UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Faculty Of Electrical Engineering

THE INTEGRATION OF FUZZY LOGIC SYSTEM FOR OBSTACLE AVOIDANCE BEHAVIOUR OF MOBILE ROBOT

AERUN A/L MARTIN

Bachelor Degree of Mechatronics Engineering

2017

THE INTEGRATION OF FUZZY LOGIC SYSTEM FOR OBSTACLE AVOIDANCE BEHAVIOUR OF MOBILE ROBOT

AERUN A/L MARTIN

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

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

I declare that this report entitled "The Integration of Fuzzy Logic System for Obstacle Avoidance Behaviour of Mobile Robot" is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ....

Name : Aerun A/L Martin

Date : ....

"I hereby declare that I have read through this report entitle "The Integration of Fuzzy Logic System for Obstacle Avoidance Behaviour of Mobile Robot" and found that it has comply the partial fulfilment for awarding the degree of Bachelor of Mechatronics Engineering.

Signature : ....

Supervisor's Name : Cik Nur Ilyana Binti Anwar Apandi

Date : ....

To my beloved mother and father

ACKNOWLEDGEMENT

First and foremost, I would like to express my immeasurable appreciation and deepest gratitude to University of Technical Malacca (UTeM) for providing an opportunity for me to undertake my Final Year Project in partial fulfilment for Bachelor of Mechatronics Engineering.

I am deeply indebted towards my project supervisor, Cik Nur Ilyana Binti Anwar Apandi for her patience guidance and keen interest on me at every stage of my project progression. Her prompt encouragements, timely assistance, erudite advice, and warm kindness have motivate me to perform better and widen my research boundaries in the completion of my Final Year Project. Apart from that, I thank profusely all the final year project student under supervision of Cik Nur Ilyana for all the generous efforts, assistances and mentoring provided in enlightening me. Their notable efforts in teaching me to use and conduct software simulation are deeply appreciated.

Thanks are also extended to all my course mate, their assistance and support throughout the completion of my research. Also, I would take this opportunity to express my gratitude to my parents for their continuous shower of love, unceasing encouragement and support throughout all these years.

Last but not least, I place on record, my sense of gratitude to one and all who, directly or indirectly, have offered their helping hand during the entire period of Final Year Project.

ABSTRACT

A mobile robot is an automatic machine that has the capability of sensing its environment, understanding the sensed information to receive the knowledge of its location and surrounding environment, planning a real-time path from a starting position to goal position with obstacle avoidance, and controlling the robot steering angle and its speed to reach the target. Mobile robots are the ideal solution for the process of eliminating landmines which is an explosive device that usually planted under the ground to destroy or disable enemy targets once they have direct contact by applying right amount of pressure to trigger the landmine. Based on statistic report from Landmine & Cluster Munition Monitor about 3,678 people has been victim of landmines explosion around the world in year 2014. Eliminating the landmines using the mobile robots can improve the safety of personnel as well as the efficiency, productivity and flexibility of the work. In order to carry out such an important tasks mobile robot should be equipped with a robust controller to analyze the input and output that help the mobile robot to navigate in an uncertain environment without colliding with any obstacles. Fuzzy Logic Controllers is an intelligent technique that proves to be the one of the most reliable controller that suits well for nonlinear system like robot due to the simple control based on user input without any prior knowledge to the mathematical model. Thus, this project develops two controller of Fuzzy Logic System which are Mamdani and Sugeno controller by using an input from five proximity sensor of differential drive robot to control the linear velocity of left and right motor. The control focuses on linear velocity to avoid considering the kinematic model which will be required for control of angular velocity. The smoothness and efficiency the generated by Mamdani and Sugeno is analyzed based on simulation of Pioneer P3-DX robot in virtual robotic software for single and multirobot environments with static obstacles.

ABSTRAK

Sebuah robot mudah alih adalah sebuah mesin automatik yang mempunyai keupayaan penderiaan persekitarannya, memahami maklumat yang dikesan untuk menerima pengetahuan lokasi dan persekitaran, merancang laluan masa nyata dari tempat yang bermula hingga ke kedudukan gol dengan mengelakkan halangan, dan mengawal sudut robot dan kelajuan untuk mencapai sasaran. Robot mudah alih merupakan penyelesaian yang sesuai untuk proses menghapuskan periuk api yang merupakan bahan letupan yang ditanam di bawah tanah untuk memusnahkan atau melumpuhkan sasaran musuh dengan menggunakan tekanan untuk mencetuskan periuk api itu. Berdasarkan laporan statistik dari "Landmine & Kluster Munition Monitor" dianggarkan 3678 orang telah menjadi mangsa letupan periuk api di seluruh dunia pada tahun 2014. Robot mudah alih boleh meningkatkan keselamatan kakitangan serta kecekapan, produktiviti dan fleksibiliti kerja dalam menghapuskan periuk api. Dalam usaha untuk menjalankan apa-apa tugas-tugas penting robot mudah alih dilengkapi dengan pengawal teguh untuk menganalisis input dan output yang membantu robot mudah alih untuk menavigasi dalam persekitaran yang tidak menentu tanpa berlanggar dengan apa-apa halangan.Sistem Logik Fuzzy adalah teknik yang dibuktikan untuk menjadi salah satu pengawal yang paling boleh dipercayai yang sesuai dengan baik bagi sistem bukan linear seperti robot kerana kawalan yang mudah berdasarkan input pengguna tanpa pengetahuan sebelum model matematik. Oleh itu, projek ini dikembangkan untuk menggunakan dua jenis Sistem Logik Fuzz, iaitu Mamdani dan Sugeno dengan mendapatkan input daripada lima sensor kedekatan pengkamiran memandu robot untuk mengawal had laju linear motor kiri dan kanan. Kawalan ini memberi tumpuan kepada had laju linear untuk mengelakkan mempertimbangkan model kinematik yang diperlukan untuk mengawal had laju sudut. Kelancaran dan kecekapan yang dihasilkan oleh Mamdani dan Sugeno dianalisis berdasarkan simulasi "Pioneer P3-DX" robot dalam perisian robot maya untuk persekitaran 'singlerobot' dan 'multirobot' dengan halangan statik.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

m	:	Meter
m/s	:	Meter per second

LIST OF ABBREVIATION

FLS	:	Fuzzy Logic System
FLC	:	Fuzzy Logic Controller

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

A mobile robot is an automatic machine that has the capability of sensing its surrounding. Mobile robot able to understand the sensed information to receive the knowledge of its location, to plan a real-time path from a starting position to goal position with obstacle avoidance capability, and to control the robot steering angle and its speed to reach the target. Mobile robots could be utilized in different applications such as monitoring, transportation and many other potential applications. The ability 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. But the most reported designs rely on intelligent control approaches such as Fuzzy Logic System. FLS is a powerful soft computing technique to control complex and non-linear systems based on human expert knowledge. This project emphasis on the integration of FLS for the obstacle avoidance behavior of mobile robots.

1.1 Motivation

The removal of the dangerous debris in any form particularly landmines after a war is a very important process for any region to guarantee the safety of people in areas that have been cleared. A land mine is an explosive device that usually planted under the ground to destroy or disable enemy targets as they have direct contact by applying right amount of pressure to trigger the landmine. The areas where the landmines are planted become unusable or usable only at significant risk and preventing the areas to be developed as a part of economic growth of a country. Approximately more than 100 million mines are still left to be found that can lead to significant hazards or fatalities around the world as stated by United Nations Department of Human Affairs (UNDHA) in [1].

The unremoved landmines around the world causes carnage every year. A statistic reported in [2] by Landmine & Cluster Munition Monitor shows that 3,678 people has been victim of landmines explosion around the world in year 2014. Figure 1.1 illustrates that 80% unarmed civilians including children, women and elderly and the rest 18% and 2% were security forces and definers respectively. The statistic is solely based on the recorded cases. The true casualty figure is more likely to be higher than the current causality rate if unrecorded cases are recorded.

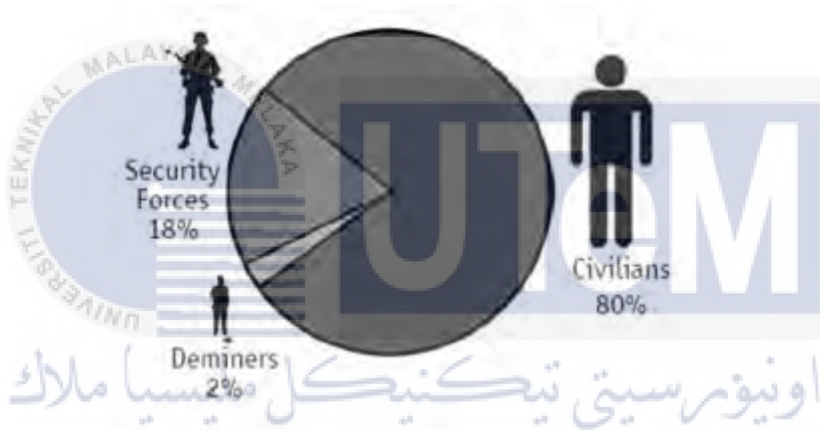


Figure 1.1: Percentage of Victim of Landmine [2]

In [3], approximately around 300,000-400,000 landmine survivors who may be out of death zone but they still suffering from permanent disability, musculoskeletal injuries and loss of home. There are more than 25,000 amputees in Cambodia due to mine blasts whereas one of every 334 individuals are suffering from landmine amputation. Due to the increasing number of victims, the direct cost of medical treatment exceeds US$750 million as reported in [1]. This makes the government unable to provide medical assistance to the victims due to lack of medical subsidies. Other than that, these people also will be denied access to their living land and its resources to avoid any causalities. This may lead to loss of biodiversity and country will suffer from economy crisis.

Despite the number of causalities, this deadly weapon is still widely used as a tactic to win a war. To solve this severe problem various method are using to remove the landmines from the contaminated area. The procedure of clearance of landmine should be well prepared and followed accordingly so that the confidence level that the area is free from landmine will be higher. The mines can be neutralized by either removal or detonation using various technique such as manual demining, the use of animals, insects and bacteria, mechanical demining and robot demining [4]. Eliminating the landmines using the robots is the ideal solution because this technique can improve the safety of personnel as well as the efficiency, productivity and flexibility of the work. Due to the uncertainty of the environment, using autonomous robots that able to coordinate their movement by avoiding obstacles and reaching the position of the landmine will speed up the process of detection and elimination of landmines [5], [6], and [7]. These robots also need a robust controller to analyze the input and output that help the mobile robot to move in an uncertain environment without colliding with any obstacles.

1.2 Problem Statement

Mobile robot needs a robust controller to adapt the fast integration between the input and output due to the navigation in uncertain environment. Due to nonlinearity property of mobile robot, it is difficult to obtain absolute mathematical model of a system for designing its controller [8]. Amongst the various techniques available in this paradigm, Fuzzy Logic Controller offers a promising solution to handle the vague and imprecise information because Fuzzy based controller does not require mathematical model of the system [9].

Many mobile robots use a drive mechanism known as differential drive where each wheel is independently driven by an actuator. Thus, the direction of mobile robot can be controlled by vary the linear velocity (m/s) or angular velocity (rev/sec) of left and right wheels. Since the control of angular velocity needs prior knowledge on kinematic model of the robot [10] which consists of complex mathematical terms, the control of linear velocity provides an easy solution to control the direction of robot as it does not require any mathematical modelling.

There are two type of FLC, Takagi-Sugeno [11] and Mamdani [12] controller. The efficiency of these two controllers for obstacle avoidance behaviour of robot has been studied by many researchers. However, there are less contribution towards the comparison between these two controllers for obstacle avoidance behaviour of mobile robot. So, comparative comparison between Mamdani and Sugeno type FLCs for obstacle avoidance behaviour of mobile robot able to contribute for future works.

1.3 Objectives

1. To develop Mamdani FLC and Sugeno FLC for obstacle avoidance behaviour of mobile robot to control the linear velocity of the left and right wheels.
2. To validate the developed Mamdani FLC and Sugeno FLC in a mobile robot using robot simulator.
3. To compare the smoothness and efficiency of Mamdani FLC and Sugeno FLC in various environment.

1.4 Scopes

This project focuses on developing and comparing Mamdani FLC and Sugeno FLC for the obstacle avoidance behavior of mobile robot. Both system is developed using Fuzzy Toolbox in Matlab based on the Pioneer P3-DX robot under the library of Virtual Robot Experimentation Platorm (V-REP). The readings of five proximity sensors of Pioneer P3-DX robot were chosen as input, whereas linear velocity of left wheel and right wheels are chosen as output for both Mamdani FLC and Sugeno FLC. The target seeing behavior is achieved using distance and angular formula based on the coordinate and orientation provided by SICK S300 Professional laser scanner in V-REP robotic simulator. Only static obstacles are taken into consideration during the simulation. The simulation by integrating V-rep and Matlab via is achieved by remote Application Program Interface (API) configuration.

1.5 Thesis Organization

This report and project is about the development of FLCs for obstacle avoidance behaviour of mobile robot. There are total 5 chapter in this report where each chapter is divided into subsections which will discuss on specific topic in detailed. In chapter 1, the motivation for designing FLC for mobile robot will be explain, then followed by problem statement, objective, and scope of this project. While in chapter 2, method and technique used from previous related work is discussed. Synthesis and analysis based on previous related work also will covered in chapter 2. Several theoretical analyses of the body of method and principles associated with the project needed to accomplish in this project will be described in detail in chapter 3. Fourth chapter will be covered the results and analysis from the simulation followed by final chapter where conclusion of the overall project and recommendation for future works is explained.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the background of the project will be explained briefly for better understanding of the research. A review of previous related works will be discussed to obtain some useful information by synthesizing their work to make this research successful.

2.2 Robot

Czech Playwright Karel Capek depicted robots as machine which resembles people [13] with ability to function on their own to perform hard, strenuous, hazardous, repetitive, boring or dangerous works reliably, repeatedly, and accurately with little or no attention from human [14] until there is any technical error. Robots able to work with high precision, high speed and high level of strength to complete various tasks including tasks that are harmful to human being such as space or underwater exploration, volcano craters, chemical spill clean-up, nuclear waste disposal, demining.

2.2.1 Multi Robot System (MRS)

A multi robot system can be defined as a group of robots cooperatively carrying out a common task to complete the tasks with more precision and less completion time compared to a single robot. The concept of this multi-agent system is becoming an important tool to design intelligent and complex software applications due to their ability to carry out some peculiar tasks, such as cooperative localization, cooperative behaviour, cooperative planning and cooperative control [15]. Multi robot system able to complete the given tasks without any interruption, as many robots works as a group, the process will be continued till the end by the robots even one of the robots in the group could not perform the task due to any technical error.

2.3 Mobile Robots

Mobile robots are a type of robot that are capable of ground locomotion as they able to react to the physical properties of the environment through its sensors to identify features, to detect pattern and regularities, to learn from experience, to localize, to build map and to navigate without colliding with any obstacles in an unknown or known environment [13]. Autonomous mobile robots are equipped with controller that can take own decision based on the environment and take necessary action to execute the tasks such as transportation, exploration, surveillance, guidance and inspection with minimal or no intervention from any human operator.

2.3.1 Mobile Robots Sensory System

Mobile robot uses a variety of sensors to collect data by interacting to outside world or external environment by providing an accurate data for decision making. The type of sensors that used in mobile robot is depend on the application of the mobile robot. The sensing capabilities of a robot can be classified as local or global [16]. Local sensing captures information about the immediate environment through on board sensors such as proximity sensors and vision sensors. Global sensing captures a global perspective of the environment with an external sensor like GPS or an overhead camera.

2.3.1.1 Proximity Sensors

Proximity sensors abundantly used in mobile robotics due low cost and simple mechanism compared to vision sensor. Proximity sensors basically detect the presence of objects without physical contact. A proximity sensor detects objects when the objects approach within the detection range and boundary of the sensor. Proximity sensors includes all sensors that perform non-contact such as ultrasonic sensors and infrared sensors.

2.3.1.1.1 Ultrasonic Sensors

Ultrasonic sensor or ultrasonic range finders sensor that transmits sound wave [16] and wait for the reflected signal to measure the range of the obstacles. The obstacles only will be detected if the obstacles are located somewhere on the arc of radius r within angular detection range as in Figure 2.1. In [17], the author states that, the researchers showing more interest in equip their mobile robots with ultrasonic sensors due to its simplicity, low-cost and the distance measurements are provided directly. However, ultrasonic sensors suffer from some drawbacks such as multiple reflections, wide radiation cone, low angular resolution, absorption and specular reflection which can lead to detection of the distance parameter from the robot and obstacles with high error [18].

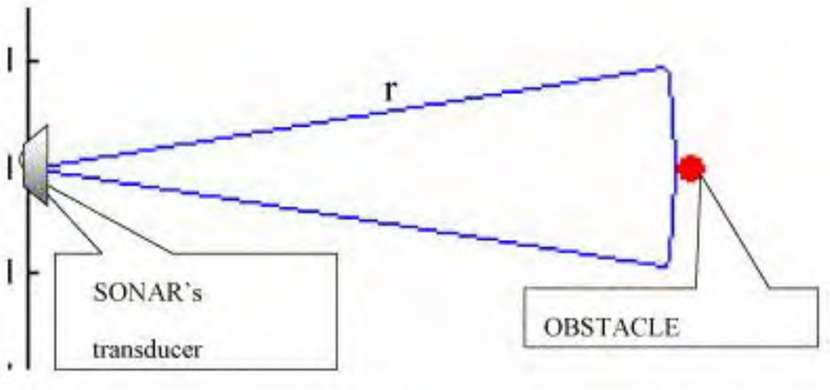


Figure 2.1: Registration of an obstacle by an ultrasonic range finder [17]

2.3.1.1.2 Infrared Proximity Sensor

Infrared sensor emits infrared ray which exists in electromagnetic spectrum to sense the obstacles. This infrared sensor has an IR transmitter and a position sensitive device. The position sensitive device is an optical detector which can detect the light falling on a plane. By processing the signal from position sensitive device and interpreting the signal gives the distance of the obstacle in front of it. The sensing mechanism of the infrared sensors are depicted in figure 2.2. Since this infrared ray is projected in straight angle, many sensors is needed to detect the obstacles in multiple angle [19].

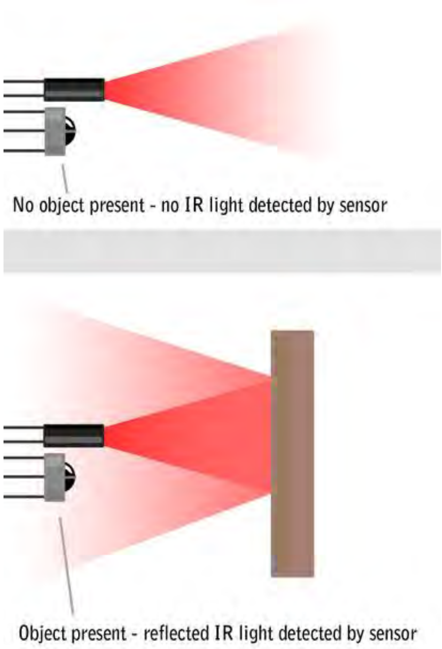


Figure 2.2: Depiction of the operation of an IR sensor during obstacle detection [19]

2.3.1.2 Vision Sensor

Robot scan the environment and identify objects by analysing the pixel-based images using vision sensors such as charged- couple devices. This process is need complex control system to adapt the complex sensing and perception process involves extracting, characterizing and interpreting information from several images to detect, identify and describe objects obstacles along the moving path of the robot. A vision sensor usually mounted in a camera to converts the captured visuals information to electrical signal which will be processed by the electronic system and further converts into digital image to be processed by the computer [16]. According to authors in [20] states simple sensors are not enough to provide sophisticated information for advanced robot applications for advanced analysis and accurate decision-making of robots. However, due to the complexity of data processing, this sensor is expensive and only uses in high-tech mobile robots.

2.3.1.3 Summary on Mobile robot sensory system

Table 2.1: Mobile robot sensory system

Sensors	Proximity Sensors		Vision Sensors
Type	Ultrasonic Sensor	Infrared Sensor	Charge-coupled device (CCD)
Detection of obstacles	Using sound wave	Using infrared ray	Using pixel-based images
Accuracy	Low	Low (higher than ultrasonic)	High
Detection range	High	Low	High
Interphase	Easy	Easy	Hard
Price	Low (higher than Infrared)	Low	High

Both proximity sensors and vision sensor are used in mobile robot to integrate with outside world to detect the presence of obstacles. But the use of sensors differs due to the characteristics of sensors. Vision sensor able to detect more precisely than other two proximity sensors. This is due the image processing of obstacle by vision sensor able to visualize the shape and position of obstacles. This helps the robot to avoid the obstacles more precisely than other two sensors. Vision sensors has more range detection than the Infrared sensors and ultrasonic sensors due to high resolution of the sensors. But these advantages of the vision sensor make it very hard for the interphase with controllers and expensive. Thus, vison sensors only able to equip in high end mobile robots. Infrared and ultrasonic sensors both give direct feedback to the controller unlike vision sensors which needs image processing and converts the image to electical signal before giving feedback to controller. The only difference in these two sensors are, the detection of obstacle is done by sound wave for ultrasonic whereas the electromagnetic for infrared sensors. Ultrasonic sensors are not prone to outside interference such as noise or light intensity, but these sensors' detection may differ due to the its own echoes which makes the sensors unable to detect obstacles which are too close. Unlike the ultrasonic sensors, infrared sensors are prone to the light intensity and can cause detection error if the light intensity is not controlled properly but able to detect short range obstacles. Thus, designing the controller for the vision sensors based system is more complex than ultrasonic and infrared sensors. In this project, short distance detection and cheaper sensors are needed to complete the system. So, infrared proximity sensors are more suitable for this sensor based mobile robot navigation.

2.3.2 Collision avoidance technique of mobile robot

Collision avoidance or obstacle avoidance is very essential issue that need to be implemented in the mobile robots. While moving in uncertainty area, mobile robot may face a lot of obstacles. The obstacles may be static or dynamic obstacles depend on the mobile robot roaming area. The mobile robots should consist of ability to avoid any kind of obstacles to carry out any specified tasks without any interruption. In the multi robot system the dynamic obstacles are commonly another robot. Avoiding collision with each other is very crucial tasks for each robot in multi robot system.

2.3.2.1 Potential Field

The potential field method is primarily developed by Kathib [21] for autonomous mobile robot to avoid obstacles. This method become popular due to the simple mathematical analysis compared to other approaches. In this method, a repulsive force potential field will be built around the obstacles and gravitational potential field will be built in target location. These two force will create a composite artificial potential field to move the robot. The robot will move to the target location by the mean of attraction of the gravitational potential field and avoid the obstacles by the mean of repelling to the repulsive force around the obstacles [22].

2.3.2.2 Artificial Neural Network

Artificial Intelligence is a new technique that able to adapt the behaviour of human. One of the AI technique that inspired from physical structure of biological neurons and nervous system are Artificial Neural Networks (ANNs) that adjust the connection weight that links the processing units (artificial neurons) to solve the path planning problem for mobile robots. that distributed in layers to reduce the classification error. Multiple Layer Perceptron (MLP) [23] is considered as popular architecture of ANN which characterized by the presence of at least one hidden layer that located between the input layer and the respective output layer of neurons and can solve non-linear problems especially obstacles avoidance.

2.3.2.3 Genetic Algorithm

The development of Genetic Algorithm starts from selecting random candidate solution of the optimization problem which represents as "chromosome" [24]. This chromosome will be modified over the time repeatedly for a new generation with a better chromosome. This algorithm able to estimates the advantages in each chromosome. At each step, the GA able to selects the population randomly to use it as parents to produce children for new population based on a natural selection process that mimics biological evolution. For each successive generation, the population gets better to produce an optimal solution. The best sequence of actions that executed by previous robots will have more chance to be executed again by current robot. [25].

2.3.2.4 Particle Swarm Optimization (PSO)

Particle swarm optimization Algorithm (PSO) is another approach for multi robot path planning. In [18], a classical PSO which known as CPSO which utilizes swarm intelligence such as behaviour of school of fish schooling or bird flocks to reach the optimization goal. The velocity parameter of the CPSO will be updated by the particles based on their own experience. To find the best position in the search space, the information will be passes through the entire members to change each particle position. In [26], the authors modified the CPSO equation to improved PSO by proposing the PSO in the terms of adaptive weight adjustment and acceleration coefficients to increase the convergence rate to optimum value in PSO which the proposed technique is proved to be best over other PSO technique for navigation of multi-mobile robot.

2.3.2.5 Fuzzy Logic System

FLS serves as another intelligent technique used in the design of local navigation, global navigation, path planning, steering control and speed control of a mobile robot. FLS is widely used approach as it is provides user friendly interface controller [27] since FLS is a combination of many forms of logic values of the inputs. The FLS offers a nonlinear control with robustness for any system with uncertain parametric and function, as well as disturbances. FLS consists of four modules as shown Figure 2.3 to measure all the inputs and analyse them as per user defined rules to compute the output. Two common FLS are zero order Takagi-Sugeno [11] and Mamdani [12]. Both system compute the output as per the block diagram shown in Figure 2.4. The only differences between these two controllers are lies in the process called defuzzification.

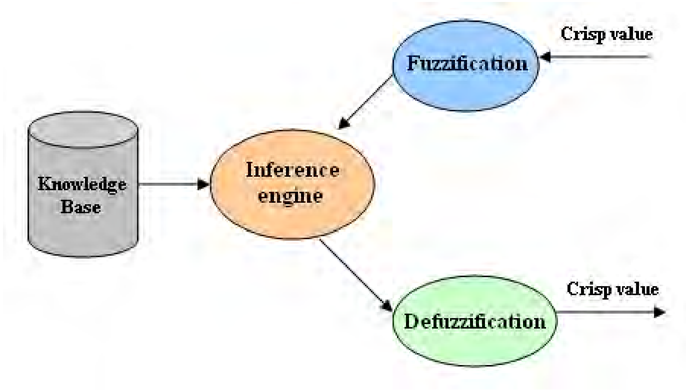


Figure 2.3: Four modules of FLS [27]

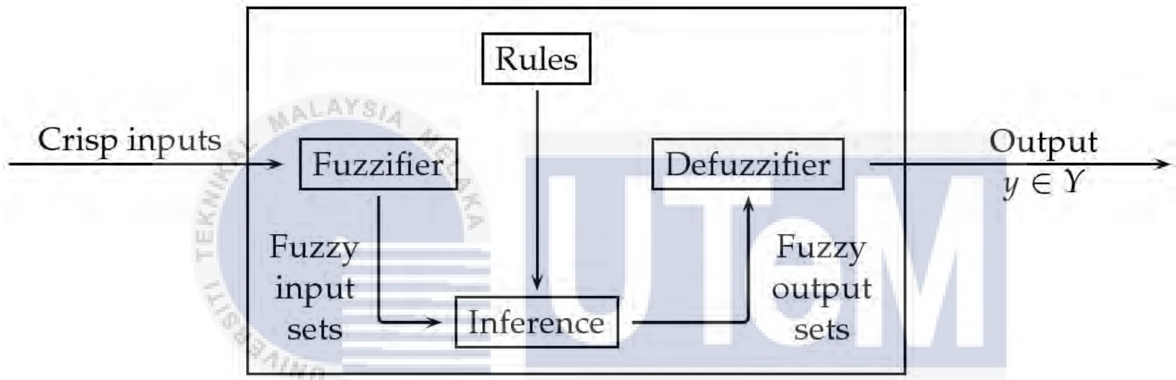


Figure 2.4: Block diagram of FLS [27]

2.3.2.5.1 Mamdani FLC

Mamdani's FLC method is the most commonly seen fuzzy method. Mamdani's method was among the first control systems built using fuzzy set theory. It was proposed in 1975 by Ebrahim Mamdani [28] as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. Mamdani's effort was based on Lotfi Zadeh's 1973 paper on fuzzy algorithms for complex systems and decision processes [29]. Figure 2.5 is an illustration of how a two-rule Mamdani FLC derives the overall output z when subjected to two crisp inputs x and y.

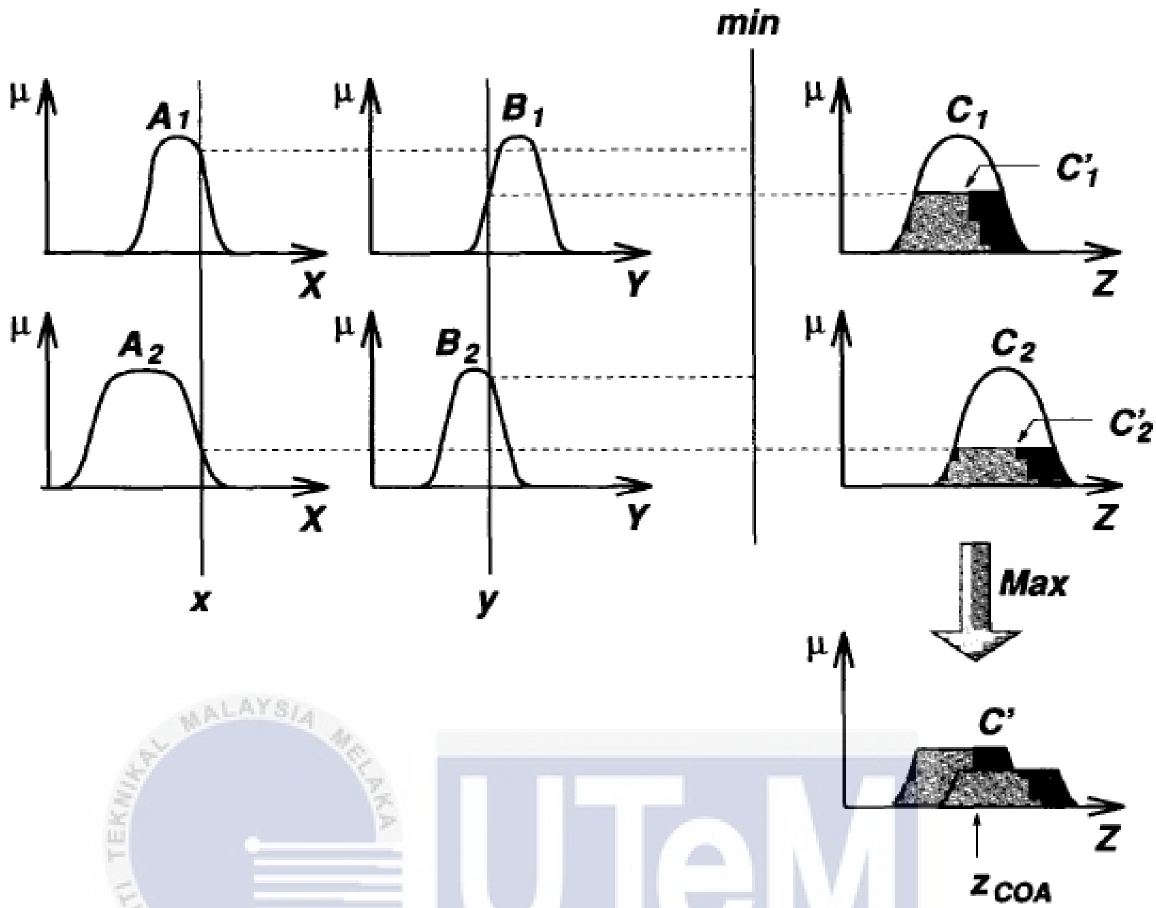


Figure 2.5: The Mamdani FLC using min and max operator [27]

Since the output for Mamdani FLC is in fuzzy sets, these fuzzy sets need to be converted to crisp output by process called defuzzification. There are several defuzzification methods such as centroid area, bisector of area, mean of maximum, smallest of maximum and largest of maximum. But the most popular one is the centroid technique. It finds the point where a vertical line would slice the aggregate set into two equal masses. The formula of centroid defuzzification to determine the crisp output is as in Equation (1) based on Figure 2.6.

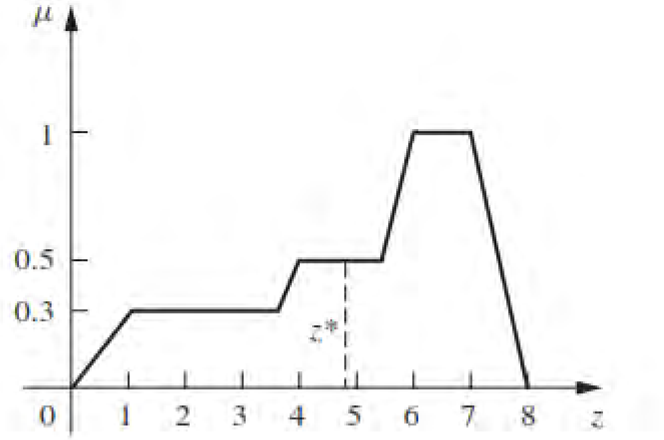


Figure 2.6: Sampe output of Mamdani FLC [27]

Formula of defuzzification for Figure 2.6 is given by,

$$z^* = \frac{\int_z \mu_A z \, dz}{\int_z \mu_A \, dz} \tag{1}$$

The calculation needed to carry out this defuzzification operations is time-consuming unless special hardware support is available. Furthermore, these defuzzification operations are not easily subject to rigorous mathematical analysis, so most of the studies are based on experimental results.

2.3.2.5.2 Sugeno FLC

Takagi Sugeno Kang method of FLC is introduced in 1985 [30], this method is similar to the Mamdani method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator are the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant. Mamdani-style inference requires finding the centroid of a two-dimensional shape by integrating across a continuously varying function. In general, this process is not computationally efficient. Michio Sugeno suggested to use a single spike, a singleton, as the membership function of the rule consequent. A singleton, or more precisely a fuzzy singleton, is a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else. A typical fuzzy rule in a Sugeno fuzzy model has the form,

If Input 1 is x=A and Input 2 is y=B, then Output is z= ax + by + c (2)

For zero-order Sugeno fuzzy model the output level z is a constant. Each rule weights its output level, z_i, by the firing strength of the rule, w_i. For example, for an AND rule with,

Input 1 = x and Input 2 = y, (3)

the firing strength is,

$$W_i = \text{AndMethod}\ (F1(x), F2(y)) \quad (4)$$

where F1(x), F2(y) are the membership functions for Inputs 1 and 2 respectively. The final output of the system is the weighted average of all rule outputs, given by,

$$\text{Final Output} = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \quad (5)$$

where N =is the number of rules. Figure 2.7 shows the fuzzy reasoning procedure for a first-order Sugeno fuzzy model. Since each rule has a crisp output, the overall output is obtained via weighted average, thus avoiding the time-consuming process of defuzzification required in a Mamdani model.

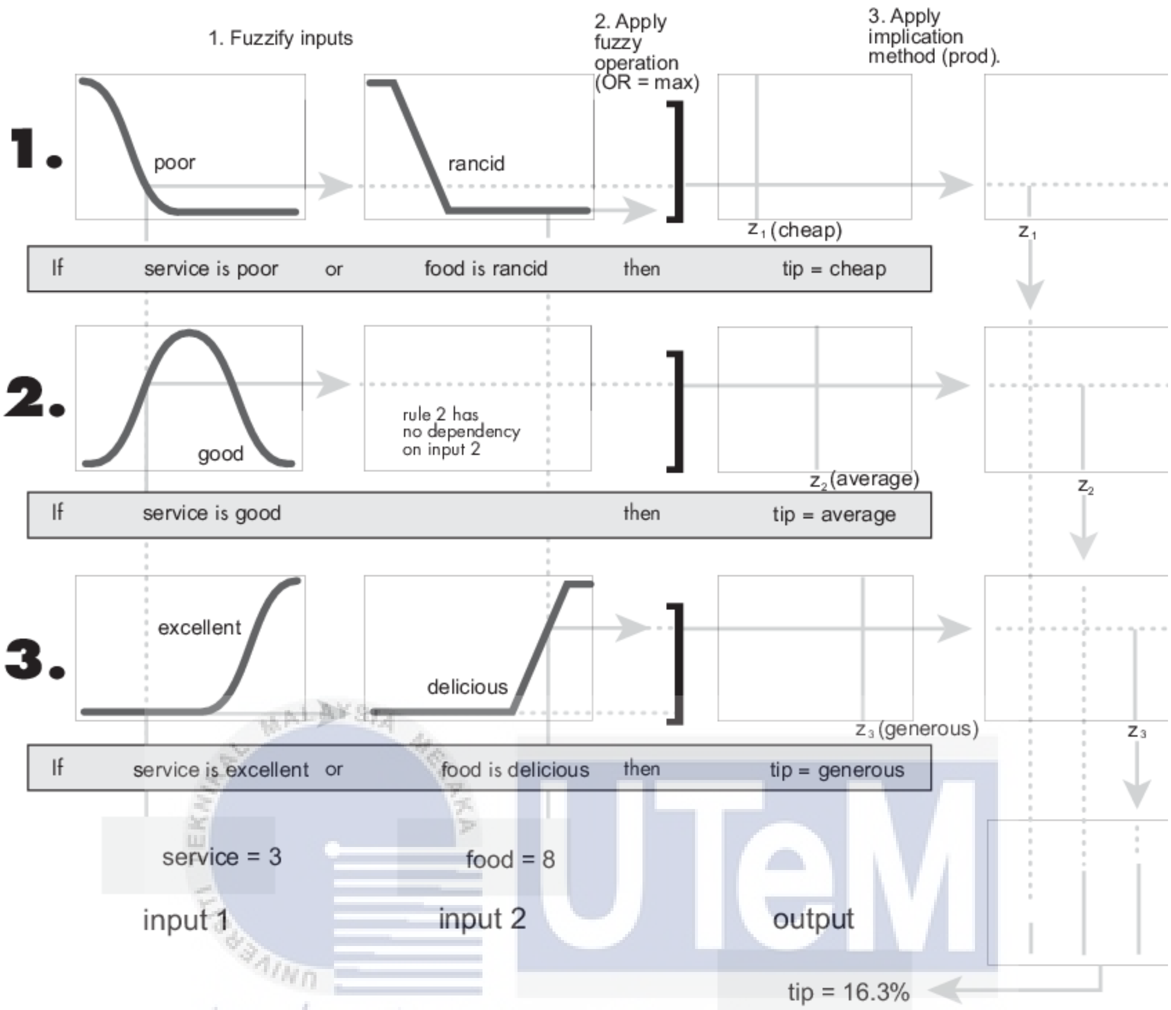


Figure 2.7: Working principle of Takagi-Sugeno FLC [27]

2.3.2.6 Summary on Obstacles Avoidance Approach

In contrast with the all the approaches in collision avoidance strategies, fuzzy offers a cheap, reliable, and simpler system. Fuzzy does not require any complex mathematical modelling as PSO and GA that makes it more suitable controller for non-linear systems such as mobile robots. Fuzzy also proves as the flexible controller as it can combine with other other controllers to produce a hybrid controllers that gives more efficient results. Some of the hybrid controllers that enhances FLS are Fuzzy – PID controller [31], fuzzy- genetic algorithm [32], neuro-fuzzy [33] and etc. The two most common fuzzy technique used in mobile robotics is

Mamdani and Sugeo. Each FLC, Mamdani and Sugeno techniques can be used to customize the membership functions so that the fuzzy system best models the data. In overall comparison Sugeno is more compact and computationally efficient representation than a Mamdani system. Thus, the Sugeno system lends itself to the use of adaptive techniques for constructing fuzzy models. However, these two system has their own advantages as depicted in Table 2.2.

Table 2.2: Advantages of Sugeno and Mamdani

Advantages of Mamdani	It is computationally efficient.
	It works well with linear techniques (e.g., PID control).
	It works well with optimization and adaptive techniques.
	It has guaranteed continuity of the output surface.
	It is well suited to mathematical analysis.
Advantages of Sugeno	It is intuitive.
	It has widespread acceptance.
	It is well suited to human input.

2.4 Related Research on FLS Based Obstacles Avoidance

Fuzzy design controller usually emphasis on three particular fuzzy systems: fuzzy steering, fuzzy linear velocity control and fuzzy angular velocity control for obstacle avoidance and target seeking.

In [34]- [35], the authors have developed Mamdani FLC that has four input, position error, angular error between the orientation of the robot and the target, angle between the axis carrying the obstacle and the reference mark center and distance between the robot and obstacle, whereas the output of this system is steering angle and speed of mobile robot. In [34] author did not mention about the type of sensor used and number of rules applied for this system but mobile robot used by author have prior knowledge on position of obstacles to avoid and move in a safe path. In [35], the authors have use 25 rues the to let the fuzzy control to navigate the robot in safe position from initial point to final point. However, in this two works, authors have derived the kinematic model of the robot to control the angular velocity. Even the process of deriving is complex to solve, the path generated by the robots are smooth as shown in Figure 2.8 and Figure 2.9.

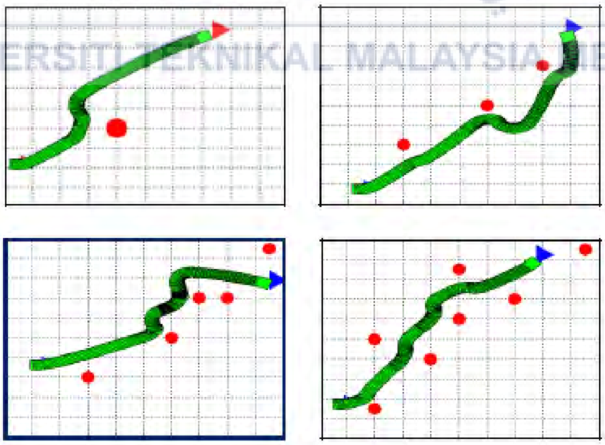


Figure 2.8: Navigation of mobile robot in [34]

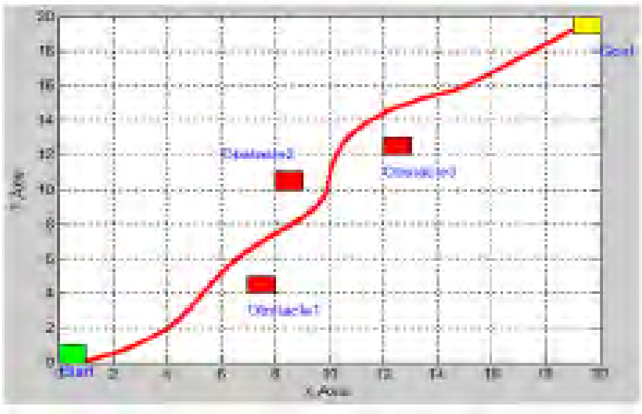


Figure 2.9: Navigation of mobile robot in [35]

To avoid the derivation of kinematic model of the controller, in [36] – [39], authors have developed FLC only for obstacle avoidance behaviour and achieve target seeking with other techniques. In all the papers, the authors have developed a FLC to get input from sensors and use the data provided by sensor to control the left and right linear velocity of the robot to move the robot in a safest path. In [36] author have developed Sugeno FLC which takes input from three unknown sensors where three membership functions for each input to control the linear velocity of the robots based on 27 fuzzy rules defined by authors. In [37] – [39] authors have developed Mamdani FLC which takes input from 8 infrared sensors to control the linear velocity of left and right velocity of robot by using E-Puck mobile robots in Webot robotic simulator. In [37] author have tested the developed FLC with 256 rules in three different environments, simple environment with few obstacles, average environment with medium obstacles and complex environment with plenty obstacles. The E-Puck robot able to avoid obstacles in all the environments and reach the target using ground sensor but the time taken of the mobile robot increases with the number of obstacles. In [38], the similar testing is done with just 9 rules FLS which produce the same results as in the previous paper. The table of the time taken for E-Puck robot for [37] and [38] is shown in Table 2.3 and Table 2.4 respectively. However, in [39], the author done both simulation and hardware testing of developed FLC by using line following technique as in Figure 2.10. In this paper, the robot follows the line as it avoids the static and dynamic obstacles, other e-puck robots unlike in previous papers where the only static obstacles are considered and there is no fixed path for the robot.

Table 2.3: The results of E-puck Robot through all environment [37]

	Number of trials	Time	Behavior
Simple environment	10	1m 22.6 s	Successfully avoided all solid objects
Average environment	10	1m 23.3s	Successfully avoided all solid objects
Complex Environment	10	2m 16.7s	Successfully avoided all solid objects

Table 2.4: The results of E-puck Robot through all environment [38]

PERFORMANCE	SIMPLE	AVERAGE	Complex
Distance	1.317m	1.326m	1.584
Time	30.112s	40.128s	1.15.216 min
Reach Foal	Success	Success	Success

Figure 2.10: Navigation of mobile robot with static & dynamic obstacles [39]

Since there are two type of FLS available, the difficulties in choosing the best controller for navigation of mobile robot is always been an issue for the researchers. In [40]- [41] -, the authors have done comparative comparison between the Mamdani and Sugeno FLC to contribute in that area of research. In [41], author has designed a prototype of a rectangular shaped differentially steered mobile robot equipped with two SRF05 ultrasonic sensors to sense the environment with a microcontroller board incorporating AT89C52 microcontrollers to run Fuzzy controllers. In this experiment, the authors have tested the smoothness of path generated, RAM used and time taken by Mamdani FLC and Sugeno FLC. The test resuts shows that, Mamdani's motion are more soother than Sugeno but the RAM used and time taken by Sugeno is more efficient than the Mamdani. In [40], the similar test is carried out by the author to compare the smoothness and time taken by the wall- following robot for both Mamdani and Sugeno. The author concludes that, Sugeno able to create more smoother path and lesser time compared to Mamdani due to the computation efficiency of Sugeno.

2.5 Summary

From the previos studies [34]- [40], we can conclude that FLS is great tool to navigate mobile robots in known or unknown environment without colliding with any static or dynamic obstacle. However, the type of FLS and number of fuzzy rules are different among the authors due to the expertise of the researchers, number of parameters used in the experiment and membership function considered or each parameter. The more the input, output and membership functions the more the fuzzy rules should be applied to guide the robot in a safest path.

The FLS can be designed for both obstacle avoidance behaviour and target seeking behavior as in [34]- [35], but the kinematic of the robot should be studied and mathematical expression should be derived to find the range of parameters to control the angular velocity of mobile robot to achieve the target seeking behavior of mobile robot. To avoid such a problem, in [36] – [39] authors have designed FLC only for the obstacle avoidance behaviour by controlling linear velocity of wheels and achieve the target seeking behaviour with various of other techniques such as using vision or ground sensors and formulas.

Moreover, for better input data acquisition, the sensors should be more responsive and has ability to detect short range obstacles. Because the sensor which unable to detect short range obstacles will response too soon to the obstacles and too late which may cause collision with the obstacles. Proximity infrared sensors offers greater flexibility in detecting short range obstacles as in [37] – [39].

Furthermore, the type of FLC in developing obstacle avoidance behaviour of mobile robot is crucial task that needed extra attention. This is because the controller functions as the brain of the system to process the input and give responsive output to move the robot in a safe path. The two-common type of fuzzy method are Mamdani and Sugeno. Both method offers a reliable obstacle avoidance capability. However, the time taken by robot and path generated by the robot is differs due to the computational efficiency of each controller. In [40] and [41], Sugeno offers faster response time than the Mamdani due to computational efficiency of Sugeno which produce constant output. In [41] the path generated by Mamdani is smoother than Sugeno whereas, in [40], the path generated by Sugeno is smoother than the Mamdani. Since there are not many researches on the smoothness of path, more focus should be given on the comparison of path generated by Mamdani FLC and Sugeno FLC to validate the results obtained in [40] and [41].

So, this project emphasis on the development and comparison of Mamdani FLC and Sugeno FLC which takes input from proximity sensors to control the linear velocity of left and right wheel. The target seeking behaviour of robot will be achieved using the distance and angle formula.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the method and tools used to design integration of FLS for obstacle avoidance behaviour of mobile robot is explained in detail.

3.2 Process Flow Chart

Figure 3.1 disclose the overall process flow chart of the implementation of FLS for obstacle avoidance behaviour of mobile robot. As the initial step, the designation of the robot is carried out to specify the input range and output range for the controller. Next, Mamdani FLC and Sugeno FLC is developed based on the input range and output range of the robot. For the comparison of the controller, the robot should have a specific target so that the path generated by both controller can be compared. So, target seeking behaviour is developed based on the distance and angle of robot from the target. Next, different environment is created using robotic simulator to implement the controllers in robot and observe the path generated by robots for validation process. Since, the FLC and robotic environments are created in different softwares, application programming interphase (API) configuration is done to integrate the both softwares. Once the api configuration is success, the validation of controllers using robotic simulator is carried out. Upon the successful implementation of the controller, the results generated by both controller is compared and analysed.

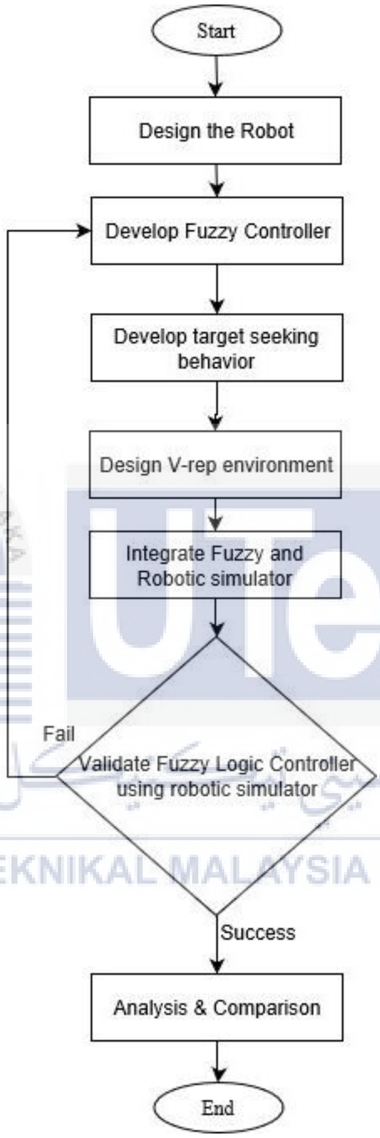


Figure 3.1: Overall process flow chart

3.2.1 Block Diagram of System

Figure 3.2 shows the block diagram of the integration of FLS for obstacle avoidance behavior of mobile robot. The main role of the robot is to move from the initial position to the target location in collision-free path. Mobile robot will move towards to the target location by using goal seeeking algorithm. At the same time, if there is presence of obtacles detected by mobile robot proximity sensors, FLCs use value measured by sensors as input to produce right and left velocity to navigate the robot to avoid obstacles.

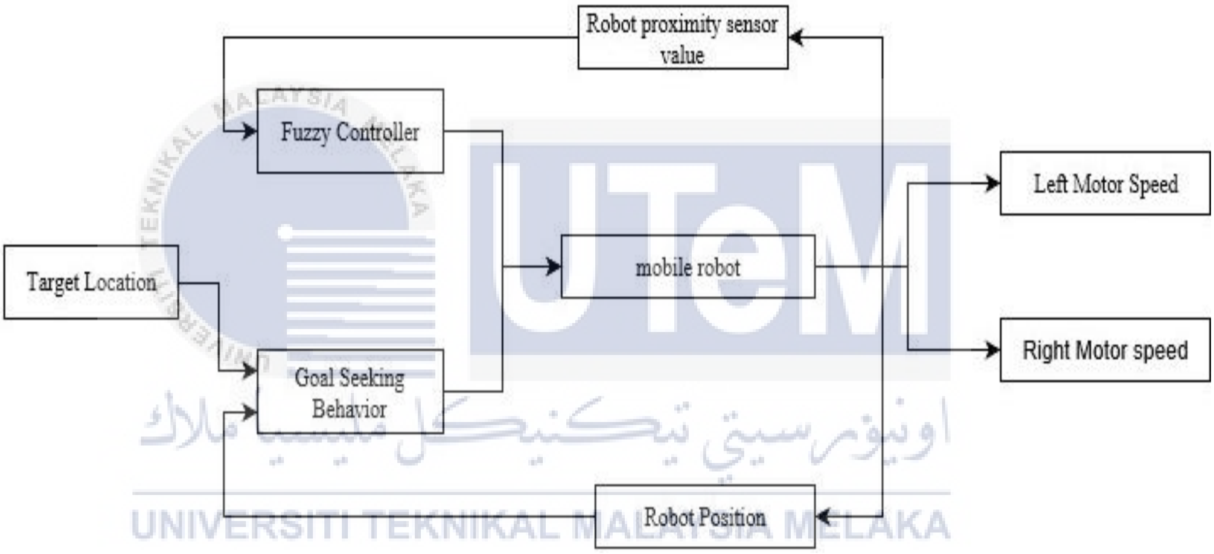


Figure 3.2: Block Diagram of The Integration of FLS for Obstacle Avoidance Behaviour of Mobile Robot

3.4 Designation of the Robot

An input and output of the controller as well as range of each input and output is essential for developing a robust FLC. In this project, both Mamdani and Sugeno FLC are developed based on Pioneer P3-DX robot as shown in Figure 3.3. Pioneer P3-DX is a small lightweight differential drive robot equipped with two wheels where each wheel controlled by a motor. The robot comes with 16 proximity sensors, one battery, wheel encoders, a microcontroller with ARCOS firmware, and the Pioneer SDK advanced mobile robotics software development package [42]. Their versality, reliability and durabiliity made them one of the most common platformfor adavnaced intellligent robotics.

Figure 3.3: Pioneer P3-DX robot [42]

The designation of the robot is completely done using Virtual Robotic Experimentation Platform or simply known as V-REP robotic simulator. Five proximity sensors equipped in the Pioneer robot, S1, S2, S3, S4 and S5 are used to detect the distance between the robot and obstacles. The angle between the each proximity ray is 45 degree and the range of proximity sensor detection are 1m. The value of this five sensors are used as input. The output of the robot is the linear left and right wheel velocity of the robot. The input and output are depicted as in Figure 3.4. The maximum velocity of the wheel is 1m/s and minimum wheel velocity is -0.5m/s. The Table 3.1 shows the rotation of the wheel direction based on the wheel velocity.

Table 3.1: Wheel direction based on wheel velocity

Wheel Velocity	Wheel Direction
Positive velocity	Forward
Zero Velocity	Stop
Negative Velocity	Reverse

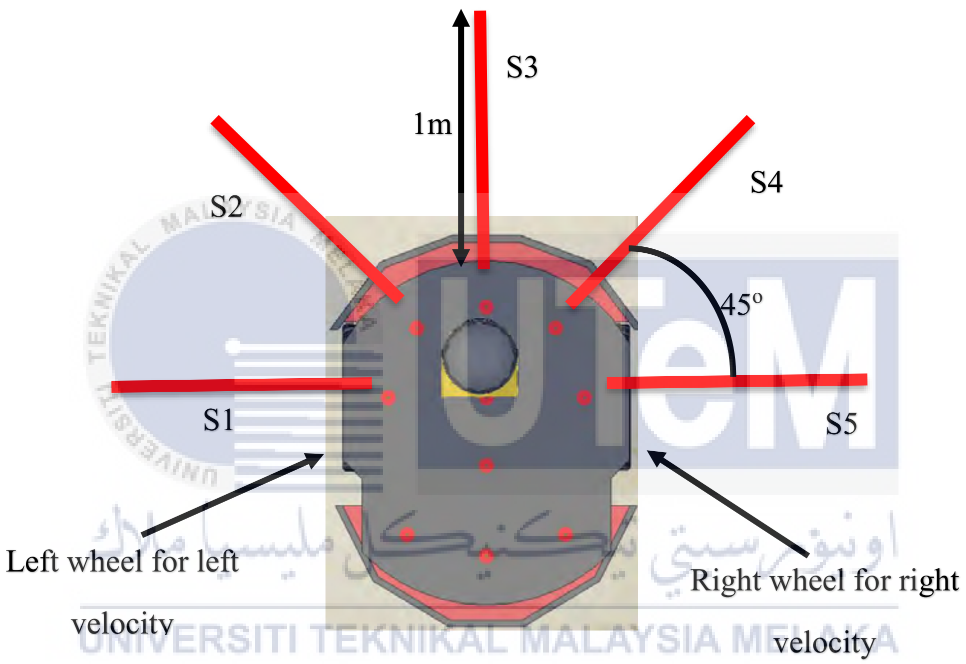


Figure 3.4: Input and output of FLC based on the Pioneer P3-DX robot [42]

This robot is then equipped with Sick 300 Safety Laser Sensors as in Figure 3.5 to get the instanteneos position and orientation of the robot which will be used to build the target seeking behavior.

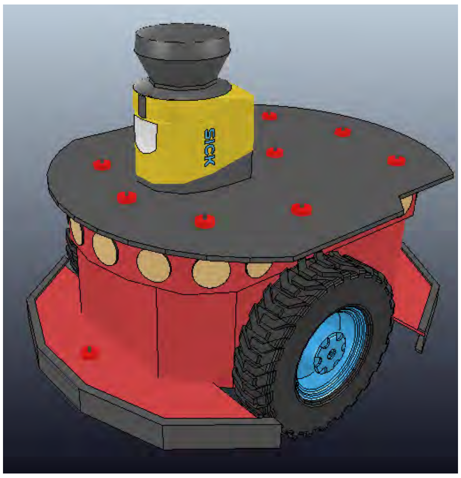


Figure 3.5: Sick 300 safety laser sensors Pioneer P3-DX robot

3.4 Development of FLC

Two FLCs has been designed for this project, Mamdani FLC and Sugeno FLC to navigate the robot from initial position to goal position without colliding with any obstacles. The input of both controller is the distance between the obstacles and robot measured by five proximity sensors, while the output of the controllers is left and right wheel velocity of Pioneer P3-DX. Mamdani and Sugeno were developed using Fuzzy Toolbox in Matlab 2016b.

3.4.1 Designing Input Membership Function

Five input variables, namel 'S1', 'S2', 'S3', 'S4' and 'S5' were considered for both controllers. Figure 3.6 shows the input variable of Mamdani FLC which is similar to the Sugeno FLC. Distance information from the five sensors is described with the help of two fuzzy sets: Detect and Nodetect. The membership functions for each sensors, are shown in Figure 3.7 and are described by the following expressions:

$$\text{Si, where i= 1,2,3,4,5} \tag{6}$$

$$\mu_{\text{Si, Detect}} = \begin{cases} 1\,, & 0 \leq Si \leq 0.5 \\ \frac{0.6-Si}{0.1}\,, & 0.5 \leq Si \leq 0.6 \\ 1, & 0.6 \leq Si \end{cases} \tag{7}$$

$$\mu_{\text{Si, Nodetect}} = \begin{cases} 0\,, & Si \leq 0.5 \\ \frac{Si-0.6}{0.1}\,, & 0.5 \leq Si \leq 0.6 \\ 1, & 0.6 \leq Si \leq 1 \end{cases} \tag{8}$$

The membership function parameter for input variables are shown in Table 3.2.

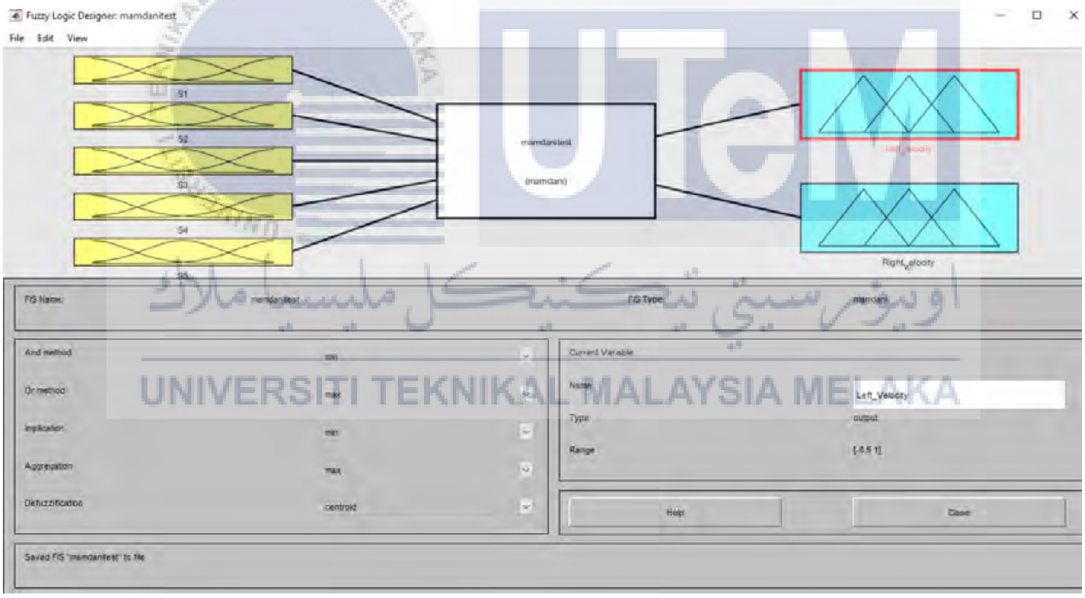


Figure 3.6 :Five input for Mamdani FLC and Sugeno FLC

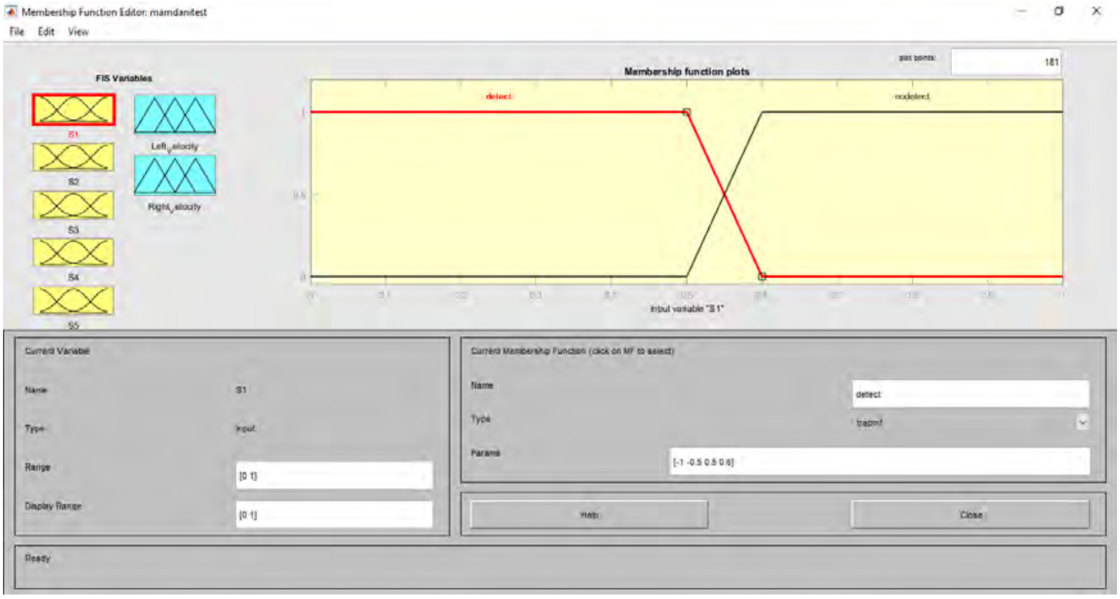


Figure 3.7: Two trapezium membership function for input of Mamdani FLC and Sugeno FLC

Table 3.2: Membership function parameters for input variables of Mamdani FLC and Sugeno FLC

Input Variables	Membership Functions Parameter	
	Detect (m)	Nodetect(m)
S1	[0 0 0.5 0.6]	[0.5 0.6 1 1]
S2	[0 0 0.5 0.6]	[0.5 0.6 1 1]
S3	[0 0 0.5 0.6]	[0.5 0.6 1 1]
S4	[0 0 0.5 0.6]	[0.5 0.6 1 1]
S5	[0 0 0.5 0.6]	[0.5 0.6 1 1]

3.4.2 Designing Output Membership Function

3.4.2.1 Output of Mamdani FLC

Two output variables, 'Left Velocity' and 'Right Velocity'were considered for Mamdani FLC as in Figure 3.6. The output of Mamdani FLC is described with the help of four fuzzy sets: 'Reverse', 'Stop', 'Medium' and 'Fast'. The membership functions for each velocity, are shown in Figure 3.8 and are described by the expressions:

$$n\_Velocity, \text{ where } n = Left/Right \tag{9}$$

$$\mu_{n\_Velocity,\ Reverse} = \begin{cases} 0, & n\_Velocity \leq -0.5 \\ \frac{0 - n\_Velocity}{0.5}, & 0.5 \leq n\_Velocity \leq 0 \\ 0, & 0 \leq n\_Velocity \end{cases} \tag{10}$$

$$\mu_{n\_Velocity,\ Stop} = \begin{cases} 0, & n\_Velocity \leq -0.5 \\ \frac{n_{Velocity}+0.5}{0.5}, & -0.5 \leq n\_Velocity \leq 0 \\ \frac{0.5 - n\_Velocity}{0.5}, & 0 \leq n\_Velocity \leq 0.5 \\ 0, & 0.5 \leq n\_Velocity \end{cases} \tag{11}$$

$$\mu_{n\_Velocity,\ Medium} = \begin{cases} 0, & n\_Velocity \leq 0 \\ \frac{n_{Velocity}-0.5}{0.5}, & 0 \leq n\_Velocity \leq 0.5 \\ \frac{1 - n_{Velocity},}{0.5} & 0.5 \leq n\_Velocity \leq 1 \\ 0, & 1 \leq n\_Velocity \end{cases} \tag{12}$$

$$\mu_{n\_Velocity,\ Fast} = \begin{cases} 0, & n\_Velocity \leq 0.5 \\ \frac{n\_Velocity - 0.5}{0.5}, & 0.5 \leq n\_Velocity \leq 1 \\ 0, & 1 \leq n\_Velocity \end{cases} \tag{13}$$

The membership function parameter for input variables are shown in Table 3.3.

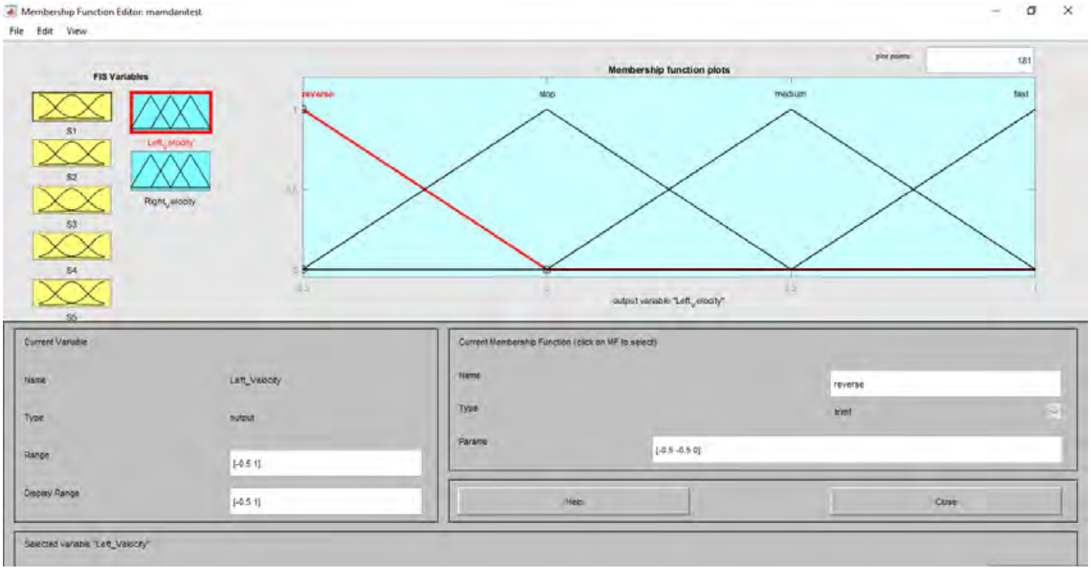


Figure 3.8: Five triangular membership function for output of Mamdani FLC

Table 3.3: Membership function parameters for output variables of Mamdani FLC

OutputVariables	Membership Functions Parameter			
	Reverse (m/s)	Stop(m/s)	Medium(m/s)	Fast(m/s)
Left_Velocity	[-0.5 -0.5 0]	[-0.5 0 0.5]	[0 0.5 1]	[0.5 1 1]
Right _Velocity	[-0.5 -0.5 0]	[-0.5 0 0.5]	[0 0.5 1]	[0.5 1 1]

3.4.2.2 Output of Sugeno FLC

Two output variables, ‘Left Velocity’ and ‘Right Velocity’were considered for Sugeno FLC as in Figure 3.9. The output of Sugeno FLC is described with the help of four fuzzy sets: ‘Reverse’, ‘Stop’, ‘Medium’ and ‘Fast’. The membership functions for each velocity, are constant as shown in Figure 3.9. The membership function parameter for output variables are shown in Table 3.4.

Figure 3.9: Five membership function for output of Sugeno FLC

Table 3.4: Membership function parameters for output variables of Sugeno FLC

OutputVariables	Membership Functions Parameter			
	Reverse (m/s)	Stop(m/s)	Medium(m/s)	Fast(m/s)
Left_Velocity	-0.5	0	0.5	1
Right _Velocity	-0.5	0	0.5	1

3.4.3 Constructing the Rule Base for Mamdani and Sugeno FLC

After the constructing membership functions for inpu variables, a total of 15 rules is applied to the controller, so that the robot able to avoid obstacles as shown in the Table 3.5. These rules were applied in a step-wise manner with a careful observation of the robot's reaction in V-REP environments before proceeding to the next rule. The inference output from these rules is computed by Mamdani (max-min) operator for composition, minimum operation for implication, and center of area for defuzzification as shown in Figure 3.10 . Since each rule has a crisp output for Sugeno, the overall output is obtained via weighted average as shown in Figure 3.11, thus avoiding the time-consuming process of defuzzification required in a Mamdani model.

Table 3.5: User Defined Fuzzy Rules for Mamdani and Sugeno FLC

	Input Variables					**Output Variables**	
Rule Number	**S1**	**S2**	**S3**	**S4**	**S5**	**Left wheel velocity**	**Right wheel velocity**
1	Detect	No detect	No detect	No detect	No detect	Medium	Stop
2	No detect	Detect	No detect	No detect	No detect	Medium	Stop
3	No detect	No detect	No detect	Detect	No detect	Stop	Medium
4	No detect	No detect	No detect	No detect	Detect	Stop	Medium
5	No detect	Detect	Detect	Detect	Detect	Reverse	Fast
6	Detect	Detect	Detect	Detect	No detect	Fast	Reverse
7	No detect	No detect	No detect	No detect	No detect	Fast	Fast
8	Detect	Detect	No detect	No detect	No detect	Medium	Reverse
9	No detect	Detect	Detect	No detect	No detect	Fast	Reverse
10	No detect	No detect	Detect	Detect	No detect	Reverse	Fast
11	No detect	No detect	No detect	Detect	Detect	Stop	fast
12	No detect	No detect	Detect	Detect	Detect	Reverse	fast
13	Detect	Detect	Detect	No detect	No detect	Fast	Reverse
14	No detect	No detect	Detect	No detect	No detect	Fast	Reverse
15	No detect	Detect	Detect	Detect	No detect	Fast	Reverse

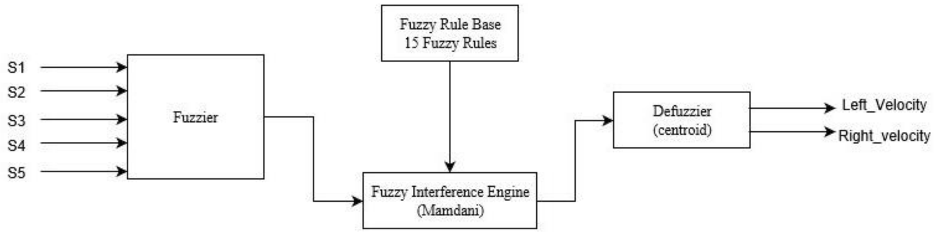


Figure 3.10: Block diagram of Mamdani FLC

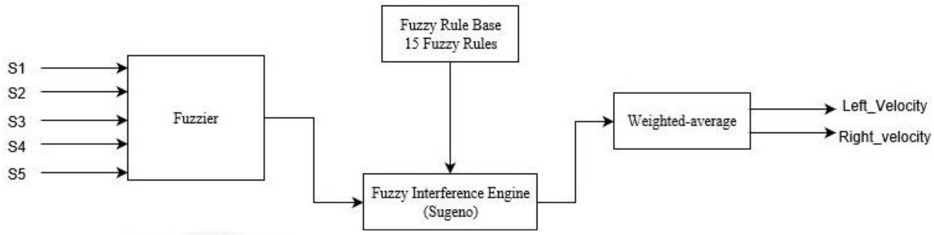


Figure 3.11: Block diagram of Sugeno FLC

3.5.1 Target Seeking Behaviour

Target seeking behaviour of Pioneer P3-DX robot is achieved using Sick 300 Safety Laser Sensors equipped on the robot. This sensor read the instanteneous coordinate in term of [x,y] of the robot and orientation of the robot relative to world coordinate and send the data to Matlab using remote API configration. The Figure 3.12 shows the he angle between robot and x axis theta (θ), the angle between robot and target is beta (β), initial position of the robot and target position of the robot.

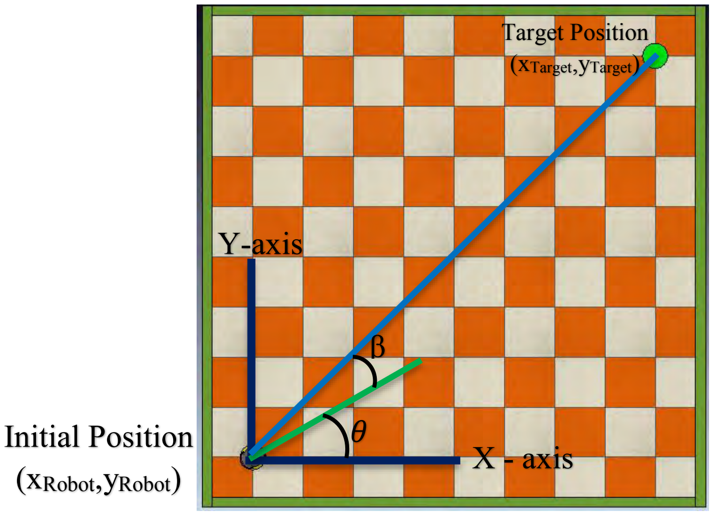


Figure 3.12: The position and orientation of the robot reative to parent frame

First step, the robot can adjust its motion direction and moves towards the target since it know current position and find the target position. The distance (d) between robot and its destination is calculated using the formula as shown below.

$$d = \sqrt{(y_{Target} - y_{Robot})^2 + (x_{Target} - x_{Robot})^2} \quad (13)$$

The robot also need adjust its orientation when it reaches the goal point. Assume that the angle between robot and x_{Robot} is theta (θ); the angle between robot and target is beta (β) in x direction so that the angle must be changed is gamma (γ)

$$\beta = arctan\frac{y_{Target}-y_{Robot}}{x_{Target}-x_{Robot}} \quad (14)$$

$$\gamma = \beta - \theta \quad (15)$$

3.6 Designing the V-REP Environment for validation of FLCs.

The Virtual Robot Experimentation Platform is a robotic simulator used for simulations of developed algorithm, fast prototyping, robotics related education, remote monitoring and safety monitoring. This simulator comes with integrated development environment, based on a distributed control architecture where each object/model can be individually controlled via an embedded script, a plugin, a ROS node, a remote API client, or a custom solution. This makes V-REP very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab or Octave. Square shape resizable floor is created measuring 10m in width and 10m in length as shown in Figure 3.13. To avoid the robot exits from the testing environment, four cuboid shaped wall is designed and placed at the side of each edge of floor. Next, cylindrical with diameter of obstacle are designed which Pioneer P3-DX should avoid to reach the target as in Figure 3.14.

.

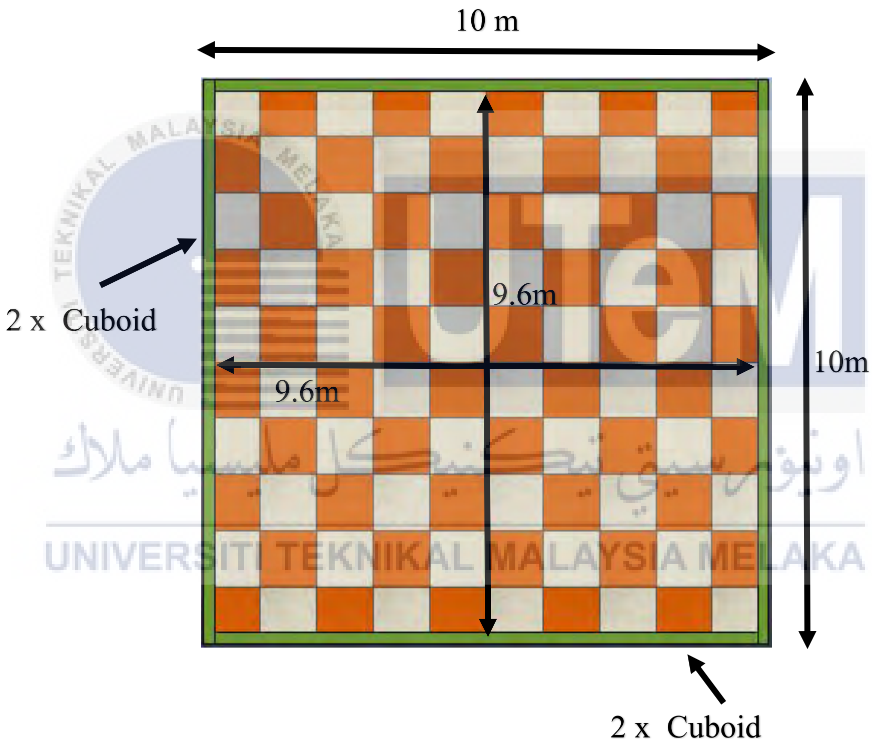


Figure 3.13: Measurement of the testing environment

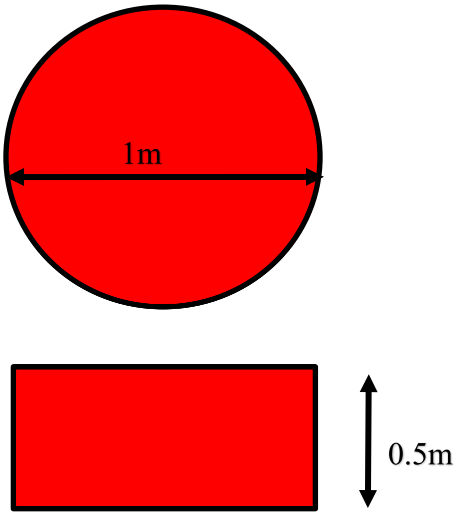
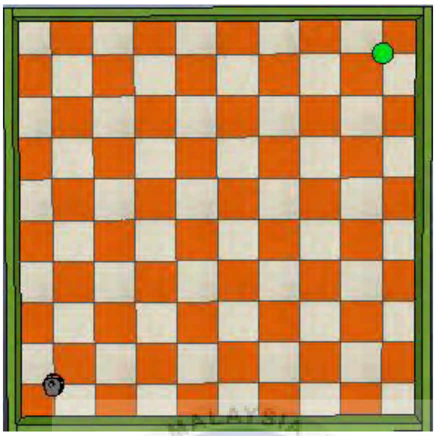
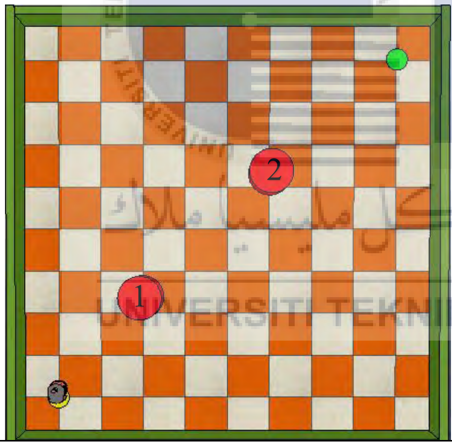
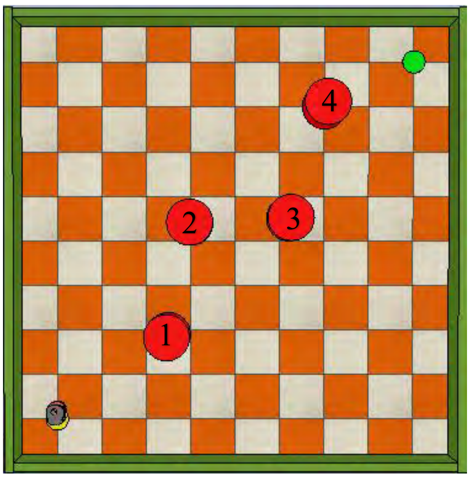


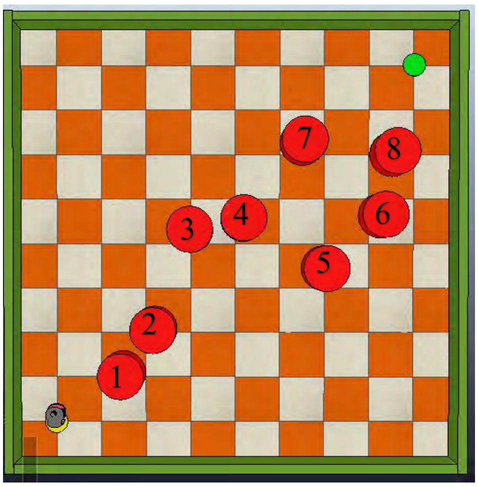
Figure 3.14: Measurement of the cylindrical obstacles

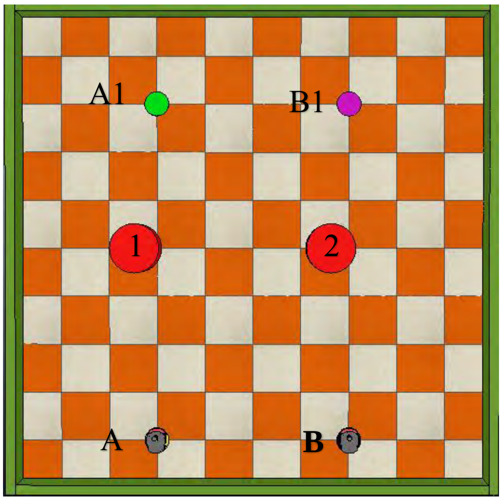
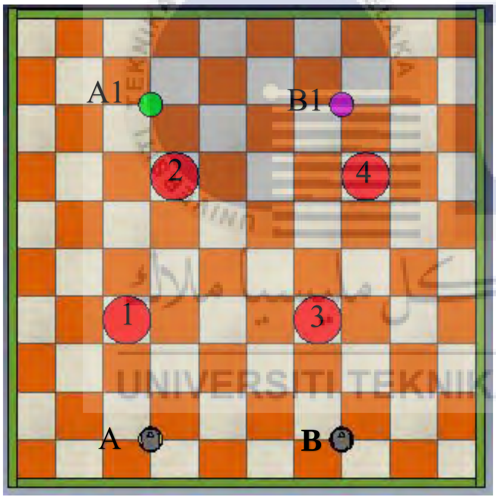
3.6.1 V-REP Environments

Four scenes are designed to validate the FLCs for obstacle avoidance behavior with target seeking using single robot and three scenes for Multirobot environment. In each environment, the starting position and target location of the robot is fixed. Only the number of obstacles in each scenes are changed. The coordinate of initial position of robot, target position and obstacle position are illustrated in the Table 3.6.

Table 3.6 : The coordinate of initial position of robot, target position, number of obstacles and position of obstacles

Scenes		
A. 	Initial Position	[0.0 , 0.0]
	Target Position	[8.0 , 8.0]
	Number of Robot	1
	Number of Obstacles	0
	Position of obstacles	-
B. 	Initial Position	[0.0 , 0.0]
	Target Position	[8.0 , 8.0]
	Number of Robot	1
	Number of Obstacles	2
	Position of obstacles	1. [2.0 , 2.4] 2. [5.0 , 5.3]
C. 	Initial Position	[0.0 , 0.0]
	Target Position	[8.0 , 8.0]
	Number of Robot	1
	Number of Obstacles	4
	Position of obstacles	1. [2.5 , 1.9] 2. [3.0 , 4.4] 3. [5.2 , 4.5] 4. [6.0, 7.0]

Scenes		
D. 	Initial Position	[0.0 , 0.0]
	Target Position	[8.0 , 8.0]
	Number of Robot	1
	Number of Obstacles	8
	Position of obstacles	1. [1 .5 , 1.1] 2. [2.2 , 2.1] 3. [3.0 , 4.3] 4. [4.2 , 4.5] 5. [6.1 , 3.5] 6. [7.3 , 4.6] 7. [5.5 , 6.3] 8. [7.5 , 6.0]
E. 	Initial Position	A. [2.0 , 0.0] B. [6.0 , 0.0]
	Target Position	A1. [2.0 , 7.0] B1. [6.0 , 0.0]
	Number of Robot	2
	Number of Obstacles	0
	Position of obstacles	-

Scenes		
F.  	Initial Position	A. [2.0 , 0.0] B. [6.0 , 0.0]
	Target Position	A1. [2.0 , 7.0] B1. [6.0 , 0.0]
	Number of Robot	2
	Number of Obstacles	1 for each robot
	Obstacles Position	1. [1 .6 , 4.0] 2. [5.6 , 4.0]
C.  	Initial Position	A. [2.0 , 0.0] B. [6.0 , 0.0]
	Target Position	A1. [2.0 , 7.0] B1. [6.0 , 0.0]
	Number of Robot	2
	Number of Obstacles	2 for each robot
	Obstacles Position	1. [1.5 , 2.5] 2. [2.5 , 5.5] 3. [5.5 , 2.5] 4. [6.5 , 5.5]

3.7 Integration of Matlab and V-REP

The integration of Matlab and V-REP done using Remote Application Program Interface Configuration. The procedure of integration using remote API configuration can be categorized into two phase, preparing V-REP file for integration and preparing Matlab file for integration.

A. Preparing V-REP file for integration

1. An associated child script is added to the scene as in Figure 3.15.

Figure 3.15: Child script in the V-REP

2. "simExtRemoteApiStart(19999)" line is added into the child script after the line "simSetThread SwitchTiming(2)". This step specifies the port on which the remote API server will run.
3. The simulation is started and stopped for few seconds.
4. All the files into the folder "C:\Program Files (x86)\V-REP3\V-REP_PRO_EDU\programming\remoteApi Bindings\ matlab\matlab" is copied to the Matlab working folder where the created '.fis' files and '.m' files are located.
5. File named "remoteApi.dll" at "C:\Program Files (x86)\V-REP3\V-REP_PRO_EDU\programming\remoteApi Bindings\lib\lib\ 64Bit" also copied into the same Matlab folder.

B. Preparing Matlab file for integration

Few important coding must be added in the Matlab script file to integrate the Matlab and V-REP. The coding and its respective functions are shown in Table 3.7.

Table 3.7: The coding that need to added in Matlab script to integrate with V-REP and its respective functions

Coding	Functions
vrep = remApi('remoteApi');	To access V-REP remote configuration
vrep.simxFinish(-1)	To close all exiting opened connections to ensure only one connectionis connected at a time
clientID = vrep.simxStart('127.0.0.1',19999,true,true,5000,5);	To start connection to V-REP with IP address and port (these are default values)
[errCode, sensorHandle] = vrep.simxGetObjectHandle(clientID, strcat('Pioneer_p3dx_ultrasonicSensor1'),vrep.simx _opmode_bloc king);	To create handles for the access of the proximity sensors of robot.
[errorCode,detectionState,detectedPoint,detectedObjectHandle,detectedSurfaceNormalVector] = vrep.simxReadProximitySensor(clientID, sensorHandle, vrep.simx_opmode_oneshot_wait);	To initialize the sensors
[errCode0, motorLeft] = vrep.simxGetObjectHandle(clientID, 'Pioneer_p3dx_leftMotor', vrep.simx_opmode_oneshot_wait);	To create the motor handles to get access to the wheels
[~,~] = vrep.simxGetStringSignal(clientID,'measuredDataAtThisTime',vrep.simx_opmode_streaming);	To initialize the laser sensor

[~, pioneer_h] = vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx' ,vrep.simx_opmode_oneshot_wait);	To create handle to receive the laser sensor data
[~, position] = vrep.simxGetObjectPosition(clientID, pioneer_h, -1, vrep.simx_opmode_streaming);	To initialize laser sensor to get the position of robot
[~, orientation] = vrep.simxGetObjectOrientation(clientID, pioneer_h, -1, vrep.simx_opmode_streaming);	To initialize laser sensor to get the orientation of robot
[~, position] = vrep.simxGetObjectPosition(clientID, pioneer_h, -1, vrep.simx_opmode_buffer);	To read the position of robot
[~, orientation] = vrep.simxGetObjectOrientation(clientID, pioneer_h, -1, vrep.simx_opmode_buffer);	To read the orientation of robot
[errorCode,detectionState,detectedPoint,detectedObjectHandle,detectedSurfaceNormalVector] = vrep.simxReadProximitySensor(clientID, sensorHandle, vrep.simx_opmode_oneshot_wait);	To get the values of the sensors
leftVel = vrep.simxSetJointTargetVelocity(clientID, motorLeft, speed(1), vrep.simx_opmode_streaming);	To set the speed for motors with the values returned by fuzzy/user

vrep.simxFinish(clientID);	To close the connection to V-REP
vrep.delete()	To call the destructor to stop V-REP

3.8 The process flow chart of simulation

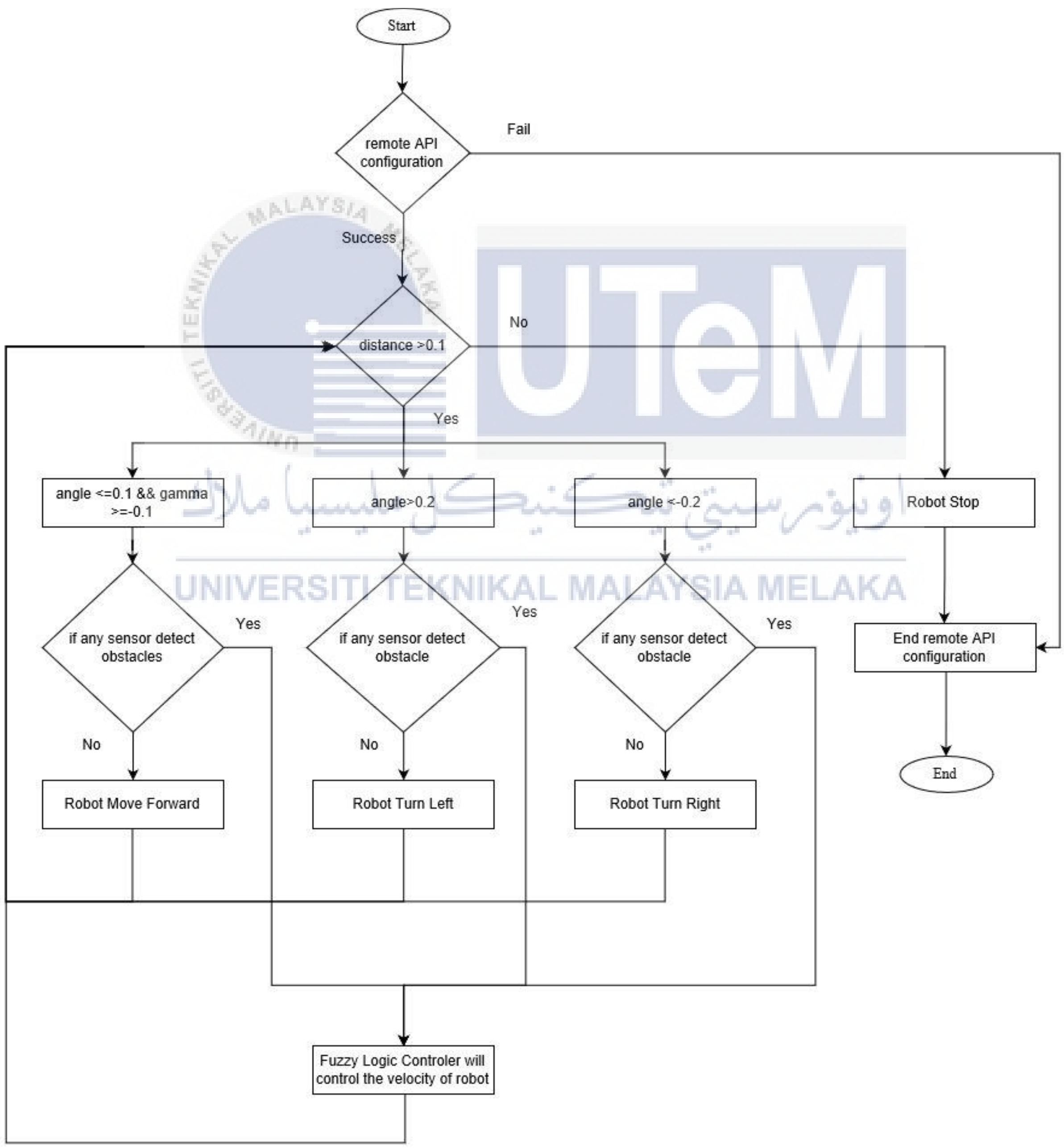


Figure 3.16: Flow chart of the simulation

Figure 3.16 shows the flow chart of simulation for integration of FLS for obstacle avoidance behaviour of mobile robot. The simulation begins upon the successful remote application program interface (API configuration) between V-REP and Matlab. If the distance from the target and robot position is more than 0.1m, the robot is far from the target location. So, the robot will start move towards the target with three condition. The second condition is, if the angle is bigger than 0.1 radian the robot will rotate left whereas the third condition is if the angle is smaller than 0.1 radian the robot will rotate right. While executing this three condition, if any of the proximity sensors of roots detects the obstacle, the FLC either Mamdani or Sugeno, will control the left velocity and right velocity of robot based on the 15 rules defined. This process will keep on repeating until the distance between the robot and target location is not more than 0.1m.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Overview

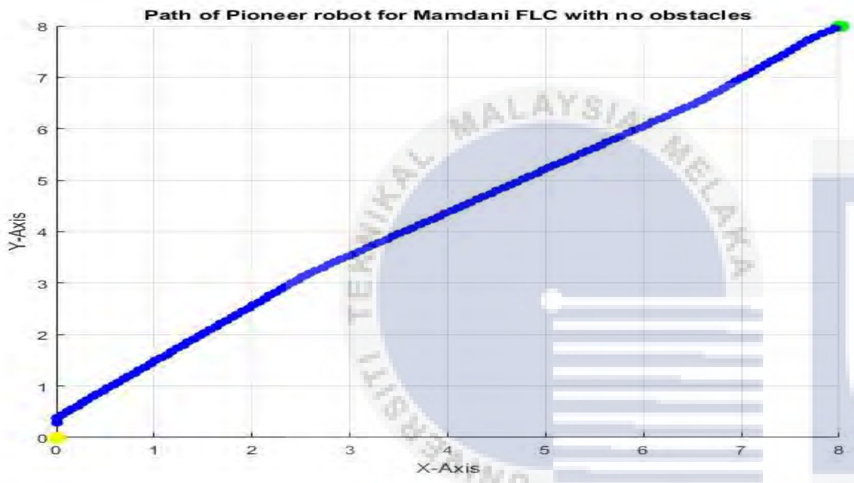
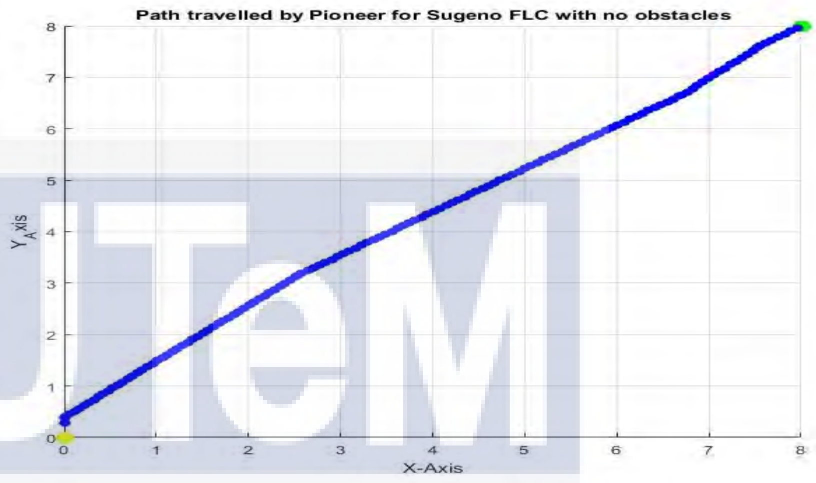
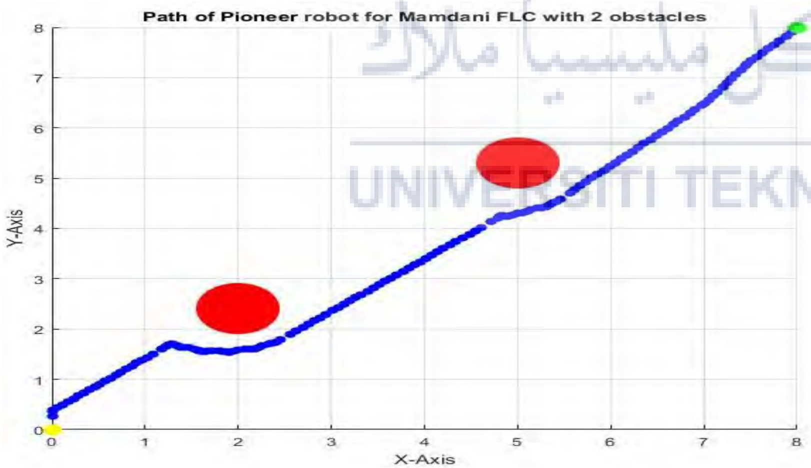
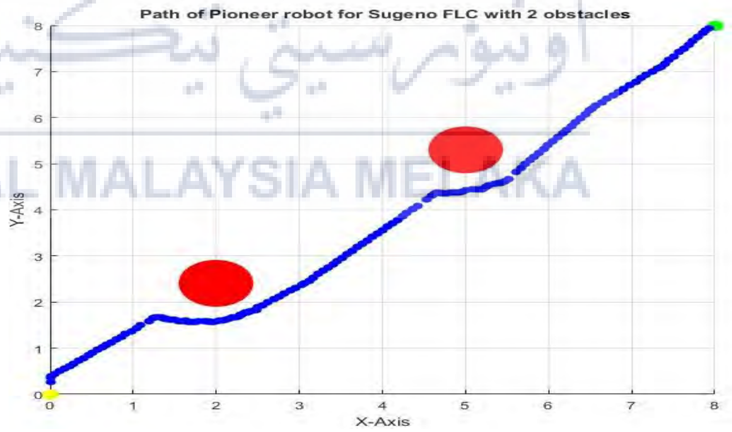
This chapter highlights the results obtained from software simulations together with its analysis and discussion. The data obtained from the simulation software is tabulated and presented in the graph. The detailed discussion is described based on how does the input data varies the output results.

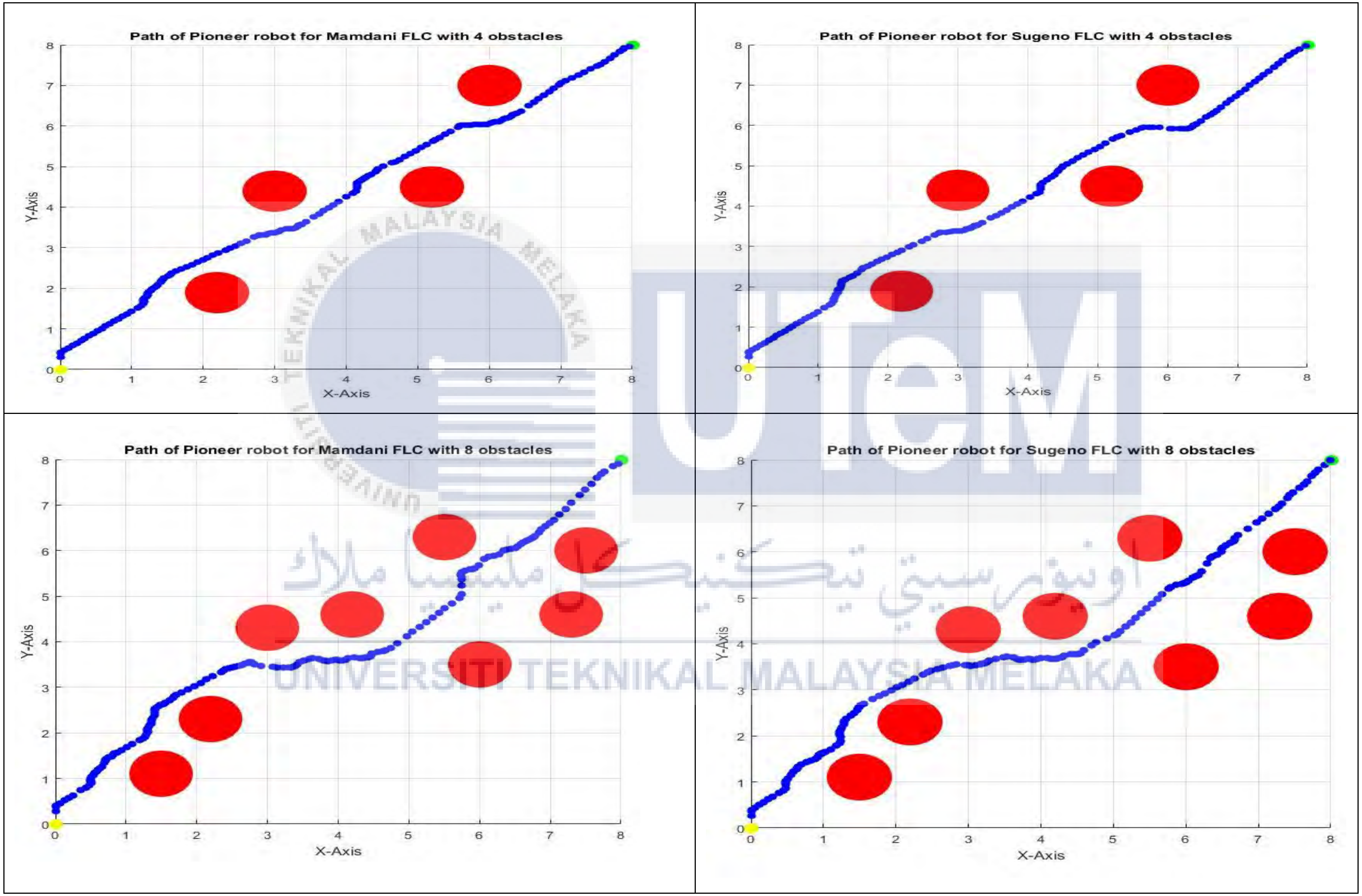
4.2 Results for Single Robot

4.2.1 Results for Path travelled by Robot

This project is developed to integrate FLS with mobile robot for obstacles avoidance behaviour. The simulation is carried out in four different environments for each controller Mamdani FLC and Sugeno FLC using single Pioneer P3-DX in each environment. The number of obstacles in each environment are given as 0 obstacle, 2 obstacles, 4 obstacles and 8 obstacles respectively. The Table 4.1 shows the scatter plot of path of Pioneer robot in each environment for the Mamdani FLC and Sugeno FLC.

Table 4.1: Path of Pioneer robots for the Mamdani FLC and Sugeno FLC in different environments.

Mamdani	Sugeno
	
	


Path of Pioneer robot for Mamdani FLC with 4 obstacles
Path of Pioneer robot for Sugeno FLC with 4 obstacles
Path of Pioneer robot for Mamdani FLC with 8 obstacles
Path of Pioneer robot for Sugeno FLC with 8 obstacles
X-Axis
Y-Axis
0
1
2
3
4
5
6
7
8

Based on Table 4.1, the 'yellow' circles, 'green' circles and 'red' circles indicates the starting positions, goal positions and obstacles positions, respectively in each diagram. The blue line indicates the scatter plot of the path travelled by robot in each environment. From Table 4.1, the Pioneer P3-DX successfully navigate itself from an initial position to goal position without colliding with any obstacles in any environment. This indicates that FLC able to integrate with proximity sensors to communicate with outside world by detecting obstacles and control the linear velocity of the left and right wheel to navigate robot in safe path based on 15 fuzzy rules defined by the user.

4.2.2 Results for smoothness of path travelled by Robot

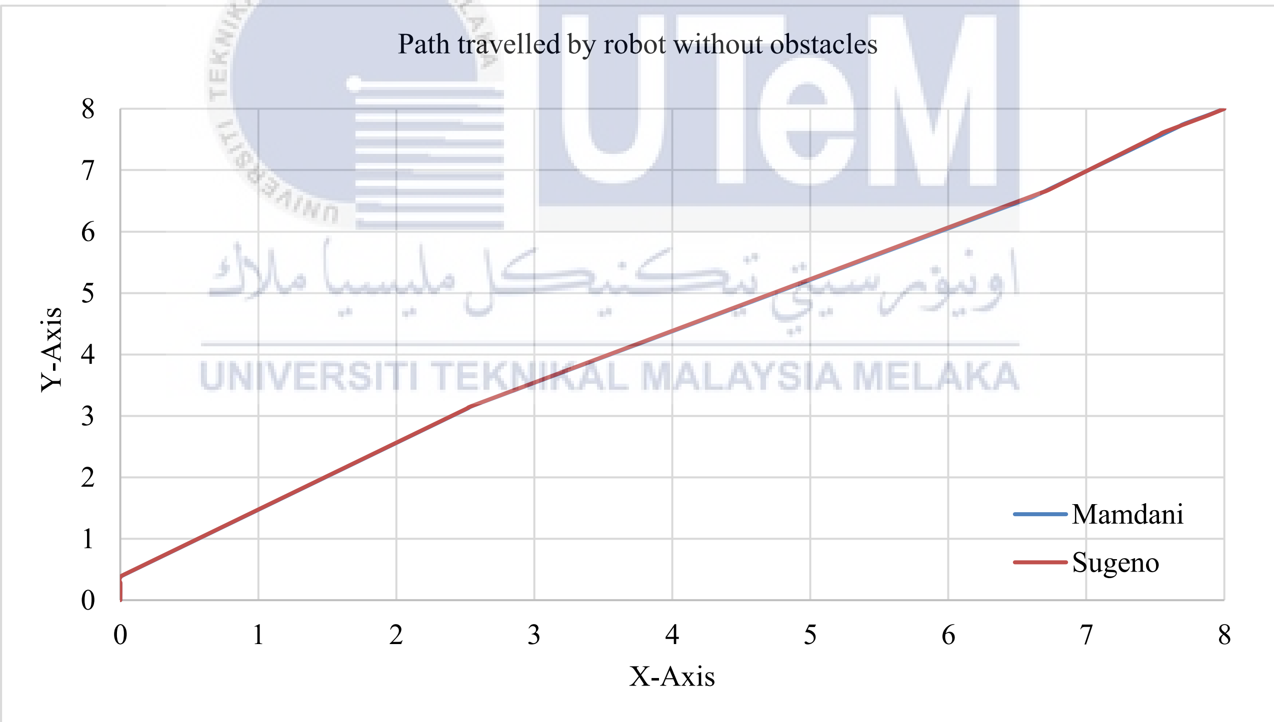


Figure 4.1: Path travelled by robot without obstacles

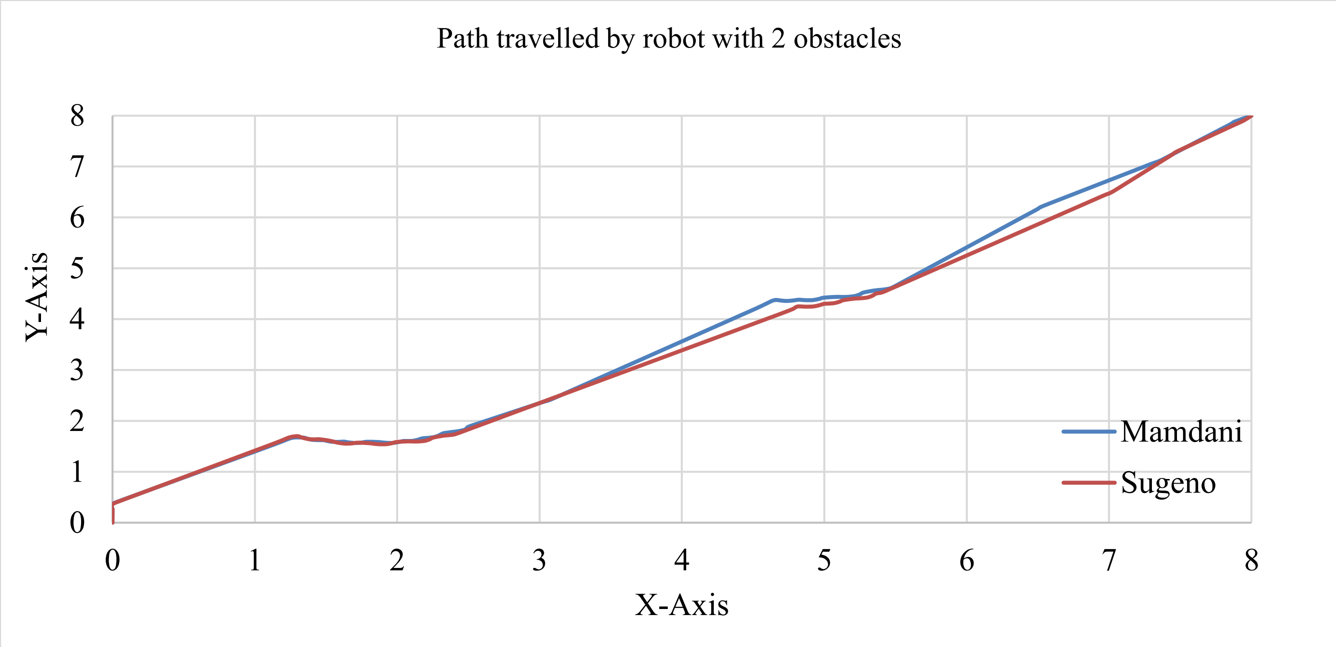


Figure 4.2: Path travelled by robot with 2 obstacles

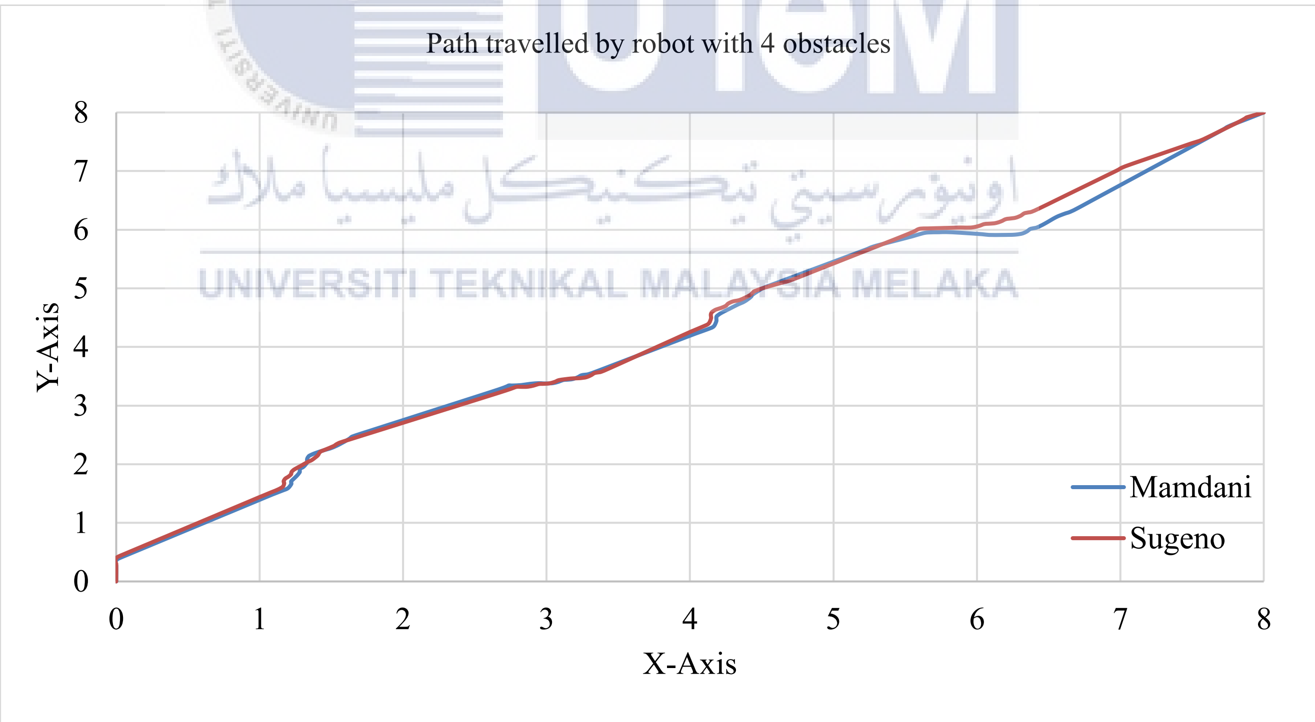


Figure 4.3: Path travelled by robot with 4 obstacles

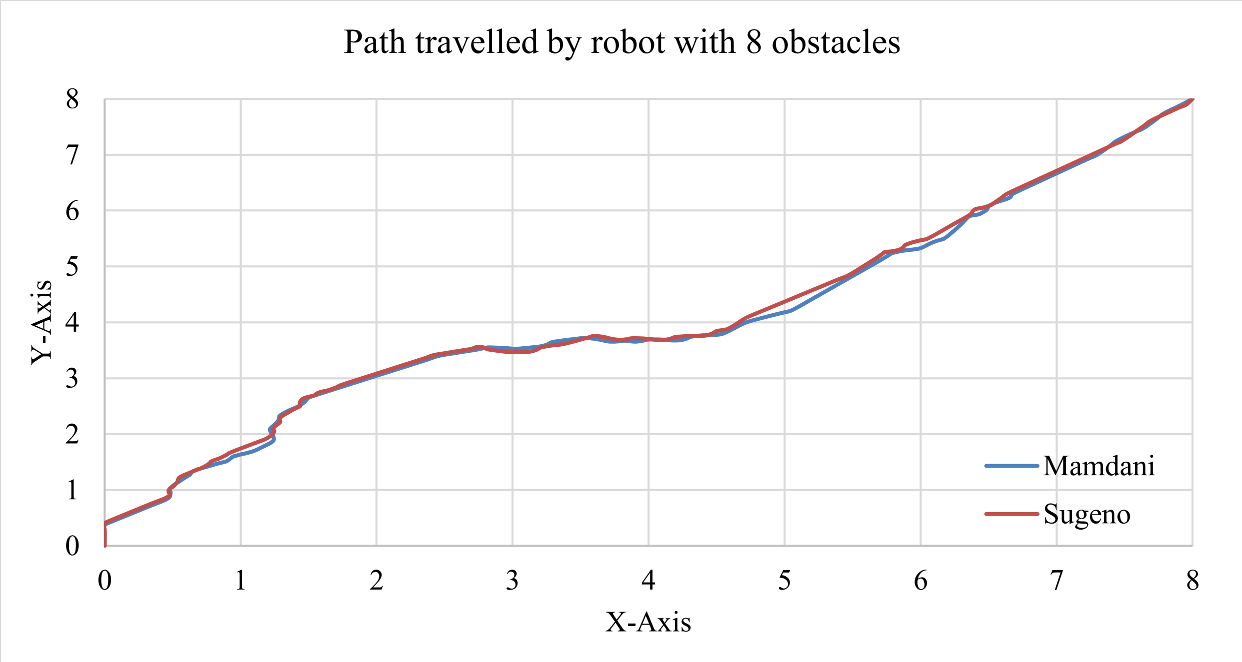


Figure 4.4: Path travelled by robot with 8 obstacles

Figure 4.1 to Figure 4.4 disclose the comparison of the path travelled by a robot from initial position to goal position in single robot environments by using Mamdani FLC and Sugeno FLC. Based on Figure 4.1, there are not significant difference in path travelled by Pioneer using the Mamdani FLC and the Sugeno FLC. Since, there is no obstacle in between the path of robot, FLC does not initialized in that environment. However, in Figure 4.2, Figure 4.3 and Figure 4.4, there are significant differences in the path travelled by the robot. This is because FLCs is initialized by proximity sensors once detects the obstacles. As we can see from the three figures, the path travelled by Pioneer robot using Sugeno FLC is more adequate than the path travelled by Mamdani FLC. This is due to the capability of the Sugeno FLC to produce constant output and faster response time unlike the Mamdani FLC which is only capable of producing crisp output after defuzzification process as the output of the Mamdani FLC is in Fuzzy sets.

4.2.3 Efficiency of the controller

The efficiency of the controller is tested in four different environments under different number of obstacles in each environment by using a single Pioneer P3-Dx robot. The initial distance of the robot from the goal position is set at 11.314 m. The distance between the robot and goal position decreases as the robot move towards the goal. The efficiency of the controller is described in term of time taken by the robot to reach the target. Figure 4.5 and Figure 4.6 shows the time taken by Pioneer robot to reach the target in four environments.

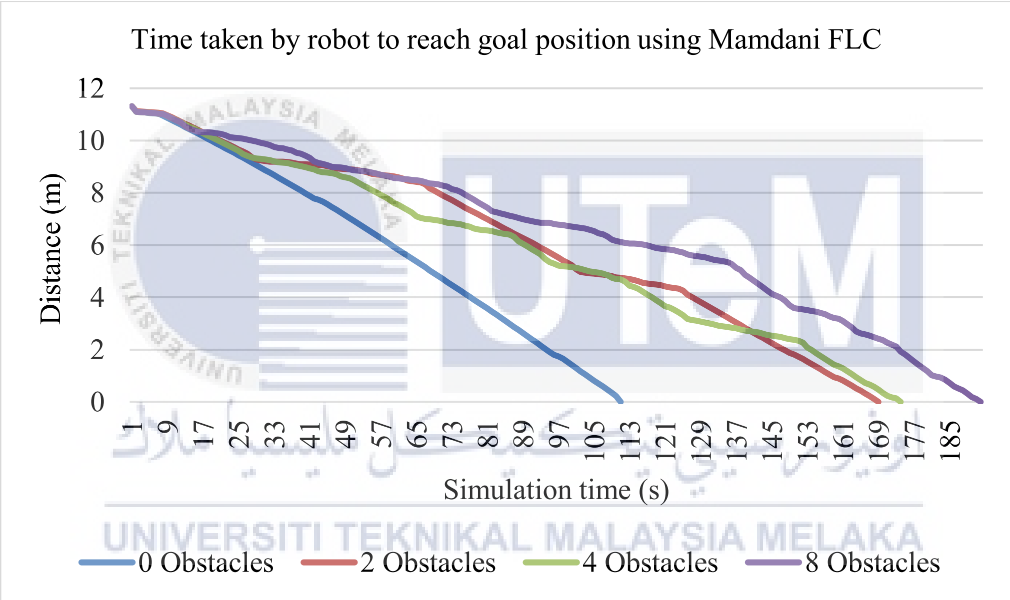


Figure 4.5: Time taken by robot to reach goal position using Mamdani FLC

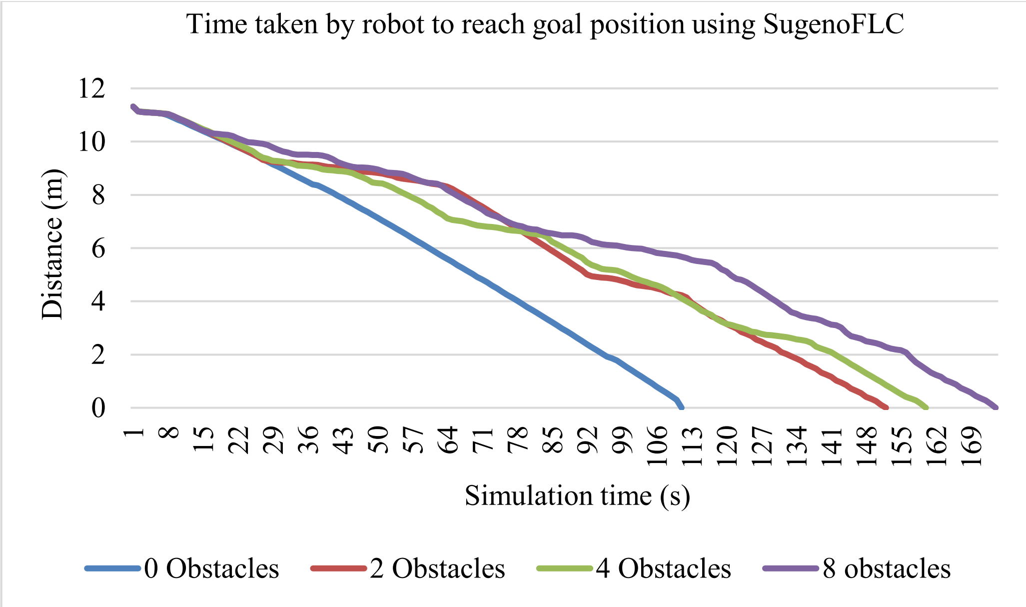


Figure 4.6: Time taken by robot to reach goal position using Sugeno FLC

Based on the Figure 4.5 and Figure 4.6, the time taken by Pioneer robot to reach the target for both the Mamdani FLC and Sugeno FLC increases as the obstacles increases. For the Mamdani FLC, time recorded by Pioneer to reach the goal position in environment with 0 obstacles, 2 obstacles, 4 obstacles and 8 obstacles are 111.539 seconds, 169.155 seconds, 174.572 seconds and 192.555 seconds respectively. For the Sugeno FLC, time recorded by Pioneer to reach the goal position in environment with 0 obstacles, 2 obstacles, 4 obstacles and 8 obstacles are 111.538seconds, 152.326seconds, 160.801 seconds and 174.554 seconds respectively. The time taken by robot increases as the obstacles increases is due to behavior of robot which uses FLC to avoid the collision with obstacles to move to the safest path. Both controller recorded an exact time which is about 111s seconds in the environment with the zero obstacles, while different time is recorded in other environments. This prove the efficiency of the Mamdani FLC and Sugeno FLC only differs in the environment with more than one obstacles.

To further validate more on the time taken by the robot to reach the goal position, 10 more iteration is done as shown in Table 4.2 and Table 4.3 for Mamdani FLC and Sugeno FLC, respectively. The average values in each table is then used to calculate the efficiency of each system as shown in Table 4.4.

Table 4.2: Average time taken by robot to reach goal position for Mamdani FLC

Iteration	Time(s) 0 Obstacles	Time(s) 2 Obstalces	Time(s) 4 Obstacles	Time(s) 8 Obstacles
1	111.322	169.253	174.253	192.258
2	111.800	169.577	174.235	192.256
3	111.278	169.523	174.226	192.663
4	111.008	168.123	174.578	192.889
5	111.664	169.248	173.248	192.563
6	111.700	168.268	173.427	192.654
7	111.738	168.846	173.055	191.465
8	111.719	169.486	174.729	193.523
9	111.525	169.647	179.249	192.654
10	111.631	169.579	174.724	192.620
Average	111.539	169.155	174.572	192.555

Table 4.3: Average time taken by robot reach goal position for Sugeno FLC

Iteration	Time(s) 0 Obstacles	Time(s) 2 Obstalces	Time(s) 4 Obstacles	Time(s) 8 Obstacles
1	111.081	152.162	160.313	174.271
2	111.321	151.458	160.821	174.135
3	111.593	152.456	160.824	175.555
4	111.834	152.284	160.767	175.646
5	111.451	152.589	160.661	174.271
6	111.999	153.895	160.975	174.178
7	111.863	152.265	161.043	174.344
8	111.829	151.456	161.017	174.796
9	111.238	152.257	160.751	174.238
10	111.174	152.439	160.842	174.110
Average	111.538	152.326	160.801	174.554

Table 4.4: Efficiency of the FLC

	Average time taken for Pioneer to reach Goal (s) using		Efficiency of the robot based on time taken to reach the goal	Comment
	Mamdani FLC	Sugeno FLC		
0 Obstacles	111.539	111.538	$\frac{111.539-111.538}{111.539} X\ 100\%$ $=\ 0\%$	Both controller achieved same efficiency
2 Obstacles	169.155	152.326	$\frac{169.155-152.326}{169.155} X\ 100\%$ $=\ 1\%$	Sugeno FLC is 1% efficient than the Mamdani FLC
4 Obstacles	174.572	160.801	$\frac{174.572-160.801}{174.572} X\ 100\%$ $=\ 7.89\%$	Sugeno FLC is 7.89% efficient than the Mamdani FLC
8 Obstacles	192.555	174.554	$\frac{192.555-174.554}{192.555} X\ 100\%$ $=\ 9.35\%$	Sugeno FLC is 9.35% efficient than the Mamdani FLC

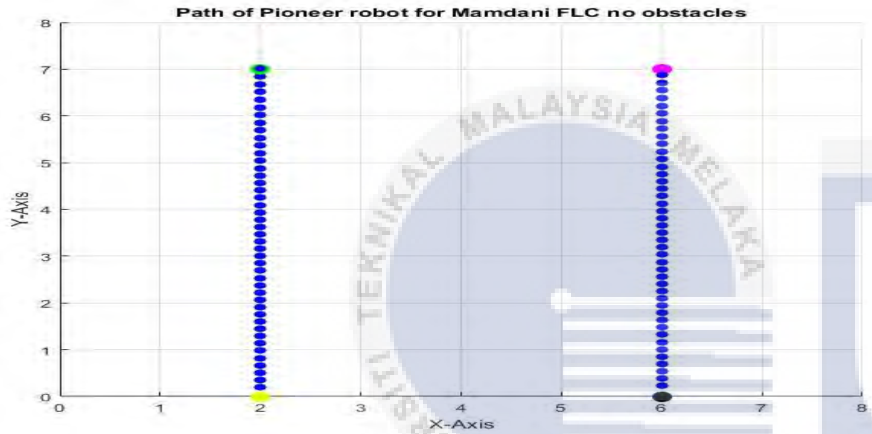
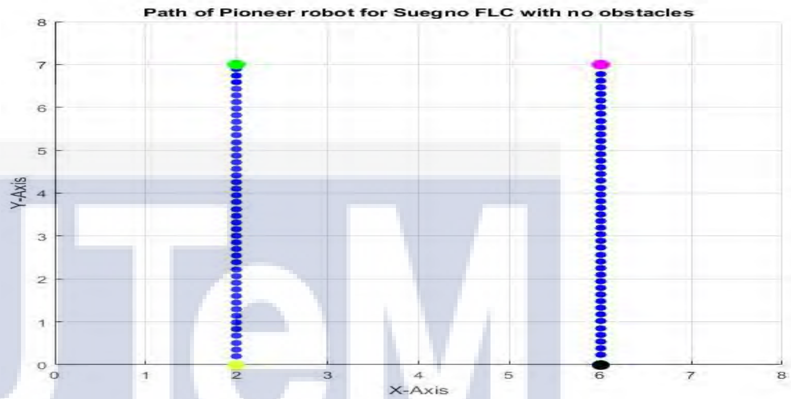
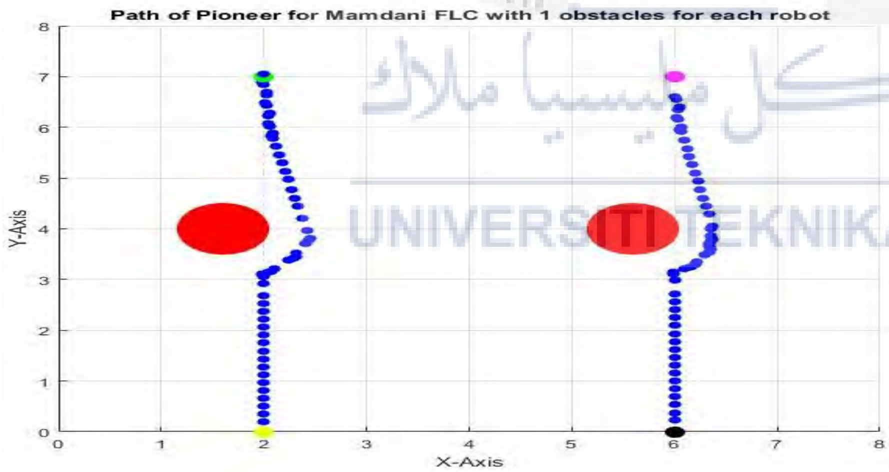
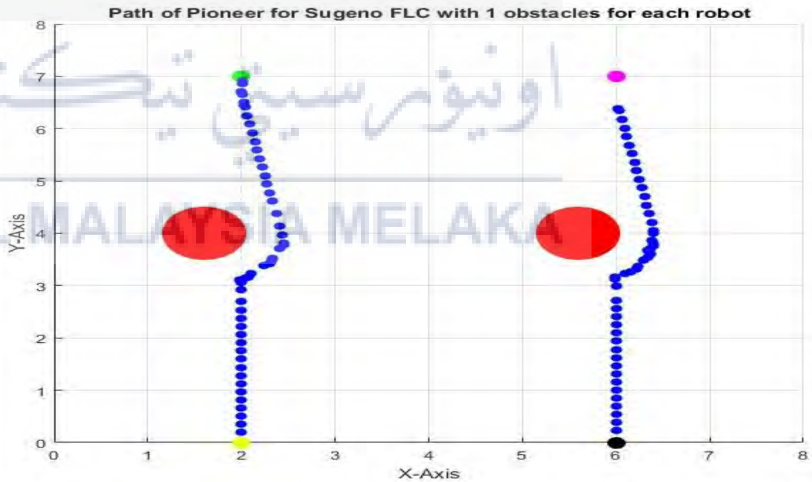
From Table 4.3 we can conclude that the Sugeno FLC is about 7.5% efficient than the Mamdani FLC in single robot environment. The faster response time of the Sugeno FLC is due to its computational efficiency that eliminates defuzzification process and ability to produce connstant output which is suitable in controlling the robot’s linear velocity than Mamdani FLC. However, both FLC able to avoid obsatacles and reach the target successfully in all environments for single robot.

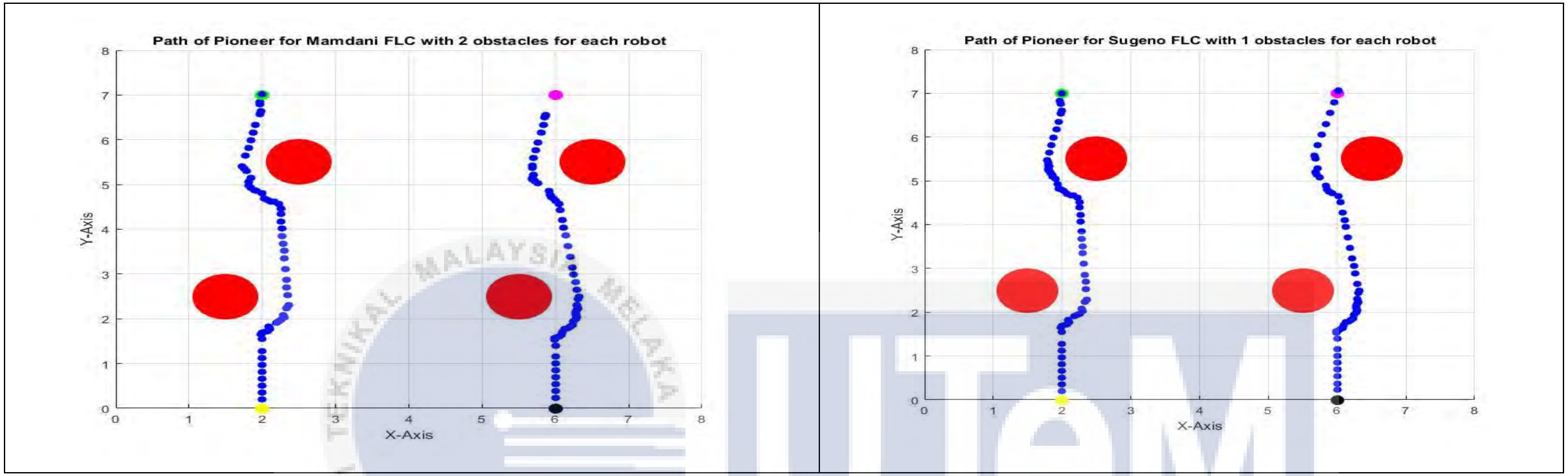
4.3 Results for Multi Robot

4.3.1 Results for Path travelled by Robots

The purpose of multirobot environment simulation is to validate the results of single robot simulation that shows that the Sugeno FLC is more faster and generates more smoother path than the Mamdani FLC. The simulation of multi robots is carried out in three different environments with the number of obstacles in each environment are 0 obstacle, 1 obstacles and 2 obstacles for each robot respectively. Two Pioneer P3-DX robots are used in multi robot simulation. The Table 4.5 shows the scatter plot of path generated by two Pioneer robots in each environment for the Mamdani FLC and Sugeno FLC.

Table 4.5: Path of Pioneer robot for Mamdani and Sugeno FLC in different environments.

Mamdani	Sugeno
	
	


Path of Pioneer for Mamdani FLC with 2 obstacles for each robot
Y-Axis
X-Axis
Path of Pioneer for Sugeno FLC with 1 obstacles for each robot
Y-Axis
X-Axis

Based on Table 4.5, the 'yellow' circles and 'green' circles, represents the initial position and goal position respectively for first Pioneer robot, whereas the 'black' circles and 'magneta' circles, represents the initial position and goal position respectively for the second Pioneer robot. The 'red' circles represents the obstacles and the blue line indicates the scatter plot of the path generated by robot in each environment. Based on Table 4.5, both Pioneer P3-Dx robots successfully navigate itself from an initial position to goal position without colliding with any obstacles. This results guaranteed the developed FLC can be used in the multirobots environments too.

4.3.2 Results for smoothness of path travelled by Pioneer P3-DX

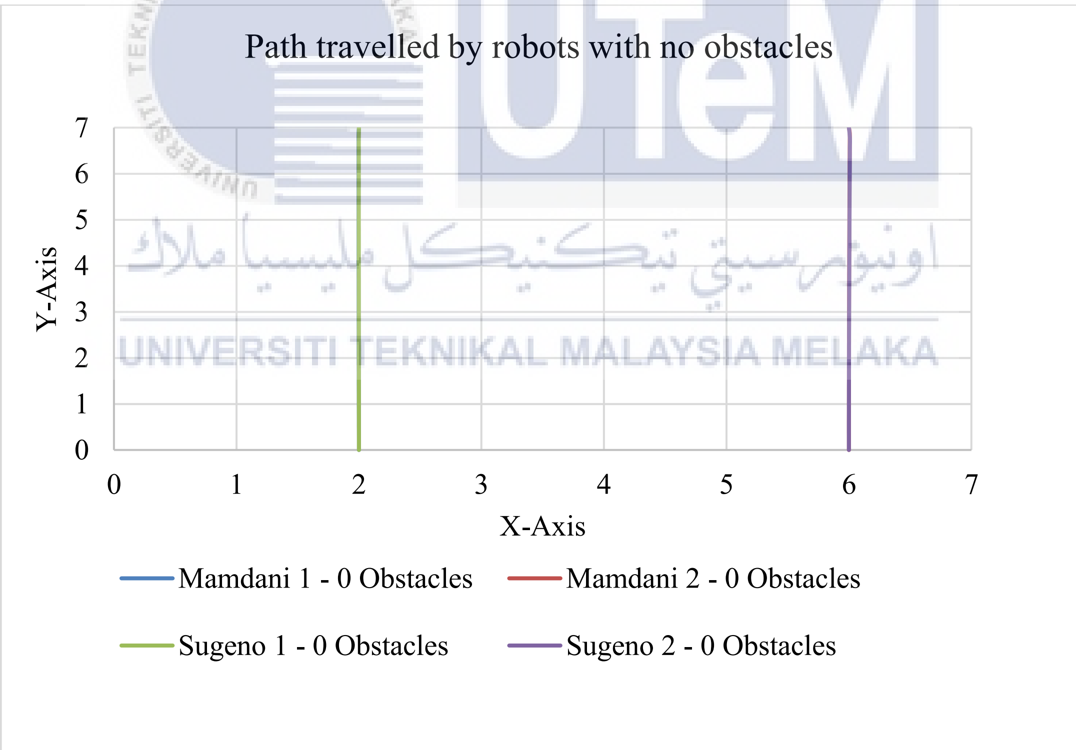


Figure 4.7: Path travelled by robots with no obstacles

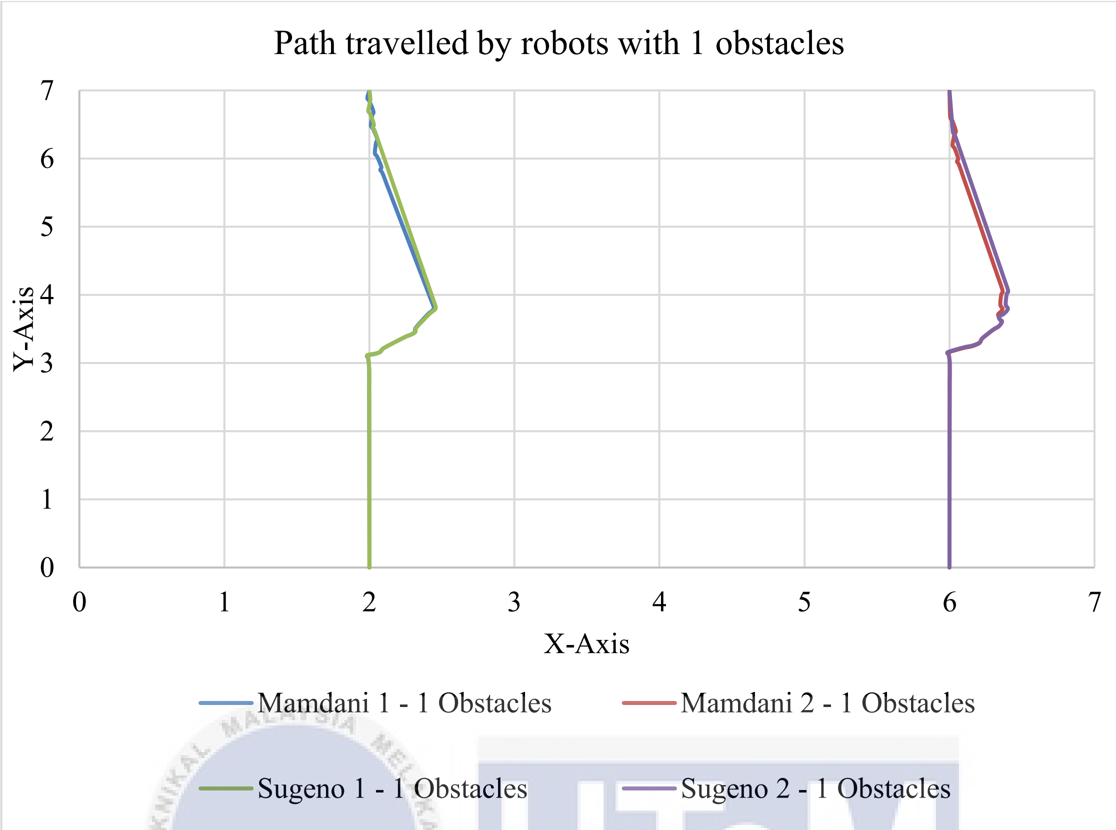


Figure 4.8: Path travelled by robots with 1 obstacles each

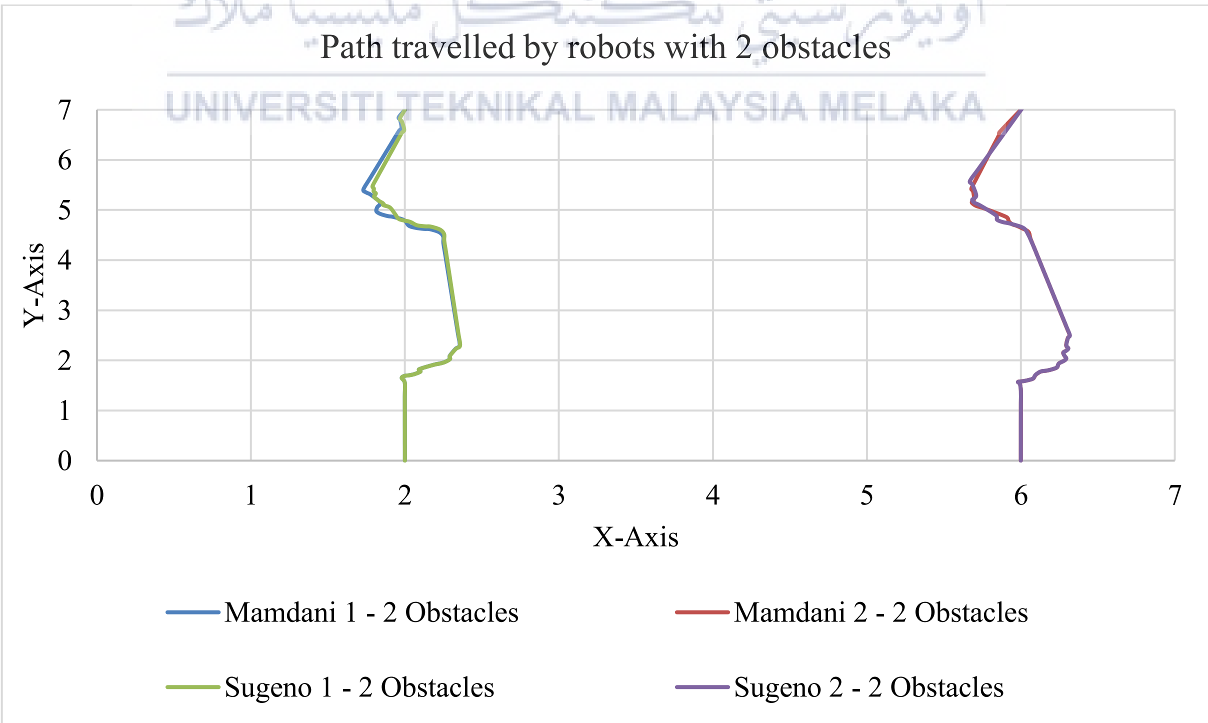


Figure 4.9: Path travelled by robots with 2 obstacles each

Figure 4.7, Figure 4.8 and Figure 4.9 disclose the comparison of the path travelled by the multirobot from initial position to goal position in different environments by using the Mamdani FLC and Sugeno FLC. Based on Figure 4.7, there are not significant difference in path travelled by Pioneer using Mamdani FLC and Sugeno FLC. Since there is no obstacle in between the path of robot, FLC does not initialized in that environment. However, close observation in Figure 4.8 and Figure 4.9 reveals there are significant differences in the path travelled by the robots between Mamdani FLC and Sugeno FLC. This is because the controllers processed the input data as the sensors detects the obstacles. As we can see from Figures 4.8 and Figure 4.9, the path travelled by robots using the Sugeno FLC is more adequate than the path travelled by robots using the Mamdani FLC. This is due to the capability of the Sugeno FLC to produce constant output and faster response time unlike the Mamdani FLC which only capable of produsing crisp output after defuzzification process due to the output of the Mamdani FLC is in Fuzzy sets. The results in multirobot environment are similar to the results obtained in single robot environment.

4.3.3 Efficiency of the controller

The efficiency of the controller for multirobot environments is tested in three different environments under different number of obstacles in each environment by using two Pioneer P3-Dx robot. The initial distance of the robot from the goal position is 7 m. The distance between the robots and goal positions decreases as the robots move towards the goal. The efficiency of the controller is described in term of time taken by both robot to reach the target. Figure 4.10 and Figure 4.11 shows the average time taken by two Pioneer P3- DX for each environment using Mamdani FLC and Sugeno FLC.

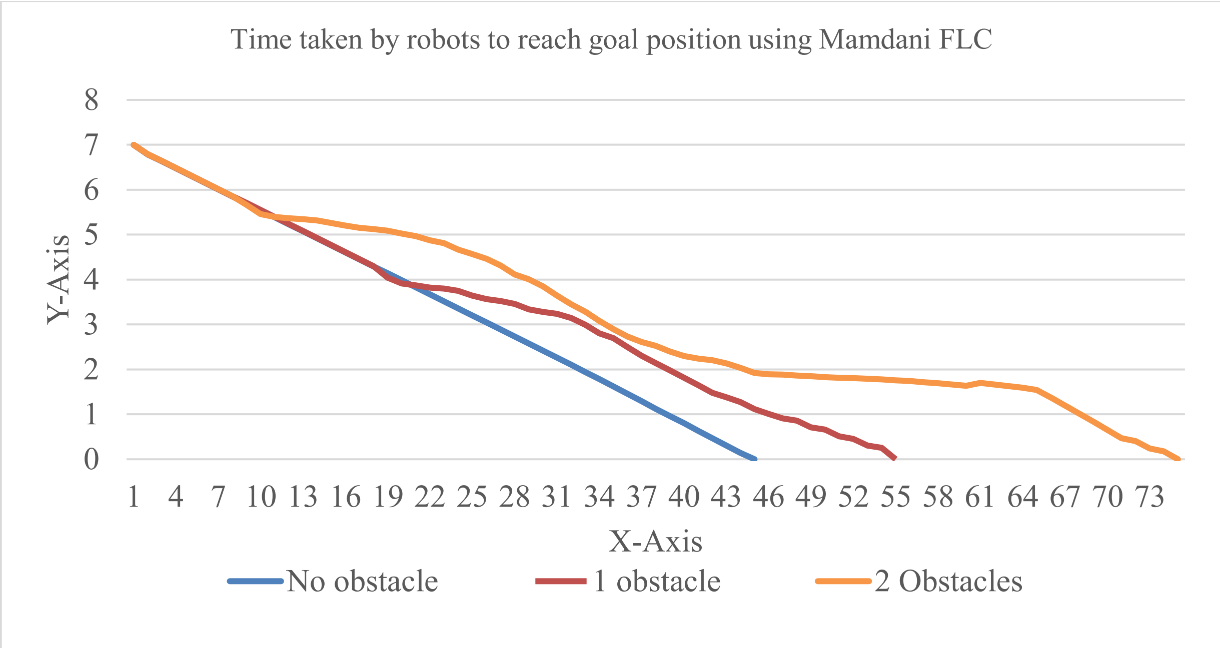


Figure 4.10: Time taken by robots to reach goal position using Mamdani FLC

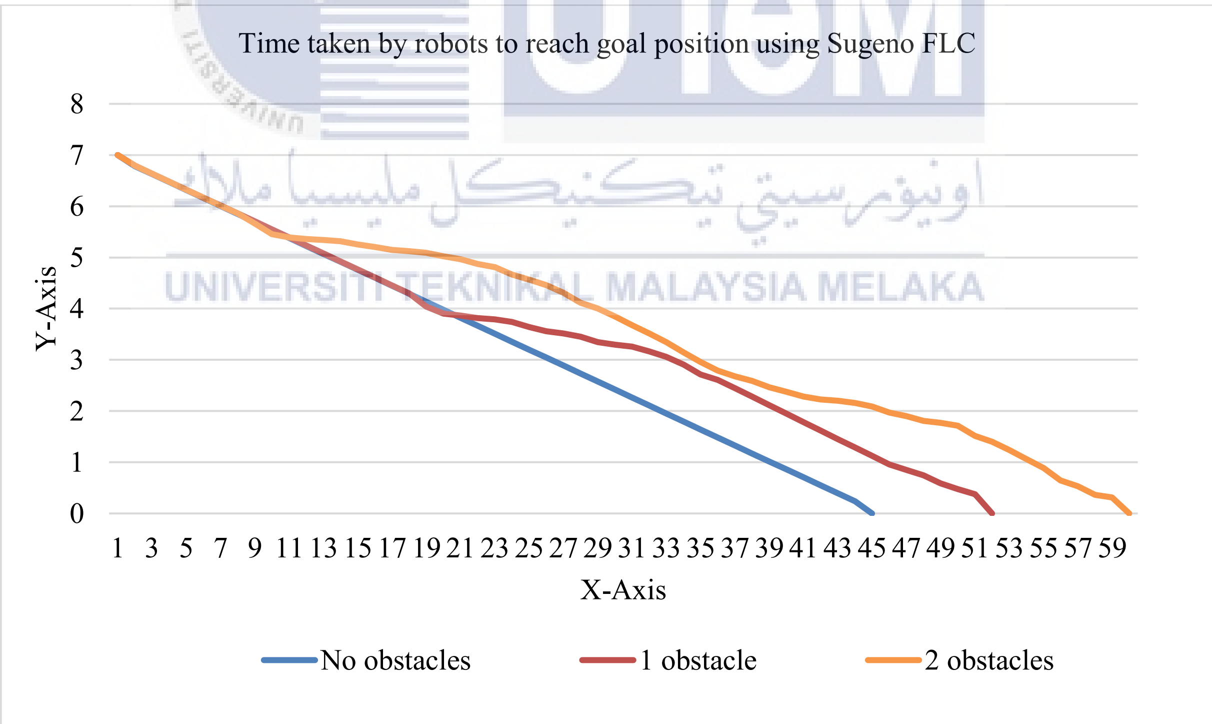


Figure 4.11: Time taken by robots to reach goal position using Sugeno FLC

Based on the Figure 4.10 and Figure 4.11, the average time needed for two Pioneer P3-DX robot to reach the target increases as the obstacles increases in both Mamdani FLC and Sugeno FLC. For the Mamdani FLC, average time recorded by two Pioneer robots to reach the goal position in environments with 0 obstacles, 1 obstacles and 2 obstacles are 43.456 seconds, 53.646 seconds and 73.456 seconds respectively. For the Sugeno FLC, an average time recorded by two Pioneer robots to reach the goal position in the environments with 0 obstacles, 1 obstacle and 2 obstacles are 43.679 seconds, 50.993 seconds and 160.801 seconds respectively. The average time taken by robot increases as the obstacles increases is due to the behavior of robot which uses FLC to avoid the collision with obstacles. Both controller recorded a same time which is about 43s seconds in the environment with the zero obstacles with various of time in all other environmetns. This to proves the efficiency of the Mamdani FLC and Sugeno FLC that is only differs in the environment with one or more obstacles. To validate more on the average time taken by the two robots to reach the goal position, 10 more iteration is done as shown in Table 4.6 and Table 4.7 for Mamdani and Sugeno, The average values in each environment is then used to calculate the overall efficiency of each system as shown in Table 4.8.

Table 4.6: Average time taken by robots to reach goal position for Mamdani FLC

Iteration	Time(s) 0 Obstacles	Time(s) 2 Obstalces	Time(s) 4 Obstacles
1	43.456	53.646	73.456
2	43.575	53.987	73.546
3	43.565	53.468	73.644
4	43.548	53.176	73.210
5	44.847	53.147	73.0456
6	43.795	53.079	73.0468
7	44.0155	53.6478	73.0648
8	43.578	53.046	73.0679
9	43.012	53.648	73.489
10	43.057	53.977	74.021
Average	43.645	53.482	73.359

Table 4.7: Average time taken by robots to reach goal position for Sugeno FLC

Iteration	Time(s) 0 Obstacles	Time(s) 1 Obstacles	Time(s) 2 Obstacles
1	43.679	50.993	58.982
2	43.578	50.442	58.324
3	43.004	50.433	58.607
4	43.698	50.154	58.647
5	43.679	50.487	58.632
6	43.578	50.460	53.475
7	43.579	50.214	53.024
8	43.293	49.687	53.046
9	43.792	50.132	53.047
10	43.674	50.876	53.987
Average	43.555	50.389	55.977

Table 4.8: Efficiency of the FLC

	Average time taken for Pioneer to reach Goal (s) using		Efficiency of the robot based on time taken to reach the goal	Comment
	Mamdani FLC	Sugeno FLC		
0 Obstacles	43.645	43.555	$\frac{43.646-43.555}{43.645}\ X\ 100\%$ $=\ 0\%$	Both controller achieved same efficiency
2 Obstacles	53.482	50.389	$\frac{53.482-50.389}{53.482}\ X\ 100\%$ $=\ 5.7\%$	Sugeno FLC is 5.7% efficient than the Mamdani FLC
4 Obstacles	73.359	55.977	$\frac{73.539-55.977}{73.359}\ X\ 100\%$ $=\ 23.694\%$	Sugeno FLC is 23.694% efficient than the Mamdani FLC

From Table 4.8 we can conclude that the Sugeno FLC is about 12.06% efficient than the Mamdani FLC in multirobot environment. The faster response time of the Sugeno FLC is due to its computational efficiency that eliminates defuzzification process and ability to produce a constant output which is suitable in controlling the robot's linear velocity than Mamdani FLC. Moreover, both system able to avoid obstacles and reach the target successfully in all the multi robot environments.

4.4 Summary

The designed Mamdani FLC and Sugeno FLC is able to navigate the robot from initial position to target position by avoiding obstacles along the path of robot in both single and multirobot environments. This validates the efficiency of developed controllers in processing input data to give a systematic output based on 15 Fuzzy rules. However, the path generated by Sugeno FLC is more smoother than path generated by Mamdani FLC. Furthermore, the Sugeno FLC reach the target location more faster than the Mamdani FLC. This is mainly because of the computational efficiency of Sugeno FLC which can produce a constant output without any delay. The Mamdani FLC does not able to produce such efficiency due to the delay during the defuzzification process to change the output variable in fuzzy sets to a single number. Thus, the Sugeno FLC offers an efficient and faster response

CHAPTER 5.0

CONCLUSION & RECOMMENDATION

5.1 Conclusion

Autonomous mobile robot should consist of a controller that help the robot to avoid collision along its way from one point to another. The obstacles may be dynamic and static regardless of it sizes. FLC is a powerful tool to that help the robot to achieve a good collision avoidance behaviour. This project is mainly focuses on obstacle avoidance behaviour of mobile robot in single robot and multi robot environment consists of static obstacles using two type of FLC. Theoretical background and related researched has regarding autonomous robot and FLC are studied and analysed for guidance to complete this.

In a nutshell, all the objectives of this project are achieved. To achieve the first objective, two type of FLC namely, Mamdani and Sugeno FLC are developed using Fuzzy Logic Toolbox in Matlab 2016b based on the Pioneer-P3-DX robot. The input for both controller is the sensor value of five proximity sensors equipped in the robot. Whereas the output for the controllers are the left and right velocity of the robot. 15 fuzzy rules are implemented for each controller based on the movement of robot in Virtual Robotic Experimentation Platform (V-REP). Upon completing the designation of the controller, first objective are successfully achieved.

We also test the designed controller is then tested in V-REP robotic simulator to validate the developed controller. Four environment with no obstacles, 2 obstacles, 4 obstacles and 8 obstacles are created for single robot testing and three environments with no obstacles, one obstacle for each robot and 2 obstacle for each robot is created for multirobot testing. Only two robots are considered for multirobot environment. After the implementation of the controller in Pioneer robot, the robot able to navigate from initial position to final position without colliding with any obstacles in both single and multirobot environments. This validate the developed controllers and second objectives are achieved.

Finally, smoothness and efficiency of Mamdani and Sugeno Fuzzy Logic Controller are compared based on the path and time taken to reach the goal in different environment. Based on the results and analyses, the Sugeno FLC able to produce smoother path and reach the goal faster than Mamdani FLC. This is due to the computational efficiency of Sugeno that able to give constant output which is suitable in controlling the robot much more than Mamdani FLC which give output in fuzzy sets.

5.2 Recommendation

Few recommendation are suggested for future works related to the FLS. First and foremost, designing combined fuzzy controller for both obstacle avoidance behaviour and target seeking behaviour may reduce time taken by the robot to reach the target position because, the fuzzy system have complete control on the velocity of left and right wheels compared to this project which utilize fuzzy system only for obstacle avoidance behaviour.

Usually, the workspace of mobile robot is unknown as there might be static and dynamic obstacles. Mobile robot should have the capability of avoiding static and dynamic obstacles to reach the target without colliding with any obstacles. However, the FLC developed in this project only have ability to avoid static obstacles. Thus, designing a robust FLC which able to avoid both dynamic and state obstacle is essential for mobile robot working in uncertain environment.

Furthermore, implementing hardware to test the Fuzzy Logic Controller will be adequate for validation of the controller. This is because, simulation could not produce the actual interference which may face by the robot in real situations such as unstable friction of wheel, varied sizes of the obstacles, reducing battery capacity and efficiency of the sensors. Oter than that, integrating guzzy logic system with other artificial intelligence technique such as Artificial Neural Network may increase the efficiency of the robot in avoiding obstacles and reaching target.

REFERENCES

[1] M. K. Habib, "Humanitarian Demining: The problem, Difficulties, Priorities, Demining Technlogy and Challenge for Robotics," *International Journal of Advanced Robotics System,* vol. 4, no. 2, pp. 151-172, 2007.

[2] R. Jiménez, "Cluster Munition Monitor 2015," International Campaign to Ban Landmines – Cluster Munition Coalition (ICBL-CMC), 2015.

[3] N. E. Walsh and W. S. Walsh, "Rehabilitation of Landmine Victims — the Ultimate Challenge," *Bulletin of the World Health Organization,* 2003.

[4] M. K. Habib, "Mine Clearance Techniques and Technologies for Effective Humanitarian Demining," in *27th Annual Conference of the IEEE Industrial Electronics Society*, 2002.

[5] M. K. Habib, "Service Robots in Humanitarian Landmine Clearance," in *International Symposium on Robots (ISR2001)*, Seoul, Korea, 2001.

[6] J. Nicoud, "Vehicles and Robots for Humanitarian Demining," *The Industrial Robot,* vol. 24, no. 2, pp. 164-168, 1997.

[7] M. Habib and J. D. Nicoud, "PEmex-B Autonomous Demining Robots: Perception and Navigation Strategies," *IROS'95, Pittsburgh,* p. 419–424, Aug. 1995.

[8] U. Farooq, K. M. Hasan, M. U. Asad and S. O. Saleh, "Fuzzy Logic Based Wall Tracking Controller for Mobile Robot Navigation," in *7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, July 2002.

[9] Neha Preet Kaur, Vikram Verma, "Comparison Study of Mamdani Method and Sugeno Method in The," *International Journal of Scientific Research Engineering & Technology (IJSRET,* vol. 4, no. 3, pp. 166-169, 2015.

[10] T. T. Mac, C. Copot, R. D. Keyser, T. D. Tran and T. Vu, "Control for Autonomous Mobile Robot," *Journal of Automation and Control Engineering,* vol. 4, no. 1, pp. 65-70, 2016.

[11] K. G. Sugeno M., "Structure identification of fuzzy models," Fuzzy Sets and Systems," 1988.

[12] E. Mamdani, "“Applications of fuzzy algorithms for simple dynamic plant," in *Proc. of IEE, V121*, 1974.

[13] R. M. Murray, Z. Li and S. S. Sastry, A Mathematical Introduction to Robotics, California: CRC Press, 1994.

[14] R. Manseur, Robot Medelling and Kinematics, Boston, Massachusetts: Charles River Media, an imprint of Thomson Learning Inc, 2006.

[15] R.Rocha, J.Dias and A.Carvalho, "Cooperative Multi Robot System: A study of vision based 3d mapping using information theory," *Robotics and Automation System,* Vols. 3-4, no. 53, pp. 282-311, 2005.

[16] A. K. Jitendra R.Raol, Mobile Intelligent Autonomous System, United States: CRC press, 2013.

[17] A. Elfes, "Sonar-Based Real-World Mapping and Navigation," *IEEE Journal of Roobotics and Automation,* vol. 3, no. 3, 1987.

[18] C. R. Sv. Noykov, "Calibration and interface of a polaroid ultrasonic sensor for mobile robots," *Sensors and Actuators A: Physical,* vol. 135, no. 1, pp. 169-178, 2007.

[19] M. M. Almasri, A. M. Alajlan and K. M. Elleithy, "Trajectory Planning and Collision Avoidance Algorithm for Mobile Robotics System," *IEEE Sensors Journal ,* vol. 16, no. 12, pp. 5021-5028, 2016.

[20] C.-H. Wu, I.-S. Lin, M.-L. Wei and T.-Y. Cheng, " Target Position Estimation by Genetic Expression Programming for Mobile Robots With Vision Sensors," *Instrumentation and Measurement,* vol. 62, no. 12, pp. 3218-3230, 2013.

[21] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Robotics and Automation. Proceedings*, Stanford, 2003.

[22] L. X. Zhang Ying, "Leader-follower Formation Control and Obstacle Avoidance of Multi-robot Based on Artificial Potential Field," in *27th Chinese Control and Decision Conference*, Shangai, 2015.

[23] M. F. Alves and C. R. Lopes, "Obstacle avoidance for mobile robots : a Hybrid Intelligent System based on Fuzzy Logic and Artificial Neural Network," in *IEEE International Conference on Fuzzy Systems (FUZZY)*, Brazil, 2016.

[24] H. M. Eraqi, Y. E. Eldin and M. N. Moustafa, "Reactive Collision Avoidance using Evolutionary Neural Networks".

[25] V. d. C. Santos, C. F. M. Toledo and F. S. Osório, "An exploratory path planning method based on genetic algorithm for autonomous mobile robots," in *IEEE Congress on Evolutionary Computation (CEC)*, 2015.

[26] P. Das, B. Sahoo, H.S.Behera and S.Vashisht, "An Improved Particle Swarm Optimization for Multi-Robot Path Planning," in *International Conference on*

Innovation and Challenges in Cyber Security (ICICCS 2016), Burla,Odisha,India, 2016.

[27] J.-X. Xu, Z.-Q. Guo and T. H. Lee, "Design and Implementation of a Takagi–Sugeno-Type Fuzzy Logic Controller on a Two-Wheeled Mobile Robot," *IEEE Transactions on Industrial Electronics ,* vol. 60, no. 12, pp. 5717-5728, 2013.

[28] M. E.H and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies,* vol. 7, no. 1, pp. 1-13, 1975.

[29] Z. L.A., "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 3, no. 1, pp. 28-44, 1973.

[30] S. M., "Industrial applications of fuzzy control,," *Elsevier Science Pub. Co,* 1985.

[31] V. Cazac and I. Nuca, "The control system modeling of winder with hybrid Fuzzy-PID," in *2016 International Conference and Exposition on Electrical and Power Engineering (EPE)*, 2016.

[32] K. Pytel, "Hybrid fuzzy-genetic algorithm applied to clustering problem," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)* , 2016.

[33] P. Rusu, E. M. Petriu, T. E. Whalen, A. Cornell and H. J. W. Spoelder, "Behavior-based neuro-fuzzy controller for mobile robot navigation," *IEEE Transactions on Instrumentation and Measurement ,* vol. 52, no. 4, pp. 1335-1340, 2003.

[34] N. Malik and N. Slimane, "Autonomous navigation with obstacle avoidance of tricycle mobile robot based on fuzzy controller," in *Electrical Engineering (ICEE), 2015 4th International Conference on* , Boumerdes,Algeria, 2016.

[35] "MIMO Fuzzy Control for Autonomous Mobile Robot," *Journal of Automation and Control Engineering Vol. 4, No. 1, February 2016,* vol. 4, no. 1, pp. 65-70, 2016.

[36] A. Pandey, R. K. Sonkar, K. K. Pandey and D. R. Parhi, "Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller," in *2014 IEEE 8th International Conference on Intelligent Systems and Control (ISCO)* , 2014.

[37] M. I. Ibrahim, N. Sariff, J. Johari and N. Buniyamin, "Mobile robot obstacle avoidance in various type of static environments using fuzzy logic approach," in *2014 2nd International Conference on Electrical, Electronics and System Engineering (ICEESE)* , 2014.

[38] M. R. O. A. B. U. Fuzzy, "Siti Hajar Ashikin Mohammad; Muhammad Akmal Jeffril; Nohaidda Sariff," in *IEEE 3rd International Conference on System Engineering and Technology*, Shah Alam, Malaysia, 2013.

[39] M. M. Almasri, K. M. Elleithy and A. M. Alajlan, "Development of Efficient Obstacle Avoidance and Line Following Mobile Robot with the Integration of Fuzzy Logic System in Static and Dynamic," in *Long Island Systems, Applications and Technology Conference (LISAT)*, 2016.

[40] U. Farooq, K. M. Hasan, G. Abbas and M. U. Asad, "Comparative Analysis of Zero Order Sugeno and Mamdani Fuzzy Logic Controllers for Obstacle Avoidance Behavior in Mobile Robot Navigation," in *Current Trends in Information Technology (CTIT), 2011 International Conference and Workshop on* , Dubai, United Arab Emirates, 2011.

[41] N. P. Kaur and V. Verma, "Comparison Study of Mamdani Method and Sugeno Method in The Navigation System for Indoor Mobile Robot," *International Journal of Scientific Research Engineering & Technology (IJSRET,* vol. 4, no. 3, pp. 166-169, 2015.

[42] "Coppelia Robotics V-REP: Create. Compose. Simulate. Any Robot.," [Online]. Available: http://www.coppeliarobotics.com/. [Accessed 13 June 2017].

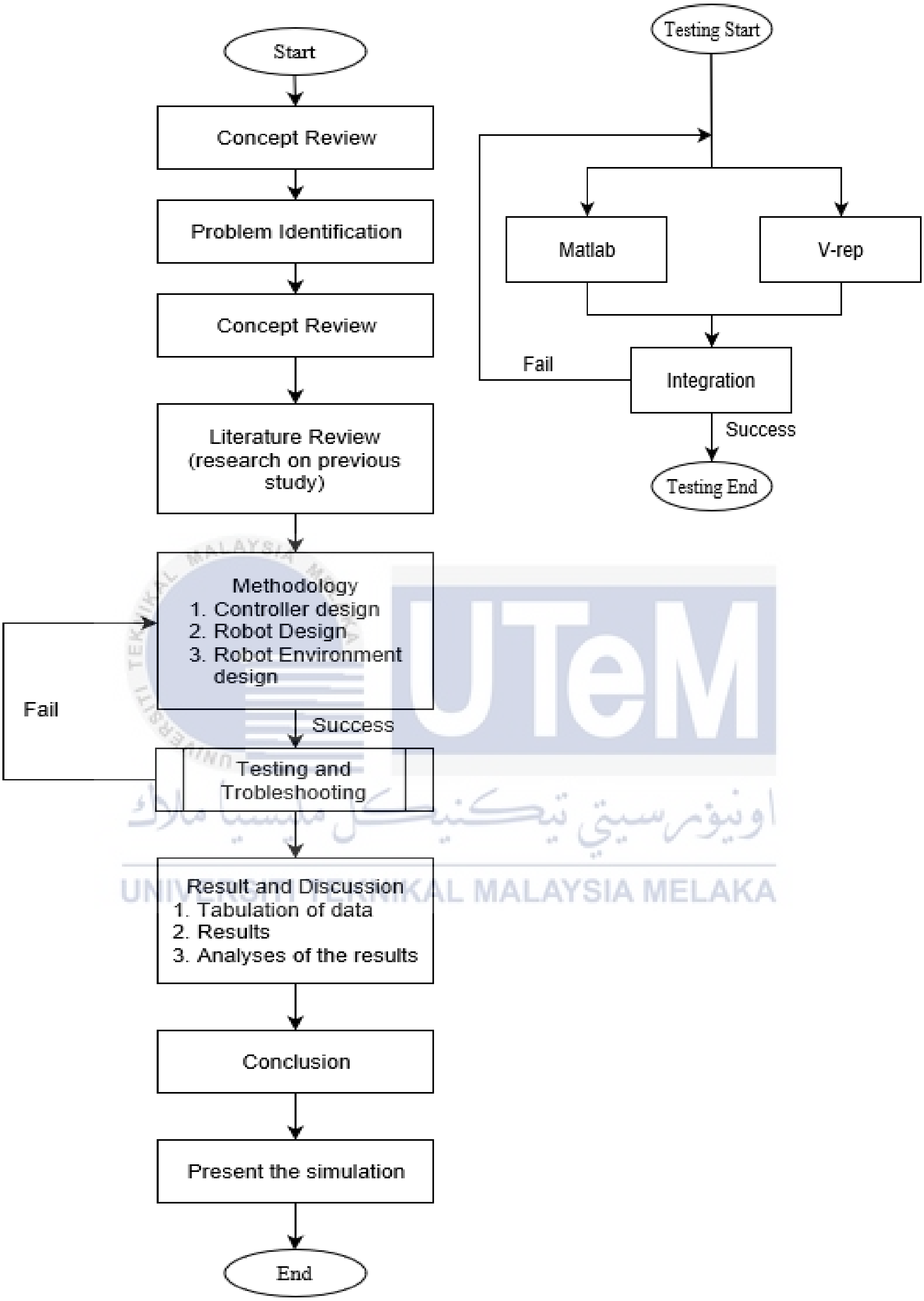
APPENDICES

A. RESEARCH GANTT CHART (FYP 1 & FYP 2)

Job Description FYP 1	Week														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Final Year Project Title Selection															
Project Prospect Discussion with Supervisor															
Journal & Research Paper Review															
Methodology Outline															
Design of Fuzzy Logic Controller															
Data Collection & Analysis															
Report Writing Preparation															
Final Year Project 1 Presentation															
Submission of Report															

Job Description FYP 2	Week														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Integrate V-rep with Matlab															
Testing and validating Fuzzy Logic															
Data Collection & Analysis															
Report Writing Preparation															
Final Year Project 2 Presentation															
Submission of Report															

B. RESEARCH PROGRESS CHART


Start
Concept Review
Problem Identification
Concept Review
Literature Review (research on previous study)
Methodology
1. Controller design
2. Robot Design
3. Robot Environment design
Fail
Success
Testing and Trobleshooting
Result and Discussion
1. Tabulation of data
2. Results
3. Analyses of the results
Conclusion
Present the simulation
End
Testing Start
Matlab
V-rep
Fail
Integration
Success
Testing End

C. MILESTONE

Milestone is a significant or important fixture or event in the progress of the project. Specific stage along a project timeline marked using milestone.

Table 3.1: Milestone

No.	Activity	Date / Period
1	Developing the controller design	27 January 2017
2	Developing the Robotic Simulator Environment	28 February 2017
3	Integrating Controller and Robotic Simulator	23 March 2017
4	Validation of controller	22 April 2017
5	Data collection	30 April 2017
6	Analyses of the controller	8 May 2017