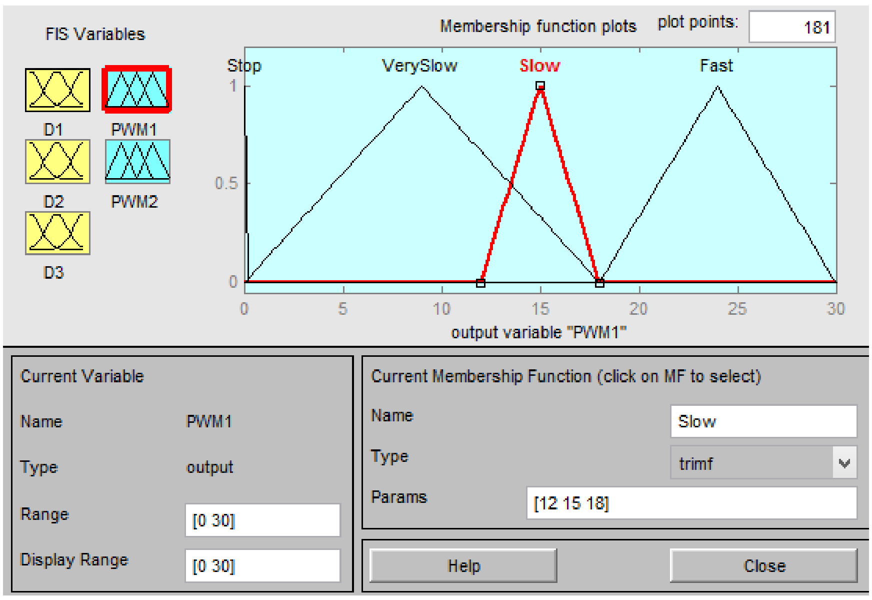


Figure 3.9 : Membership function for PWM1 normalized output in range of 0 to 30

The first output of the three membership functions with four different conditions. This speed of DC motor variable is consists Stop, Very Slow, Slow and Fast with the different range set up parameter for Stop (0 0 0), Very Slow (0 9 18), Slow (12, 15, 18) and Fast (18 24 30) respectively. This same range set up parameter also applied at the second output, Speed2.

3.5.3.2 Rule based

Table 3.4 : The fuzzy set values of the input and output variables with range based on data collection

Input/output	Linguistic Variables	Range
D1 D2 D3	➢ Near ➢ Medium ➢ Far	0 cm < D1,D2,D3 < 250 cm
Left Motor (PWM1) Right Motor (PWM2)	➢ Stop ➢ Very Slow ➢ Slow ➢ Fast	0 < PWM1 & PWM2 < 30

3.5.3.3 Fuzzy rules

In this project, the 3x3x3 fuzzy set have been developed. The fuzzy controller attempts to control the speed of DC motors and the direction of robot to avoid the obstacle based on the distance of obstacle detected by sensor. Table 3.5 shows the condition according to the detection of obstacle with three sensor (D1, D2 and D3) that will influenced the speed of both DC motor (PWM1 and PWM2). Table 3.6 shows the fuzzy rules condition to find the output using IF-THEN method. Based on this fuzzy rules condition, the movement and direction of the robot can be control to avoid the obstacle detected by sensor.

Table 3.5: The Conditions according to the Detection of Obstacles

No.	Right-Obstacle Distance (D1)	Front-Obstacle Distance (D2)	Left-Obstacle Distance (D3)
1	Near	Near	Near
2	Near	Near	Medium
3	Near	Near	Far
4	Near	Medium	Near
5	Near	Medium	Medium
6	Near	Medium	Far
7	Near	Far	Near
8	Near	Far	Medium
9	Near	Far	Far
10	Medium	Near	Near
11	Medium	Near	Medium
12	Medium	Near	Far
13	Medium	Medium	Near
14	Medium	Medium	Medium
15	Medium	Medium	Far
16	Medium	Far	Near
17	Medium	Far	Medium
18	Medium	Far	Far
19	Far	Near	Near
20	Far	Near	Medium
21	Far	Near	Far
22	Far	Medium	Near
23	Far	Medium	Medium
24	Far	Medium	Far

25	Far	Far	Near
26	Far	Far	Medium
27	Far	Far	Far

Table 3.6 : Fuzzy rules condition to find the output

Number	Fuzzy Rules Condition (IF-THEN Rules)
1	If (*D1* Near, *D2* Near & *D3* Near) then (*PWM1* Stop & *PWM2* Stop)
2	If (*D1* Near, *D2* Near & *D3* Medium) then (*PWM1* Very Slow & *PWM2* Fast)
3	If (*D1* Near, *D2* Near & *D3* Far) then (*PWM1* Very Slow & *PWM2* Fast)
4	If (*D1* Near, *D2* Medium & *D3* Near) then (*PWM1* Stop & *PWM2* Stop)
5	If (*D1* Near, *D2* Medium & *D3* Medium) then (*PWM1* Very Slow & *PWM2* Fast)
6	If (*D1* Near, *D2*Medium & *D3* Far) then (*PWM1* Very Slow & *PWM2* Fast)
7	If (*D1* Near, *D2* Far & *D3* Near) then (*PWM1* Stop & *PWM2* Stop)
8	If (*D1* Near, *D2* Far & *D3* Medium) then (*PWM1* Very Slow & *PWM2* Fast)
9	If (*D1* Near, *D2* Medium & *D3* Far) then (*PWM1* Very Slow & *PWM2* Fast)
10	If (*D1* Medium, *D2* Near & *D3* Near) then (*PWM1* Stop & *PWM2* Stop)
11	If (*D1* Medium, *D2* Near & *D3* Medium) then (*PWM1* Slow & *PWM2* Slow)
12	If (*D1* Medium, *D2* Near & *D3* Far) then (*PWM1* Very Slow & *PWM2* Fast)
13	If (*D1* Medium, *D2* Medium & *D3* Near) then (*PWM1* Fast & *PWM2* Very Slow)
14	If (*D1* Medium, *D2* Medium & *D3* Medium) then (*PWM1* Slow & *PWM2* Slow)
15	If (*D1* Medium, *D2* Medium & *D3* Far) then (*PWM1* Very Slow & *PWM2* Fast)
16	If (*D1* Medium, *D2* Far & *D3* Near) then (*PWM1* Fast & *PWM2* Very Slow)

17	If (*D1* Medium, *D2* Far & *D3* Medium) then (*PWM1* Slow & *PWM2* Slow)
18	If (*D1* Medium, *D2* Far & *D3* Far) then (*PWM1* Fast & *PWM2* Very Slow)
19	If (*D1* Far, *D2* Near & *D3* Near) then (*PWM1* Fast & *PWM2* Very Slow)
20	If (*D1* Far, *D2* Near & *D3* Medium) then (*PWM1* Fast & *PWM2* Very Slow)
21	If (*D1* Far, *D2* Near & *D3* Far) then (*PWM1* Fast & *PWM2* Very Slow)
22	If (*D1* Far, *D2* Medium & *D3* Near) then (*PWM1* Fast & *PWM2* Very Slow)
23	If (*D1* Far, *D2* Medium & *D3* Medium) then (*PWM1* Fast & *PWM2* Very Slow)
24	If (*D1* Far, *D2* Medium & *D3* Far) then (*PWM1* Slow & *PWM2* Slow)
25	If (*D1* Far, *D2* Far & *D3* Near) then (*PWM1* Fast & *PWM2* Very Slow)
26	If (*D1* Far, *D2* Far & *D3* Medium) then (*PWM1* Fast & *PWM2* Very Slow)
27	If (*D1* Far, *D2* Far & D3 *Far*) then (*PWM1* Fast & *PWM2* Fast)

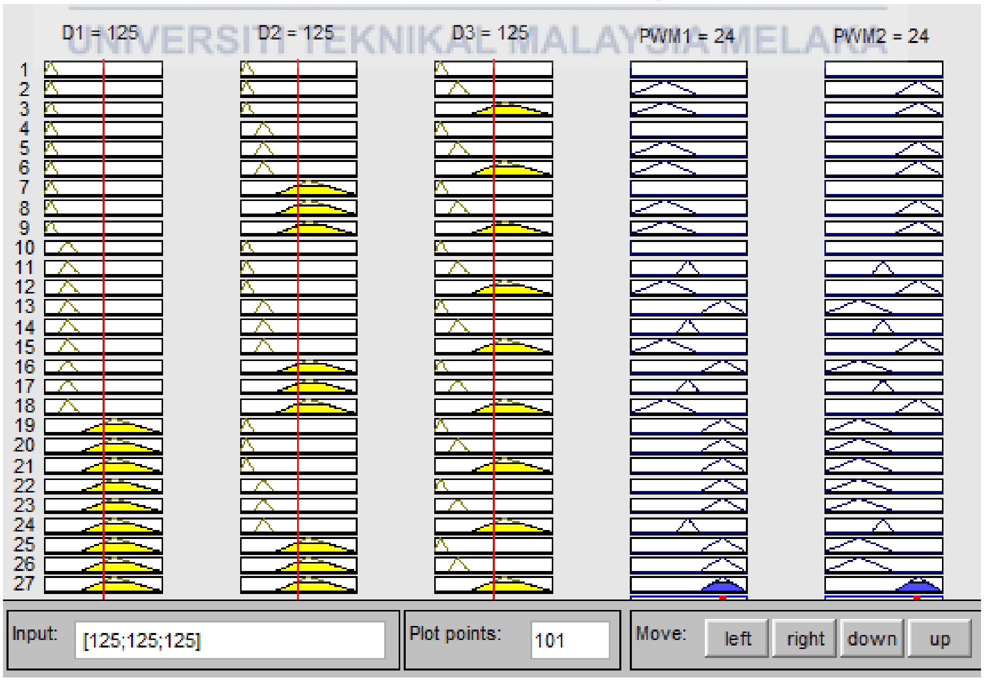


Figure 3.10 : Rules viewer for the membership functions

For the rule base of the fuzzy logic controller, twenty-seven rules have been set for the input and the output of the fuzzy logic controller for the obstacle avoidance system. By using this set of rules as show below, it is able to fulfil the requirement of the performing to controlling the movement of the mobile robot for navigate the mobile robot without collide with the obstacles. This rules base was sufficient to encompass all realistic combinations of inputs and outputs of the fuzzy control motion. Figure 3.10 shows the rules viewer for membership functions that consists of a set of IF-THEN rules that map fuzzy range variables into object distance of all three ultrasonic sensor; D1, D2 and D3 while PWM1 and PWM2 for speed for both motors.

3.5.3.4 Defuzzification

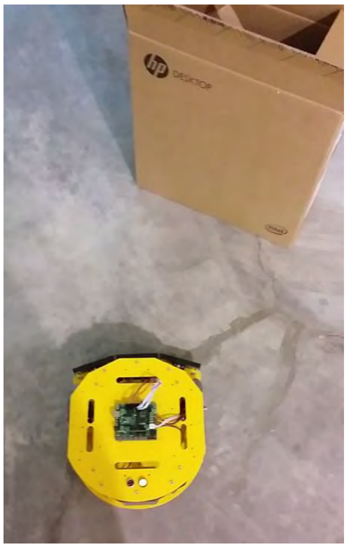
This defuzzification is defined as a mapping of fuzzy sets into a crisp point. It refers to all the actions that activated are combined and converted into a single non-fuzzy output signal which control signal of the system. So, the outputs levels are depending on rules the system have and positions is depending on non-linearity's existing on systems [20]. There are several methods used in the defuzzification which are Max Method, Weight Average Method, Mean-Max Membership Method and lastly Center Of Gravity Method.

As for the project, a Centroid or Center Of Gravity method is applied because the highly prevalent and physically noticeable of all the defuzzification method. Among of those methods, mostly more preferred technique is Center of Gravity and the Mean-Max method because more specified.

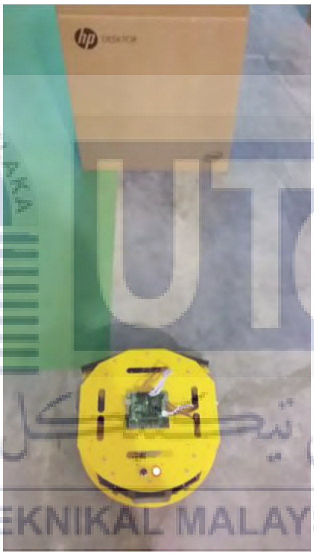
3.6 Experimental Setup of Obstacle Avoidance for Static Object

A few experiments had been carried out to test the performance of the robot in order to avoid the obstacle in six different situations which are when the obstacle at left side of the robot, when the obstacle at right side of the robot, when the obstacle is in front of the robot, when two obstacles that far apart in front of the robot, when two obstacles near of each other in front of the robot and when the robot need to do a u-turn instead of moving forward.

(a)



(b)



(c)

Figure 3.11 : The Object’s Position. (a) Left-side of the robot (b) Right-side of the robot (c) In front of the robot

Figure 3.11 (a) shows that the robot is test according case 1 : when the obstacle at left side of the robot. Figure 3.11 (b) shows case 2 : when the obstacle at right side of the robot and Figure 3.11 (c) shows case 3: when the obstacle is in front of the robot.

(a) (b)

Figure 3.12 : The obstacles placed in front of the robot. (a) The obstacles far apart of each other (b) The obstacles near of each other

Then, the robot is tested using two obstacles that placed in front of the robot with different distance from each other such. The boxes act as the obstacles. Figure 3.12 (a) shows the case 4 : when two obstacles that far apart in front of the robot. and Figure 3.12 (b) shows the case 5 : when two obstacles near of each other in front of the robot

Figure 3.13 : Situation that required The Robot do a U-Turn

Next, the robot is test for case 6 where the robot need to do a u-turn instead of moving forward as shown in Figure 3.13 above. The boxes act as obstacles.

3.7 Experimental Setup for Fuzzy Logic Controller

This part cover about a few experiment which have been conducted to analyze the performance of the fuzzy logic controller on mobile robot. This experiments have been done in different situations with different positions of obstacle. The input (D1, D2 and D3) at output (PWM1 and PWM2) will identify the accuracy of fuzzy logic controller performance for the mobile robot in order to avoid the obstacle. Figure 3.11 shows that the range of detection for the ultrasonic sensor that act as input (D1, D2 and D3) which is 4 cm as minimum distance of obstacle detection and 250 cm is the maximum distance for obstacle detection. The range of detection for input (D1, D2 and D3) is represented the parameter of linguistic variables for “Near” , “Medium” and “Far” when the robot detected the obstacle. Then the output (Motor Left and Motor Right) will analyze whether “Slow”, ”Very Slow”, “Fast” and “Stop” using fuzzy logic controller.

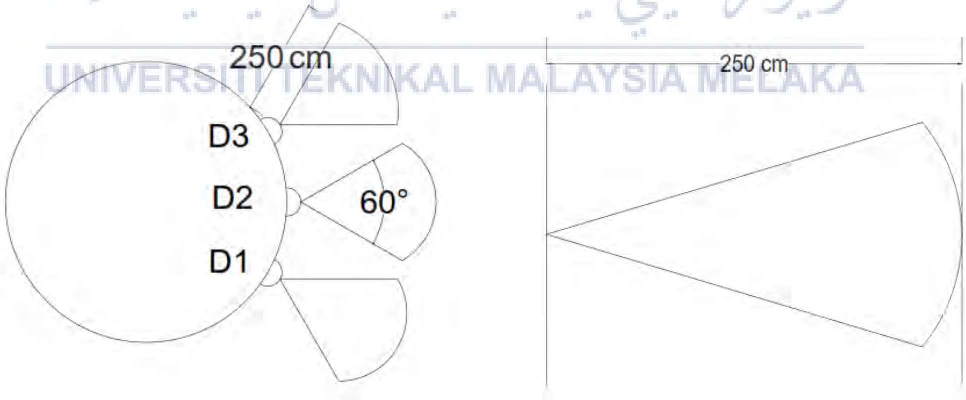


Figure 3.14 : The detection range for input D1, D2 and D3

Based on Figure 3.14 above, the robot is tested with different types of obstacle’s position that placed at right-side, left-side and in front of the robot. In Figure 3.12 and 3.13, the robot is tested using two and three obstacles that placed in the same row but different types of distance apart of each other. When the switch is turn on, the robot started moving forward until it detected the obstacles. The movement of robot to avoid the obstacle is based on fuzzy rules that had been set. The data from input (D1, D2 and D3) and the output (PWM1 and PWM2) of the fuzzy logic controller and the movement of the robot had been recorded while the robot avoiding the obstacle.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In the previous chapter methodology described the project method that has been discussed in detail use for this project. In this chapter is presenting the result from data analysis obtained from this project. The first part is discussion about accuracy of ultrasonic sensors (D1,D2 and D3) for distance of obstacle detected with different distance from 10 cm to 60 cm. The second part is discussion about accuracy of sensor for range of detection for obstacle detected. This test is to test the accuracy for range of detection for the sensor to detected the obstacle and conducted within 70º angle which is started from -35º until 35º. The third until ninth part are about analysis of experimental result of robot's response for different situations. Lastly, the tenth part, a summary of the result to discuss about rate of success involving all situations.

4.2 The Accuracy of The Ultrasonic Sensor for Distance of Obstacle Detected

Figure 4.1 : The example Accuracy Test for Ultrasonic Sensor

From the experiment, each of the ultrasonic sensor is tested with different distance from 10 cm until 60 cm from the object detected to the mobile robot. The experiment is conducted in three type of cases which is the object detected is in front of the robot, the object detected at the right side of robot and the object detected at the left side of robot. This experiment is conducted three times for each distances. Figure 4.1 shows the example accuracy test for ultrasonic sensor (D1,D2 and D3). The average reading collected from each sensor can be referred in Appendix A.

4.2.1 When The Object at Left-Side of Ultrasonic Sensors

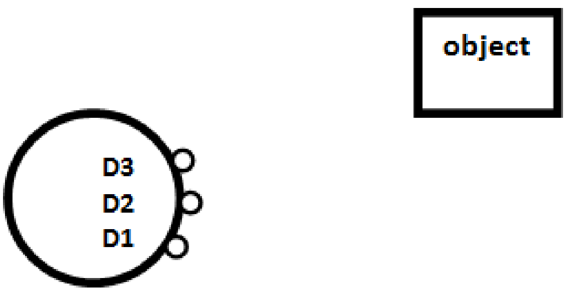


Figure 4.2 : The Object at The Left-Side of Ultrasonic Sensors

For case 1, the experiment is conducted with the object at the left side of ultrasonic sensors as shown in Figure 4.2. The object is placed at angle of 60º degree at the left side. The data from the ultrasonic sensors (D1, D2 and D3) were recorded.

From the Table 4.1, the reading for ultrasonic sensor 1 (D1) were not recorded as the object was out of the range of detection. The least accuracy percentage of ultrasonic sensor 2 (D2) is 73.33% at 10 cm distance and at 60 cm, D2 has the most percentage of accuracy which is 96.67%. Meanwhile the ultrasonic sensor 3 (D3) has 100% accuracy for all distance (10 cm – 60 cm)

Table 4.1: The Accuracy of Ultrasonic Sensors When Object at Left Side of The Robot

Expected Distance (cm)	**D2 (%)**	**D3 (%)**
10	73.33	100
20	83.35	100
30	90	100
40	94.18	100
50	94.42	100
60	96.67	100

4.2.2 When The Object at Right-Side of Ultrasonic Sensors

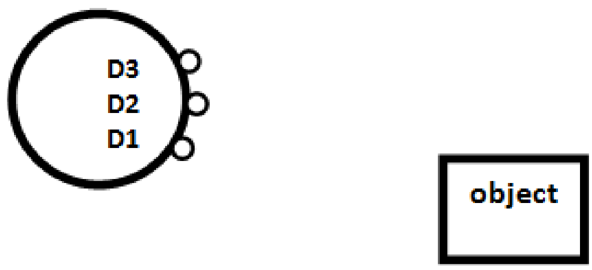


Figure 4.3 : The Object at The Right-Side of Ultrasonic Sensors

For case 2, the experiment is conducted with the object at the right side of ultrasonic sensors as shown in Figure 4.3. The object is placed at angle of 60º degree at the right side. The data from the ultrasonic sensors (D1, D2 and D3) were recorded.

From the Table 4.2, the reading for ultrasonic sensor 3 (D3) were not recorded as the object was out of the range of detection. The least percentage of accuracy for ultrasonic sensor 1 (D1) is 90% at 10 cm distance while at distance 20 cm until 50 cm, the percentage of accuracy for D1 are 100%. Meanwhile, for ultrasonic sensor 2 (D2), the least percentage of accuracy is 90 % at 10 cm and 20 cm distances while 96.67% is the most percentage of accuracy for D2 at 60 cm distance.

Table 4.2: The Accuracy of Ultrasonic Sensors When Object at Right Side of The Robot

Expected Distance (cm)	D1 (%)	D2 (%)
10	90	90
20	100	90
30	100	93.33
40	100	95
50	100	96
60	98.33	96.67

4.2.3 When The Object is In Front of Ultrasonic Sensors

Figure 4.4 : The Object at is in Front of Ultrasonic Sensors

For case 3, the experiment is conducted with the object is in front of the ultrasonic sensors as shown in Figure 4.4. The object is placed at angle of 0º degree. The data from the ultrasonic sensors (D1, D2 and D3) were recorded.

From the Table 4.3, all ultrasonic sensor gave out the reading for 10 cm until 60 cm distances. For ultrasonic sensor 1 (D1), the least percentage of accuracy is 60% at 10 cm distance while 92.78% is the most percentage of accuracy at 60 cm distance. The least accurate for ultrasonic sensor 2 (D2) is at 10 cm distance which is 90% while at 60 cm distance, D2 gave out the reading of obstacle distance the most accurate which is 98.33%. Next, for ultrasonic sensor 3 (D3), 70% is the least accuracy of sensor while detected the obstacle at 10 cm while 93.33% is the most percentage of accuracy while detected the obstacle at 60 cm.

Table 4.3: The Accuracy of Ultrasonic Sensors When Object In Front of The Robot

Expected Distance (cm)	D1 (%)	D2 (%)	D3 (%)
10	60	90	70
20	80	95	80
30	83.33	96.67	86.67
40	87.5	97.5	90
50	92	98	92
60	92.78	98.33	93.33

4.2.4 The Summary of Accuracy Test for Ultrasonic Sensor with Different Position of Obstacle

Table 4.4 shows the summary of the accuracy test for ultrasonic sensor (D1,D2 and D3) involving different position of obstacle. When the obstacle at left side of the robot, only D2 and D3 gave out the reading of 88.66% and 100% respectively. Meanwhile, when obstacle at right side of robot, only D1 and D2 gave out the reading. The reading for D1 is 98.06% and for D2 is 93.5%. All sensor gave out the reading when the obstacle is in front of the robot. The reading for D1, D2 and D3 are 82.6%, 95.92% and 85.33% respectively.

Table 4.4 : The summary of accuracy of Ultrasonic Sensors with different position of obstacle

The Position of The Obstacle Towards the Mobile Robot	**Accuracy Distance D1 (%)**	**Accuracy Distance D2 (%)**	**Accuracy Distance D3 (%)**
Left Side	-	88.66	100
Right Side	98.06	93.5	-
Front	82.6	95.92	85.33

4.3 The Accuracy of The Ultrasonic Sensor for Range of Detection for Obstacle Detected

Figure 4.5 : The Object is Place at 50 cm With Different Angles

The purpose of this experiment is to test the accuracy of the ultrasonic sensor for range of detection towards the object detected within the angle -35 º until 35 º. This experiment is conducted using one object that placed in front of ultrasonic sensor at 50 cm distance to test whether the sensor can detected the object within the range of detection or not as shown in Figure 4.5. The ultrasonic sensor 2 (D2) was used in this experiment, thus the reading for D2 were collected. The average reading collected from sensor can be referred in Appendix A.

From the Table 4.5, at angle -35º and 35º, the reading for ultrasonic sensor 1 (D2) were not recorded as the object was out of the range of detection. The least percentage of accuracy is 96% at angle 10 º while at angle 20 º the accuracy of sensor is 100%. From this test, we can conclude that the range of detection for ultrasonic sensor is 60 º which is -30 º until 30 ºangles.

Table 4.5: The Accuracy of Ultrasonic Sensors at 50 cm With Different Angles

Angle of Range Detection(º)	**Accuracy of Sensor (%)**
-35	-
-30	98
-20	98
-10	98
0	98
10	96
20	100
30	98
35	-

4.4 Experimental Result Case 1 (When the obstacle at left side of the robot)

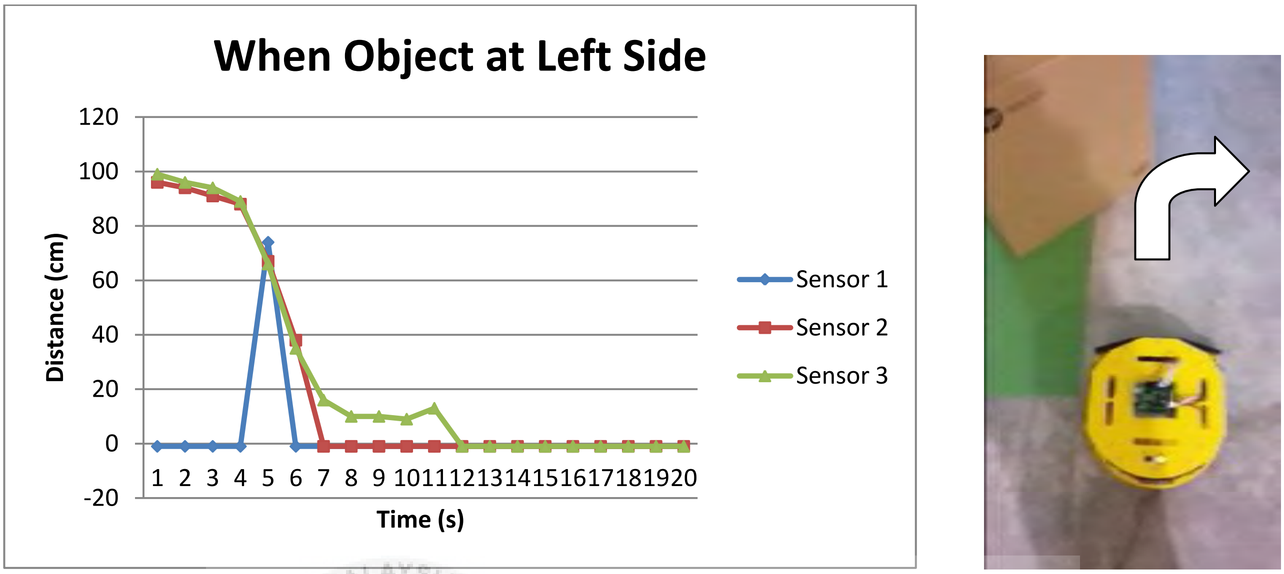


Figure 4.6: The distance graph for Case 1

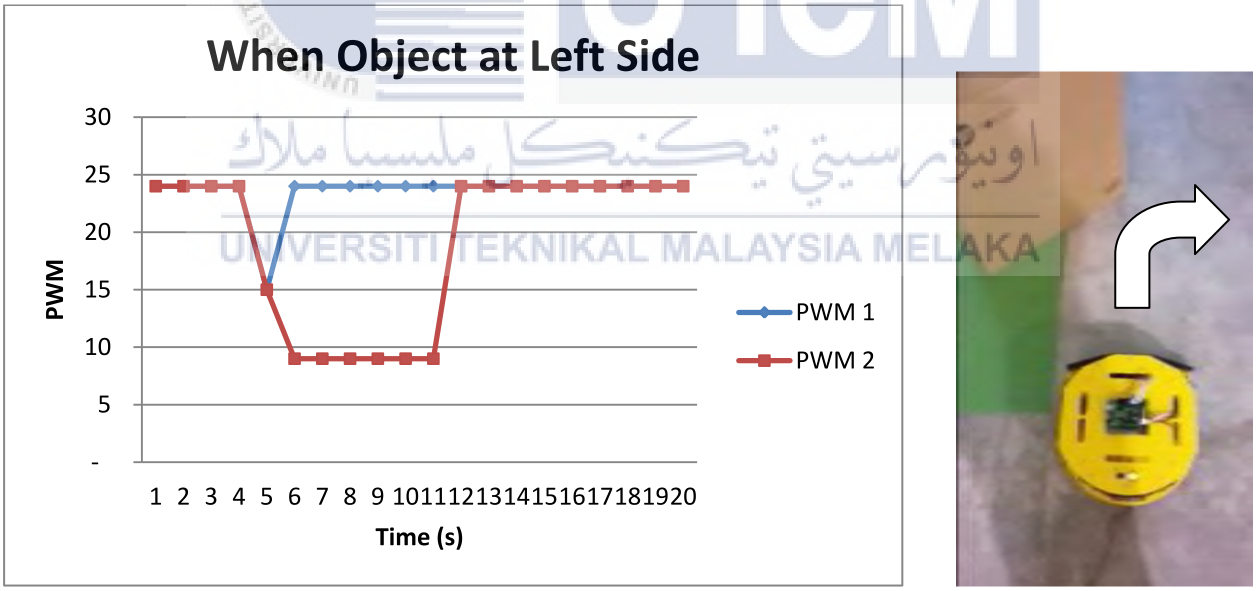


Figure 4.7: The PWM graph for Case 1

From the graph above, along the time interval from 1s to 20s show that the reading of ultrasonic sensor 1 (D1) is -1 which mean the obstacle detection are out of range. However at 5s, the reading of D1 shows is 74 cm distance. This happen due to change of condition from “Far” to “Medium” distance of input linguistic variable, the D1 is able to detect the obstacle for 1s as the distance of obstacle is in “Medium” condition range of distance. The reading go back to -1 as the robot is turning right. Meanwhile, at time interval 1s to 6s, the ultrasonic sensor 2 (D2) start to decreased from 96 cm to 38 cm. After 6s, D2 gave out the reading out of range. For ultrasonic sensor 3 (D3) also decreased from 99 cm until 13 cm at time interval 1s to 11s. Meanwhile, the PWM1 and PWM2 from 1s to 4s have just maintain at 24.

At time interval 6s until 11s, the PWM1 is 24 and PWM2 is 9. At this moment, the robot is turning right since the input of the fuzzy logic controller have meet the fuzzy rules no. 23 and 25 which are for rules no. 23, If (*D1* Far, *D2* Medium & *D3* Medium) then (*PWM1* Fast & *PWM2* Very Slow) and rule no.25, , If (*D1* Far, *D2* Far & *D3* Near) then (*PWM1* Fast & *PWM2* Very Slow). The robot is turning right until all three sensors gave out the reading out of range which mean the robot is already turning right toward the spacious area. In this experiment, the mobile robot success to turn right when the obstacle the at left side.

4.5 Experimental Result Case 2 (When the obstacle at right side of the robot)

Figure 4.8: The distance graph for Case 2

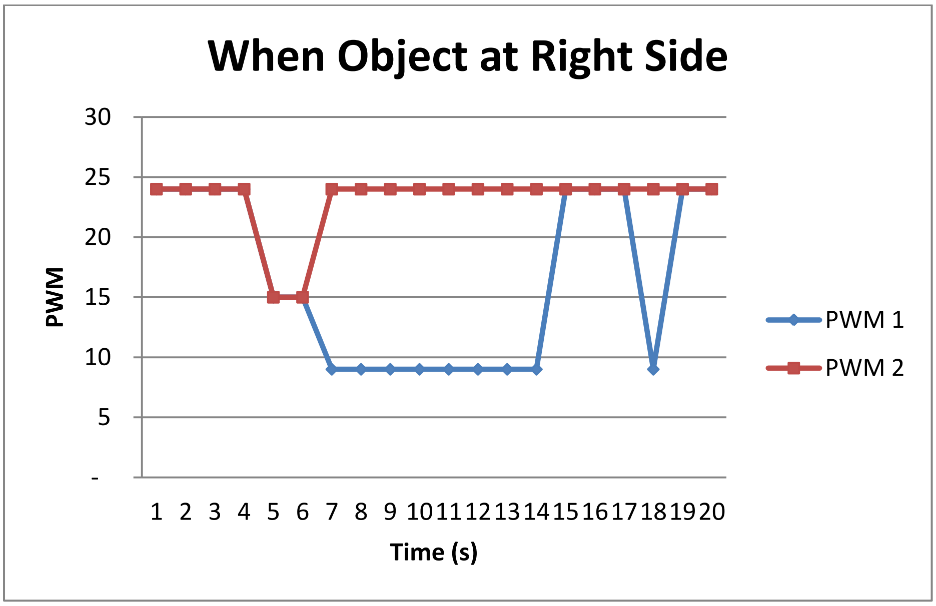


Figure 4.9: The PWM graph for Case 2

In this experiment, from the graph above along the time interval from 1s to 7s show that the reading of ultrasonic sensor 1 (D1) have decreased from 87 cm to 20 cm, ultrasonic sensor 2 (D2) also decreased from 84 cm to 17 cm at 1s to 7s time interval and ultrasonic sensor 3 (D3) show reading of -1 at time interval 1s until 20s except at 5s and 6s, the reading show are 66 cm and 41 cm respectively. This is because, due to change of condition from "Far" to "Medium" distance of input linguistic variable, the D3 is able to gave out the reading of the distance at 5s to 6s and then back to reading out of range as the robot is turning left. The PWM1 and PWM2 maintain the speed of 24 from 1s to 4s.

At time interval 7s until 14s, the PWM1 is 9 and PWM2 is 24. At this moment, the robot start to turn left since the input of the fuzzy logic controller have meet the fuzzy rules no. 3, If (*D1* Near, *D2* Near & *D3* Far) then (*PWM1* Very Slow & *PWM2* Fast) and rule no.15, If (*D1* Medium, *D2* Medium & *D3* Far) then (*PWM1* Very Slow & *PWM2* Fast). The robot is turning left until all three sensors gave out the reading out of range at time interval 19s to 20s which mean the robot is successfully turn left. However, at reading 18s, the PWM1 is 9 and PWM2 is 24 as the D1 and D2 detect the other obstacle after turning left. This reading may consider as error from this experiment.

4.6 Experimental Result Case 3 (When the obstacle is in front of the robot)

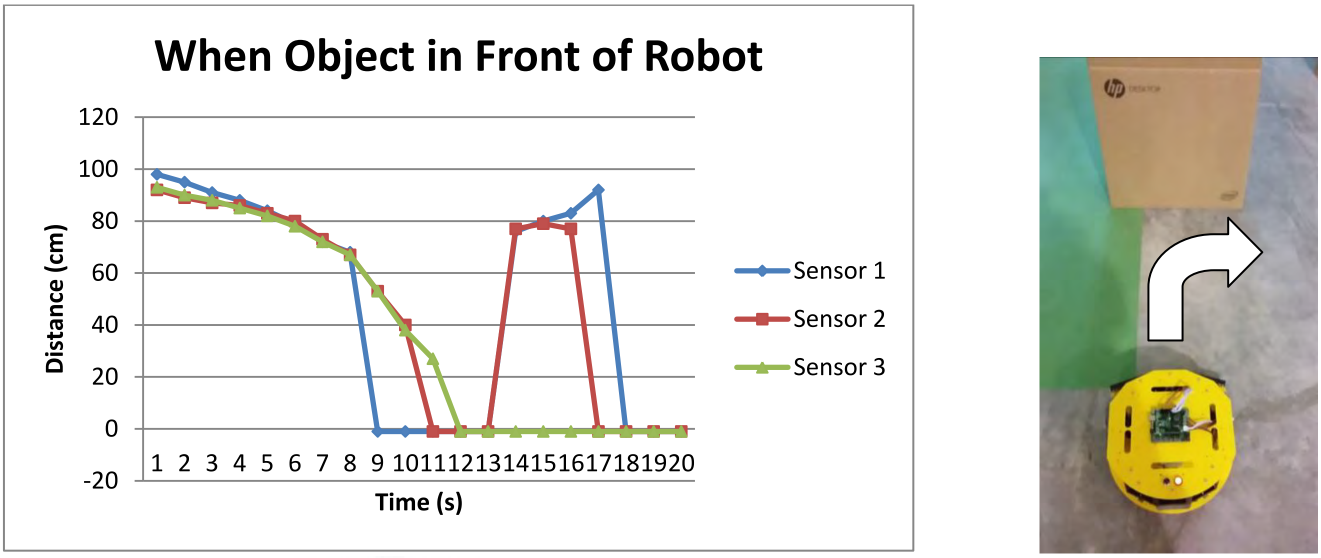


Figure 4.10: The distance graph for Case 3

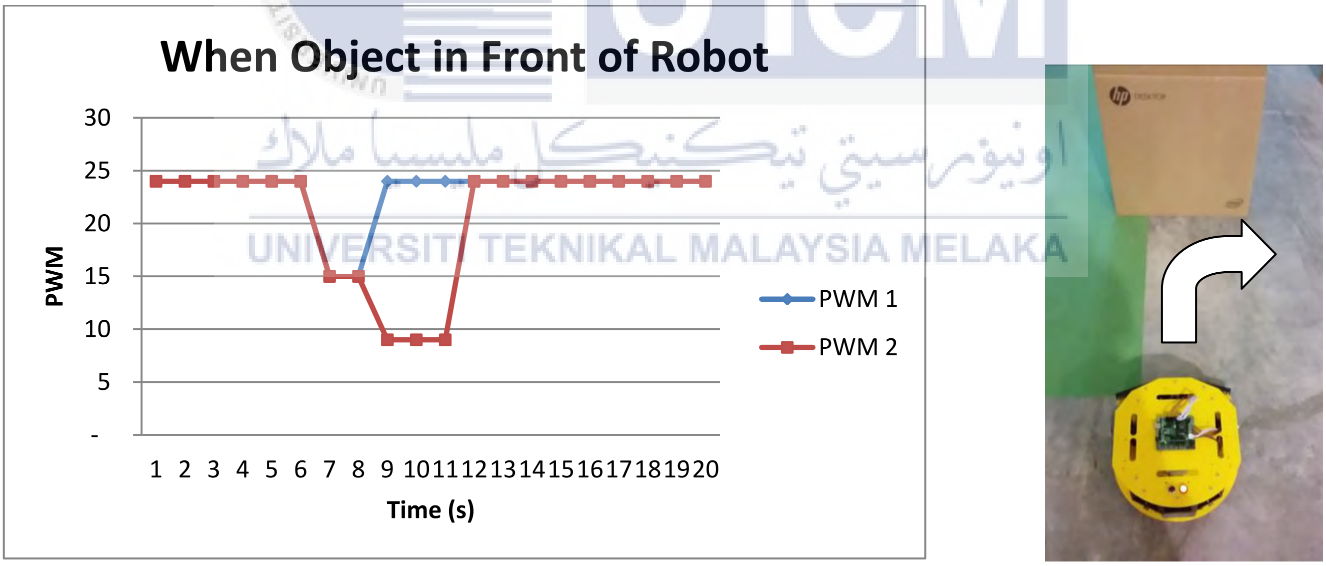


Figure 4.11: The PWM graph for Case 3

Meanwhile for this experiment, from the graph above along the time interval from 1s to 8s show that the reading of ultrasonic sensor 1 (D1) have decreased from 98 cm to 68 cm, ultrasonic sensor 2 (D2) also decreased from 92 cm to 40 cm in time interval 1s to 10s and ultrasonic sensor 3 (D3) decreased from 93 cm to 27 cm along time interval 1s to 11s.

Both PWM1 and PWM2 gave out the reading of 24 at time interval 1s until 6s. The robot start to slowing down at time interval 7s to 8s as it approach the obstacle . The PWM1 and PWM2 gave out reading 15.

At time interval 9s until 11s, the PWM1 is 24 and PWM2 is 9. At this moment, the robot start to turn left since the input of the fuzzy logic controller have meet the fuzzy rules no. 23, If (*D1* Far, *D2* Medium & *D3* Medium) then (*PWM1* Fast & *PWM2* Very Slow) and rule no.25, If (*D1* Far, *D2* Far & *D3* Near) then (*PWM1* Fast & *PWM2* Very Slow). After reading 11s, the robot has successfully avoid the obstacle by turning right as all sensor gave out reading out of range at time interval 12s to 13s. However, at reading time interval 14s to 17s, the D1 and D2 gave out the distance reading as they detect the other obstacle after turning right. This reading may consider as error from this experiment.

4.7 Experimental Result Case 4 (When two obstacles that far apart in front of the robot)

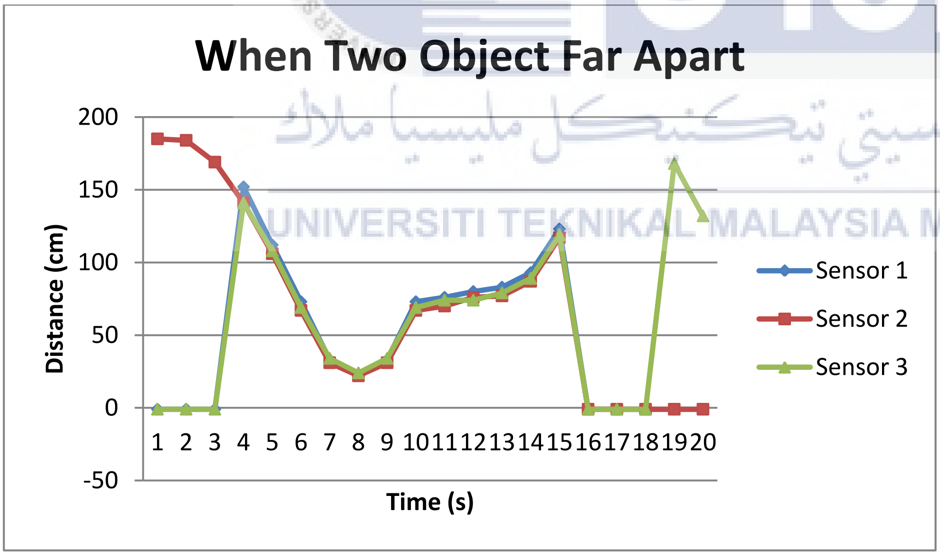


Figure 4.12: The distance graph for Case 4

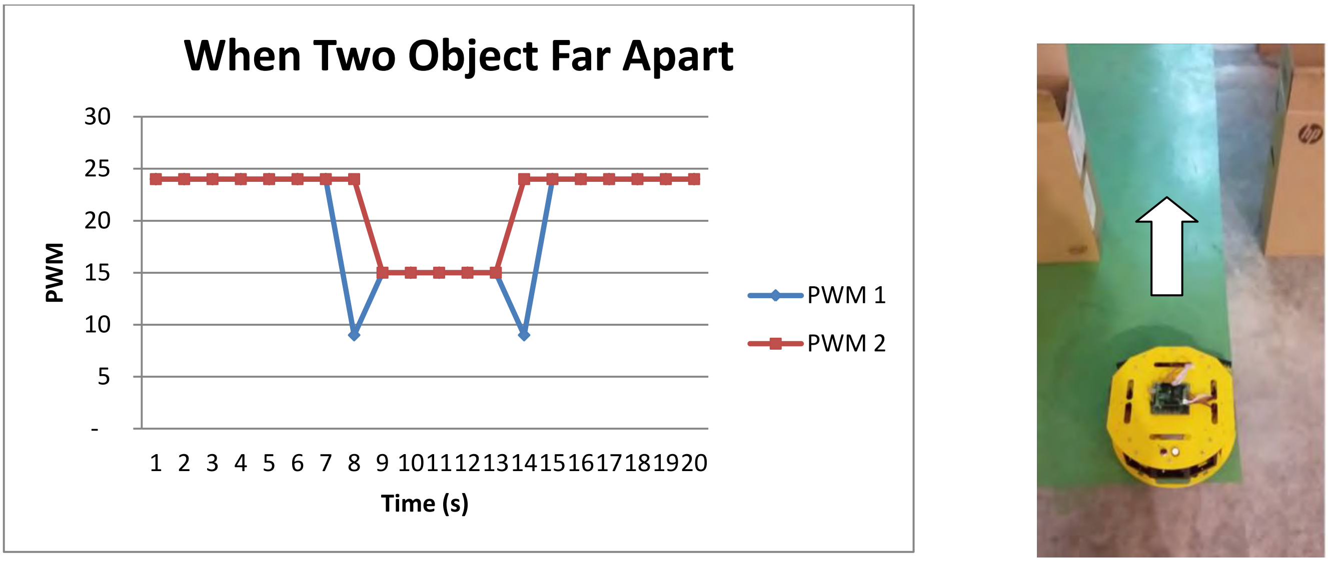


Figure 4.13: The PWM graph for Case 4

Besides that for this experiment, from the graph above the time interval from 1s to 3s show that the reading of the ultrasonic sensor 1 (D1) and ultrasonic sensor 2 (D2) are -1. At reading 4s, both D1 and D3 start to gave out reading of 152 cm and 141 cm respectively and the readings are decreased until reading of 9s. The ultrasonic sensor 2 (D2) is decreased along time interval 1s until 9s from 185cm to 31 cm. The PWM1 and PWM2 maintained at 24 from 1s to 7s.

The robot start slowing down at time interval 9s to 13s as the robot is going through between two obstacle. The PWM1 and PWM2 gave out reading 15. The distance between two obstacles is 60 cm. Meanwhile, at reading of 8s and 14s, the PWM1 is 9 and PWM2 is 24. This happen due to the robot has adjusting its steering angle so that the robot can move through that two obstacles. At time interval 10s to 15s the distance reading of D1, D2 and D3 increased from 73 cm to 123 cm for D1, 67 cm to 117 cm for D2 and 69 cm to 119 cm for D3. At reading of 16s until 20s, all sensor gave out reading out of range except at 19s and 20s the D3 gave out the reading of 168 cm and 132 cm respectively as it detect the other obstacle after going through two obstacle. This reading may consider as error from this experiment.

4.8 Experimental Result Case 5 (When two obstacles near of each other in front of the robot)

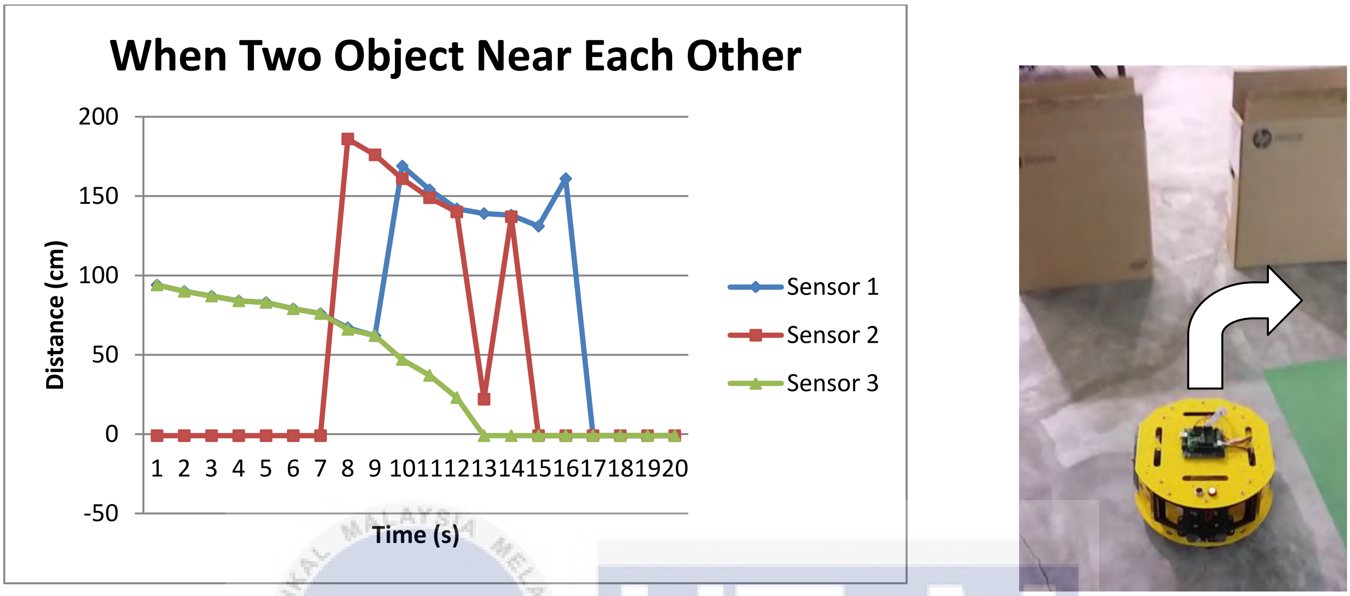


Figure 4.14: The distance graph for Case 5

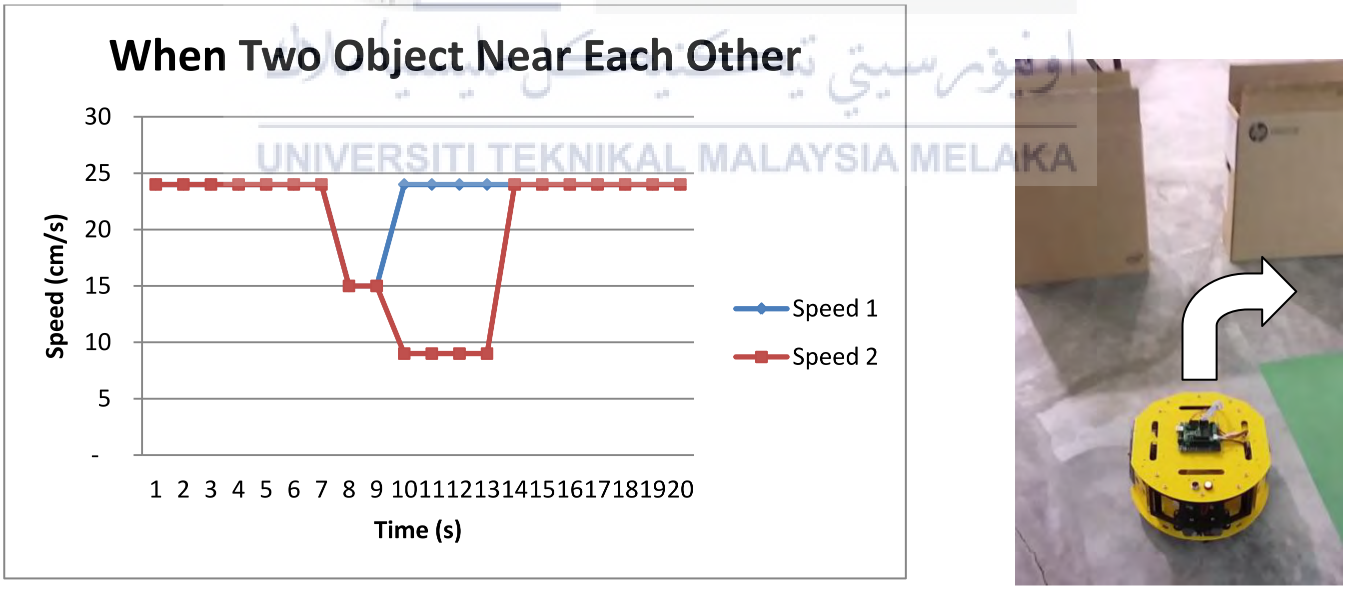


Figure 4.15: The PWM graph for Case 5

From the graph above, along time interval of 1s to 9s the ultrasonic sensor 1 (D1) decreased from 94 cm to 62 cm, while ultrasonic sensor 3 (D3) decreased from 94 cm to 23 cm along time interval of 1s to 12s. At 1s until 7s, the ultrasonic sensor 2 (D2) gave out reading of -1. This is because the D2 can't detect the gap between two obstacles. The gap of two obstacles is 30 cm width. The PWM1 and PWM2 gave out reading of 24 from time interval 1s until 7s.

The robot is slowing down as the PWM1 and PWM2 gave out reading of 15. At time interval 10s until 13s, the PWM1 and PWM2 are 24 and 9 respectively which mean the robot is turning right to avoid the obstacle. At this moment, the input of the fuzzy logic controller have meet the fuzzy rules no.25, If (*D1* Far, *D2* Far & *D3* Near) then (*PWM1* Fast & *PWM2* Very Slow) and fuzzy rules no. 21, If (*D1* Far, *D2* Near & *D3* Far) then (*PWM1* Fast & *PWM2* Very Slow). After reading 14s, the robot has successfully avoid the obstacle by turning right. Before all sensors gave reading out of range, the D1 and D2 still gave out the reading of obstacle at time interval 15s to 16s as the robot nearly collide the obstacle at right side. At 17s, all sensor gave out the out of range reading. In this experiment, the mobile robot success avoid both obstacle by turning right.

4.9 Experimental Result Case 6 (When the robot need to do a u-turn instead of moving forward)

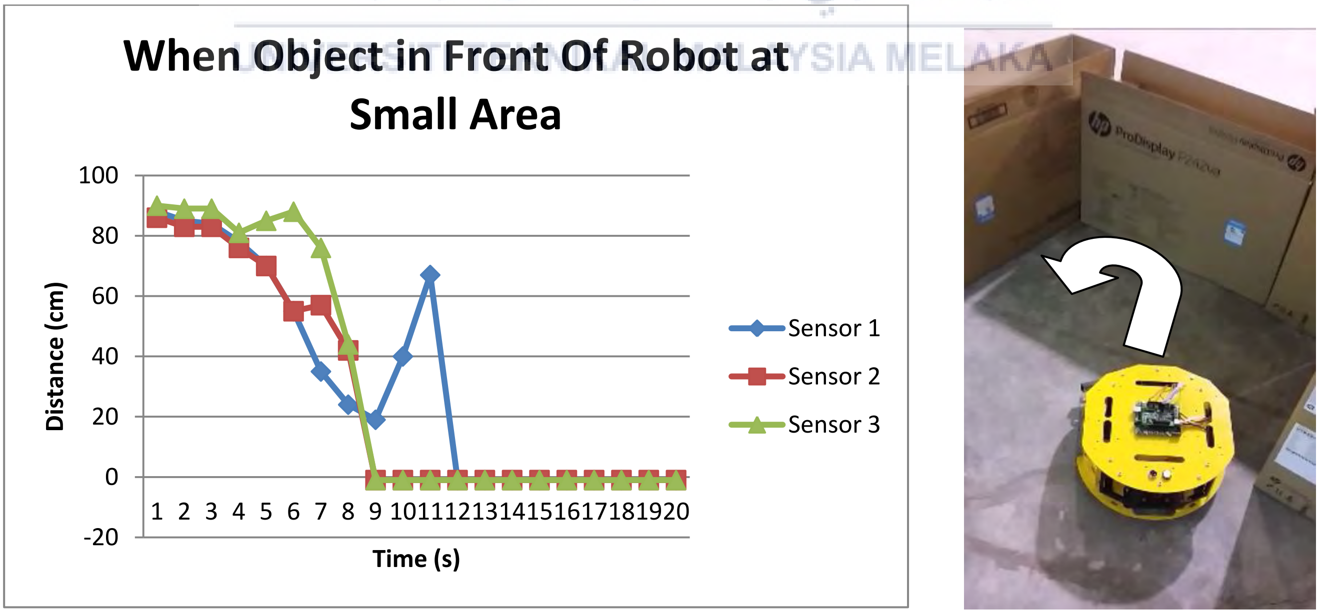


Figure 4.16: The distance graph for Case 6

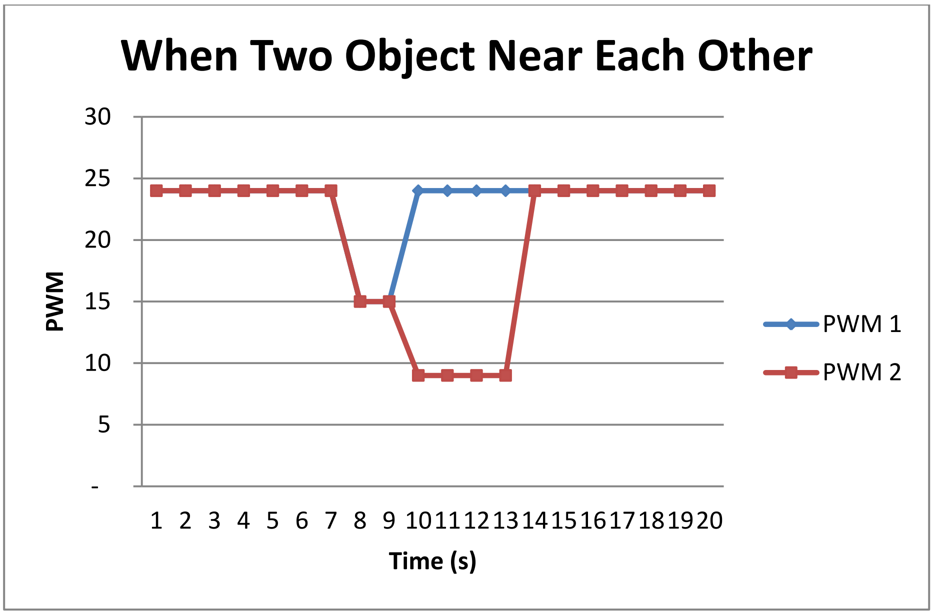


Figure 4.17: The PWM graph for Case 6

Based on the graph above, at time interval 1s to 9s, the distance reading of the ultrasonic sensor 1 (D1) decreased from 88 cm to 19 cm, the ultrasonic sensor 2 (D2) also decreased from 86 cm to 42 cm at time interval 1s to 8s and at time interval 1s to 8s, the distance reading of ultrasonic sensor 3 (D3) decreased from 90 cm to 44 cm. Both PWM1 and PWM2 maintained at 24 from 1s until 4s time interval.

At time interval 6s to 11s, the PWM1 and PWM2 gave out reading 9 and 24 respectively as the robot start to turn left until it complete the u-turn since the input of the fuzzy logic controller have meet the fuzzy rules no.15, If (*D1* Medium, *D2* Medium & *D3* Far) then (*PWM1* Very Slow & *PWM2* Fast), rule no.5, If (*D1* Near, *D2* Medium & *D3* Medium) then (*PWM1* Very Slow & *PWM2* Fast) and rule no.18, If (*D1* Medium, *D2* Far & *D3* Far) then (*PWM1* Very Slow & *PWM2* Fast). After reading 11s, the robot has been successfully do a u-turn to avoid the obstacle and go to the spacious area as all sensors gave out reading of out of range until at 20s reading.

4.10 Summary of The Results

As for the summary of the results, all experiments have successfully been done in order to test the performance of robot in avoiding obstacles using static object involving six situations. Table 4.6 show that the success rate in term of actual situation involving all cases. From this table, only case 1 and case 6 have 100% success rate as there are no error during the robot move to avoid the obstacle. Eventhough the other cases have not reach 100% success rate, but the robot still successfully in avoiding obstacles at different positions.

Table 4.6 : Success Rate in term of Actual Situation

Cases	**Success Rate by Percentage (%)**
1	100
2	95
3	85
4	85
5	85
6	100

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the sensory construction accurately measures obstacles immediately in front of the sensor and within the field of view of D1 with at test 82.6% accuracy, 95.92% at D2 and 85.33% at D3. Meanwhile, when obstacles at right side of the sensor and within the field of view of D1, the accuracy is 98.06% and 93.5% at D2. When the sensor sense the obstacle at left side of the sensor and within the field of view of D2, the accuracy is 88.66% while at D3 is 100%. The fuzzy logic controller used as control system to control the movement of the robot in terms of speed of the motors and the direction of the robot in avoiding obstacle. The fuzzy logic controller was developed using Mamdani technique and implemented to the robot based on the fuzzy logic inference system, which is composed of three inputs, two outputs, and 27 fuzzy rules. Multiple membership functions for inputs and outputs were developed. After that the robot was tested in six different situation such as Case 1 : when the obstacle at left side of the robot, Case 2 : when the obstacle at right side of the robot, Case 3 : when the obstacle is in front of the robot, Case 4 : when two obstacles that far apart in front of the robot, Case 5 : when two obstacles near of each other in front of the robot and Case 6 : when the robot need to do a u-turn instead of moving forward. The obstacle used in the test was a static object (boxes).

5.2 Recommendation

As for recommendation, Raspberry pi is needed to replace Arduino microcontroller. This is because the memory for Raspberry pi is bigger than memory of Arduino microcontroller as Raspberry pi's memory is 512 MB while the biggest memory for Arduino microcontroller, Atmega is only 12KB. Besides that, by using Raspberry pi, more programming and algorithm of fuzzy logic controller can be store and it can run faster.

Next, the optimization for algorithm can improve the delay in response motor. So that, the movement of robot can be more smooth especially in situation that the motor need to change its speed for avoiding the obstacle.

REFERENCE

[1] Masaki Takahashi & Rosario Sorbello, A Mobile Robot for Transport Applications in Hospital Domain with Safe Human Detection Algorithm, 2009.

[2] Ren C. Luo, Po K. Wang, Yu F.Tseng & T.Y.Lin , Navigation and Mobile Security System of Home Security Robot, 2006.

[3] Widodo Budiharto, Intelligent Surveillance Robot with Obstacle Avoidance Capabilities Using Neural Network, 2015.

[4] Mohammed Faisal*, Khalid Al-Mutib, Ramdane Hedjar, Hassan Mathkour, Mansour Alsulaiman, and Ebrahim Mattar, Behavior based Mobile for Mobile Robots Navigation and Obstacle Avoidance, 2014.

[5] Nassima Melik & Noureddine Slimane, Autonomous Navigation with Obstacle Avoidance of Tricycle Mobile Robot Based on Fuzzy Controller, 2015

[6] C. G. Rusu, I. T. Birou, E. Szöke, Fuzzy Based Obstacle Avoidance System for Autonomous Mobile Robot, 2010

[7] Marwah Almasri, Khaled Elleithy, and Abrar Alajlan, Sensor Fusion Based Model for Collision Free Mobile Robot Navigation, 2015

[8] Hiroyuki Takai, Masamichi Miyake, Kiyoshi Okuda and Keihachiro Tachibana, A Simple Obstacle Arrangement Detection Algorithm For Indoor Mobile Robot, 2010

[9] Jiann-Der Lee and Zih-Yang Dang, Dual-sensor fusion for obstacle avoidance in indoor environment, 2015

[10] M. Izzuddin Ibrahim, Nohaidda Sariff, Juliana Johari, Norlida Buniyamin, Mobile Robot Obstacle Avoidance in Various Type of Static Environments Using Fuzzy Logic Approach, 2014

[11] Xi Li & Byung-Jae Choi, Design of Obstacle Avoidance System for Mobile Robot using Fuzzy Logic Systems, 2013.

[12] Umar Farooq, K. M. Hasan, Ghulam Abbas, Muhammad Usman Asad, Comparative Analysis of Zero Order Sugeno and Mamdani Fuzzy Logic Controllers for Obstacle Avoidance Behavior in Mobile Robot Navigation, 2011

[13] SUN Yang-zhi, XIAO Shi-de, HAO Meng-jie, PAN Shao-fei, Study on obstacle avoidance for intelligent robot based on hierarchical fuzzy control, 2015

[14] Mostafa Sharifi and Prof. XioQi Chen, Introducing A Novel Vision Based Obstacle Avoidance Technique for Navigation of Autonomous Mobile Robots, 2015

APPENDIX A

The Distance Reading for Accuracy Test for Ultrasonic Sensors

Distance Reading Of Each Sensor When The Object In Front Of The Mobile Robot

Ultrasonic Sonar Sensor (D1)

Table A.1: Ultrasonic sonar sensor 1 reading when the object in front of the robot

Expected Distance (cm)	**Actual Distance 1 (cm)**	**Actual Distance 2 (cm)**	**Actual Distance 3 (cm)**	**Average Distance (cm)**
10	14	14	14	11
20	24	24	24	21
30	35	35	35	31
40	45	45	45	41
50	54	54	54	51
60	64	64	65	64.33

Ultrasonic Sonar Sensor (D2)

Table A.2: Ultrasonic sonar sensor 2 reading when the object in front of the robot

Expected Distance (cm)	Actual Distance 1 (cm)	Actual Distance 2 (cm)	Actual Distance 3 (cm)	Average Distance (cm)
10	11	11	11	11
20	21	21	21	21
30	31	31	31	31
40	41	41	41	41
50	51	51	51	51
60	61	61	61	61

Ultrasonic Sonar Sensor (D3)

Table A.3: Ultrasonic sonar sensor 3 reading when the object in front of the robot

Expected Distance (cm)	Actual Distance 1 (cm)	Actual Distance 2 (cm)	Actual Distance 3 (cm)	Average Distance (cm)
10	13	13	13	13
20	24	24	24	24
30	34	34	34	34
40	44	44	44	44
50	54	54	54	54
60	64	64	64	64

Distance Reading Of Each Sensor When The Object At The Right Side Of The Mobile Robot

Ultrasonic Sonar Sensor (D1)

Table A.4: Ultrasonic sonar sensor 1 reading when the object in at the right side of robot

Expected Distance (cm)	Actual Distance 1 (cm)	Actual Distance 2 (cm)	Actual Distance 3 (cm)	Average Distance (cm)
10	11	11	11	11
20	20	20	20	20
30	30	30	30	30
40	40	40	40	40
50	50	50	50	50
60	59	59	59	59

Ultrasonic Sonar Sensor (D2)

Table A.5: Ultrasonic sonar sensor 2 reading when the object in at the right side of robot

Expected Distance (cm)	Actual Distance 1 (cm)	Actual Distance 2 (cm)	Actual Distance 3 (cm)	Average Distance (cm)
10	11	11	11	11
20	22	22	22	22
30	32	32	32	32
40	42	42	42	42
50	52	52	52	52
60	62	62	62	62

Distance Reading Of Each Sensor When The Object At The Left Side Of The Mobile Robot

Ultrasonic Sonar Sensor (D2)

Table A.6: Ultrasonic sonar sensor 2 reading when the object in at the left side of robot

Expected Distance (cm)	**Actual Distance 1 (cm)**	**Actual Distance 2 (cm)**	**Actual Distance 3 (cm)**	**Average Distance (cm)**
10	13	13	12	12.67
20	23	23	24	23.33
30	33	33	33	34
40	42	43	42	42.33
50	53	53	53	53
60	62	62	62	62

Ultrasonic Sonar Sensor (D3)

Table A.7: Ultrasonic sonar sensor 3 reading when the object in at the left side of robot

Expected Distance (cm)	**Actual Distance 1 (cm)**	**Actual Distance 2 (cm)**	**Actual Distance 3 (cm)**	**Average Distance (cm)**
10	10	10	10	10
20	20	20	20	20
30	30	30	30	30
40	40	40	40	40
50	50	50	50	50
60	60	60	60	60

Distance Reading for Accuracy Of Ultrasonic Sonar Sensor At 50 cm Distance

Table A.8: Ultrasonic sonar sensor reading at 50 cm Distance

Angle of Range Detection (º)	Actual Distance 1 (cm)	Actual Distance 2 (cm)	Actual Distance 3 (cm)	Average Distance (cm)
-35	-	-	-	-
-30	51	51	51	51
-20	49	49	49	49
-10	49	49	49	49
0	49	49	49	49
10	48	48	48	48
20	50	50	50	50
30	51	51	51	51
35	-	-	-	-

Table of collected data to test the performance of mobile robot in avoiding obstacle based on six situations.

Case 1 : When the obstacle at left side of the robot

Table A.9: Ultrasonic sensor reading for Case 1

Time (s)	Sensor 1	Sensor 2	Sensor 3
1	-1	96	99
2	-1	94	96
3	-1	91	94
4	-1	88	89
5	74	67	66
6	-1	38	35
7	-1	-1	16
8	-1	-1	10
9	-1	-1	10
10	-1	-1	9
11	-1	-1	13
12	-1	-1	-1
13	-1	-1	-1
14	-1	-1	-1
15	-1	-1	-1
16	-1	-1	-1
17	-1	-1	-1
18	-1	-1	-1
19	-1	-1	-1
20	-1	-1	-1

Table A.10: DC motor reading for Case 1

Time (s)	PWM 1	PWM 2
1	24	24
2	24	24
3	24	24
4	24	24
5	15	15
6	24	9
7	24	9
8	24	9
9	24	9
10	24	9
11	24	9
12	24	24
13	24	24
14	24	24
15	24	24
16	24	24
17	24	24
18	24	24
19	24	24
20	24	24

Case 2 : When the obstacle at right side of the robot

Table A.11: Ultrasonic sensor reading for Case 2

Time (s)	Sensor 1	Sensor 2	Sensor 3
1	87	84	-1
2	85	82	-1
3	82	79	-1

4	79	76	-1
5	70	63	66
6	41	36	41
7	20	17	-1
8	26	24	-1
9	34	32	-1
10	45	42	-1
11	54	50	-1
12	60	57	-1
13	68	65	-1
14	75	71	-1
15	80	77	-1
16	79	78	-1
17	77	76	-1
18	37	40	-1
19	-1	-1	-1
20	-1	-1	-1

Table A.12: DC motor reading for Case 2

Time (s)	PWM 1	PWM 2
1	24	24
2	24	24
3	24	24
4	24	24
5	15	15
6	15	15
7	9	24
8	9	24
9	9	24
10	9	24
11	9	24
12	9	24

13	9	24
14	9	24
15	24	24
16	24	24
17	24	24
18	9	24
19	24	24
20	24	24

Case 3 : When the obstacle is in front of the robot

Table A.13: Ultrasonic sensor reading for Case 3

Time (s)	Sensor 1	Sensor 2	Sensor 3
1	98	92	93
2	95	89	90
3	91	87	88
4	88	86	85
5	84	83	82
6	79	80	78
7	72	73	72
8	68	67	67
9	-1	53	53
10	-1	40	38
11	-1	-1	27
12	-1	-1	-1
13	-1	-1	-1
14	76	77	-1
15	80	79	-1
16	83	77	-1
17	92	-1	-1
18	-1	-1	-1

19	-1	-1	-1
20	-1	-1	-1

Table A.14: DC motor reading for Case 3

Time (s)	PWM 1	PWM 2
1	24	24
2	24	24
3	24	24
4	24	24
5	24	24
6	24	24
7	15	15
8	15	15
9	24	9
10	24	9
11	24	9
12	24	24
13	24	24
14	24	24
15	24	24
16	24	24
17	24	24
18	24	24
19	24	24
20	24	24

Case 4 : When two obstacles that far apart in front of the robot

Table A.15: Ultrasonic sensor reading for Case 4

Time (s)	Sensor 1	Sensor 2	Sensor 3
1	-1	185	-1

2	-1	184	-1
3	-1	169	-1
4	152	141	141
5	112	106	108
6	73	67	69
7	33	31	34
8	23	22	24
9	33	31	34
10	73	67	69
11	76	70	74
12	80	76	74
13	83	77	79
14	93	87	89
15	123	117	119
16	-1	-1	-1
17	-1	-1	-1
18	-1	-1	-1
19	-1	-1	168
20	-1	-1	132

Table A.16: DC motor reading for Case 4

Time (s)	PWM 1	PWM 2
1	24	24
2	24	24
3	24	24
4	24	24
5	24	24
6	24	24
7	24	24
8	9	24
9	15	15
10	15	15

11	15	15
12	15	15
13	15	15
14	9	24
15	24	24
16	24	24
17	24	24
18	24	24
19	24	24
20	24	24

Case 5 : When two obstacles near of each other in front of the robot

Table A.17: Ultrasonic sensor reading for Case 5

Time (s)	Sensor 1	Sensor 2	Sensor 3
1	94	-1	94
2	90	-1	90
3	87	-1	87
4	84	-1	84
5	83	-1	83
6	79	-1	79
7	76	-1	76
8	67	186	66
9	62	176	62
10	169	161	47
11	154	149	37
12	142	140	23
13	139	22	-1
14	138	137	-1
15	131	-1	-1
16	161	-1	-1

17	-1	-1	-1
18	-1	-1	-1
19	-1	-1	-1
20	-1	-1	-1

Table A.18: DC motor reading for Case 5

Time (s)	PWM 1	PWM 2
1	24	24
2	24	24
3	24	24
4	24	24
5	24	24
6	24	24
7	24	24
8	15	15
9	15	15
10	24	9
11	24	9
12	24	9
13	24	9
14	24	24
15	24	24
16	24	24
17	24	24
18	24	24
19	24	24
20	24	24

Case 6 : When the robot need to do a u-turn instead of moving forward

Table A.19: Ultrasonic sensor reading for Case 6

Time (s)	Sensor 1	Sensor 2	Sensor 3
1	88	86	90
2	85	83	89
3	84	83	89
4	78	76	81
5	70	70	85
6	55	55	88
7	35	57	76
8	24	42	44
9	19	-1	-1
10	40	-1	-1
11	67	-1	-1
12	-1	-1	-1
13	-1	-1	-1
14	-1	-1	-1
15	-1	-1	-1
16	-1	-1	-1
17	-1	-1	-1
18	-1	-1	-1
19	-1	-1	-1
20	-1	-1	-1

Table A.20: DC motor reading for Case 6

Time (s)	PWM 1	PWM 2
1	24	24
2	24	24
3	24	24
4	24	24

5	9	24
6	9	24
7	9	24
8	9	24
9	9	24
10	9	24
11	9	24
12	24	24
13	24	24
14	24	24
15	24	24
16	24	24
17	24	24
18	24	24
19	24	24
20	24	24

APPENDIX B

Algorithm for Obstacle Avoidance Mobile Robot

```
#include <MotorWheel.h>
#include <SONAR.h>          // Include the header files
#include "fis_header.h"

#define D1   (11)
#define D2   (12)
#define D3   (13)
#define PWM1 (9) //MotorLeft
#define PWM2 (10) //MotorRight

// Number of inputs to the fuzzy inference system
const int fis_gcI = 3;
// 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 = 27;

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

```
SONAR s11(0x11);
SONAR s12(0x12);
SONAR s13(0x13);

 /* PWM1 ||-----|| PWM2   */

void setup() {
   Serial.begin(19200);
  // initialize the Analog pins for input.
  // Pin mode for Input: D1 (s11)
    pinMode(D1 , INPUT);
   // Pin mode for Input: D2 (s12)
    pinMode(D2 , INPUT);
   // Pin mode for Input: D3 (s13)
    pinMode(D3 , INPUT);
    pinMode(8  , OUTPUT);
    pinMode(11 , OUTPUT);

   // initialize the Analog pins for output.
   // Pin mode for Output: MotorRight (PWM 2)
    pinMode(PWM1 , OUTPUT);
   // Pin mode for Output: MotorLeft (PWM 1)
    pinMode(PWM2 , OUTPUT);

    SONAR::init();      //set up some parameters
    delay(100);    //100 millissecond
}
void loop()
```

```
{
digitalWrite(8,1);
digitalWrite(11,0);

//S11 IS FOR RIGHT SONAR
    s11.trigger();       //Send the trigger command to trigger S11
   int MSB1 = s11.getDist();
   int LSB1 = s11.getDist();
   Serial.print("sensor 1 : ");
   Serial.println(LSB1);//Display the distance S11 received.
   delay(10);

   s12.trigger();
   int MSB2 = s12.getDist();
   int LSB2 = s12.getDist();
   Serial.print("sensor 2 : ");
   Serial.println(LSB2);
   delay(10);

   s13.trigger();
   int MSB3 = s13.getDist();
   int LSB3 = s13.getDist();
   Serial.print("sensor 3 : ");
   Serial.println(LSB3);
   delay(10);

 if (LSB1<0) {
  LSB1=100;
}
```

```
 if (LSB2<0) {
  LSB2=100;
}
 if (LSB3<0) {
  LSB3=100;
}

  fis_evaluate();
  analogWrite(PWM1 , g_fisOutput[0]);
  Serial.print("PWM1 : ");
  Serial.println(g_fisOutput[0]);
  analogWrite(PWM2 , g_fisOutput[1]);
  Serial.print("PWM2 : ");
  Serial.println(g_fisOutput[1]);

    // Read Input: D1 (s11)
    g_fisInput[0] = LSB1;
    // Read Input: D2 (s12)
    g_fisInput[1] = LSB2;
    // Read Input: D3 (s13)
    g_fisInput[2] = LSB3;

    g_fisOutput[0] = 0;
    g_fisOutput[1] = 0;
}
//***********************************************************************
// 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[] = { 3, 3, 3 };

// Count of member function for each Output
int fis_gOMFCount[] = { 4, 4 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { 0, 15, 30 };
FIS_TYPE fis_gMFI0Coeff2[] = { 29, 50, 75 };
FIS_TYPE fis_gMFI0Coeff3[] = { 74, 150, 250 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2, fis_gMFI0Coeff3
};
FIS_TYPE fis_gMFI1Coeff1[] = { 0, 15, 30 };
FIS_TYPE fis_gMFI1Coeff2[] = { 74, 150, 250 };
FIS_TYPE fis_gMFI1Coeff3[] = { 29, 50, 75 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2, fis_gMFI1Coeff3
};
```

```
FIS_TYPE fis_gMFI2Coeff1[] = { 0, 15, 30 };

FIS_TYPE fis_gMFI2Coeff2[] = { 29, 50, 75 };

FIS_TYPE fis_gMFI2Coeff3[] = { 74, 150, 250 };

FIS_TYPE* fis_gMFI2Coeff[] = { fis_gMFI2Coeff1, fis_gMFI2Coeff2, fis_gMFI2Coeff3
};

FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff, fis_gMFI2Coeff };


// Coefficients for the Input Member Functions

FIS_TYPE fis_gMFO0Coeff1[] = { 12, 15, 18 };

FIS_TYPE fis_gMFO0Coeff2[] = { 18, 24, 30 };

FIS_TYPE fis_gMFO0Coeff3[] = { 0, 0, 0 };

FIS_TYPE fis_gMFO0Coeff4[] = { 0, 9, 18 };

FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2,
fis_gMFO0Coeff3, fis_gMFO0Coeff4 };

FIS_TYPE fis_gMFO1Coeff1[] = { 12, 15, 18 };

FIS_TYPE fis_gMFO1Coeff2[] = { 18, 24, 30 };

FIS_TYPE fis_gMFO1Coeff3[] = { 0, 0, 0 };

FIS_TYPE fis_gMFO1Coeff4[] = { 0, 9, 18 };

FIS_TYPE* fis_gMFO1Coeff[] = { fis_gMFO1Coeff1, fis_gMFO1Coeff2,
fis_gMFO1Coeff3, fis_gMFO1Coeff4 };

FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff, fis_gMFO1Coeff };


// Input membership function set

int fis_gMFI0[] = { 0, 0, 0 };

int fis_gMFI1[] = { 0, 0, 0 };

int fis_gMFI2[] = { 0, 0, 0 };

int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1, fis_gMFI2};


// Output membership function set

int fis_gMFO0[] = { 0, 0, 0, 0 };

int fis_gMFO1[] = { 0, 0, 0, 0 };
```

```
int* fis_gMFO[] = { fis_gMFO0, fis_gMFO1};


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


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


// Rule Inputs
int fis_gRI0[] = { 1, 1, 1 };
int fis_gRI1[] = { 1, 1, 2 };
int fis_gRI2[] = { 1, 1, 3 };
int fis_gRI3[] = { 1, 3, 1 };
int fis_gRI4[] = { 1, 3, 2 };
int fis_gRI5[] = { 1, 3, 3 };
int fis_gRI6[] = { 1, 2, 1 };
int fis_gRI7[] = { 1, 2, 2 };
int fis_gRI8[] = { 1, 2, 3 };
int fis_gRI9[] = { 2, 1, 1 };
int fis_gRI10[] = { 2, 1, 2 };
int fis_gRI11[] = { 2, 1, 3 };
int fis_gRI12[] = { 2, 3, 1 };
int fis_gRI13[] = { 2, 3, 2 };
int fis_gRI14[] = { 2, 3, 3 };
int fis_gRI15[] = { 2, 2, 1 };
int fis_gRI16[] = { 2, 2, 2 };
int fis_gRI17[] = { 2, 2, 3 };
int fis_gRI18[] = { 3, 1, 1 };
int fis_gRI19[] = { 3, 1, 2 };
```

```
int fis_gRI20[] = { 3, 1, 3 };

int fis_gRI21[] = { 3, 3, 1 };

int fis_gRI22[] = { 3, 3, 2 };

int fis_gRI23[] = { 3, 3, 3 };

int fis_gRI24[] = { 3, 2, 1 };

int fis_gRI25[] = { 3, 2, 2 };

int fis_gRI26[] = { 3, 2, 3 };

int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4, fis_gRI5, fis_gRI6,
fis_gRI7, fis_gRI8, fis_gRI9, fis_gRI10, fis_gRI11, fis_gRI12, fis_gRI13, fis_gRI14,
fis_gRI15, fis_gRI16, fis_gRI17, fis_gRI18, fis_gRI19, fis_gRI20, fis_gRI21, fis_gRI22,
fis_gRI23, fis_gRI24, fis_gRI25, fis_gRI26 };


// Rule Outputs

int fis_gRO0[] = { 3, 3 };

int fis_gRO1[] = { 4, 2 };

int fis_gRO2[] = { 4, 2 };

int fis_gRO3[] = { 3, 3 };

int fis_gRO4[] = { 4, 2 };

int fis_gRO5[] = { 4, 2 };

int fis_gRO6[] = { 3, 3 };

int fis_gRO7[] = { 4, 2 };

int fis_gRO8[] = { 4, 2 };

int fis_gRO9[] = { 3, 3 };

int fis_gRO10[] = { 1, 1 };

int fis_gRO11[] = { 4, 2 };

int fis_gRO12[] = { 2, 4 };

int fis_gRO13[] = { 1, 1 };

int fis_gRO14[] = { 4, 2 };

int fis_gRO15[] = { 2, 4 };

int fis_gRO16[] = { 1, 1 };

int fis_gRO17[] = { 4, 2 };
```

```
int fis_gRO18[] = { 2, 4 };

int fis_gRO19[] = { 2, 4 };

int fis_gRO20[] = { 2, 4 };

int fis_gRO21[] = { 2, 4 };

int fis_gRO22[] = { 2, 4 };

int fis_gRO23[] = { 1, 1 };

int fis_gRO24[] = { 2, 4 };

int fis_gRO25[] = { 2, 4 };

int fis_gRO26[] = { 2, 2 };

int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5,
fis_gRO6, fis_gRO7, fis_gRO8, fis_gRO9, fis_gRO10, fis_gRO11, fis_gRO12,
fis_gRO13, fis_gRO14, fis_gRO15, fis_gRO16, fis_gRO17, fis_gRO18, fis_gRO19,
fis_gRO20, fis_gRO21, fis_gRO22, fis_gRO23, fis_gRO24, fis_gRO25, fis_gRO26 };


// Input range Min

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


// Input range Max

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


// Output range Min

FIS_TYPE fis_gOMin[] = { 0, 0 };


// Output range Max

FIS_TYPE fis_gOMax[] = { 30, 30 }

//***********************************************************************

// 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, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };
    FIS_TYPE fuzzyInput2[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, fuzzyInput2, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0, 0 };
    FIS_TYPE fuzzyOutput1[] = { 0, 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);
       }
    }
}
```