FUZZY BASED MOBILE ROBOT NAVIGATION ON MOTION TRACKING USING ROS

AHMAD ALIF SULAIMAN BIN MOHD JOHARI

B. ENG(HONS.) ELCTRICAL ENGINEERING (ELECTRONIC)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	AHMAD ALIF SULAIMAN BIN MOHD JOHARI
Date of Birth	:	3/8/1996
Title	:	FUZZY BASED MOBILE ROBOT NAVIGATION ON MOTION TRACKING USING ROS
Academic Session	:	SEMESTER I 2021/2022

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☐	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

ahmad alif
(Student's Signature)

960803-01-7091
New IC/Passport Number
Date: 13/2/2022

Hamzah
(Supervisor's Signature)

Assoc Prof Ts Dr Hamzah Ahmad
Name of Supervisor
Date: 11.2.2022

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name

Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

____

(Supervisor's Signature)

Date: 11.2.2022

Stamp: DR. HAMZAH BIN AHMAD
ASSOCIATE PROFESSOR
FACULTY OF ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN
PAHANG DARUL MAKMUR
TEL : 09-424 6024 FAX : 09-424 6111

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Engineering/ Master of Science in …………………………..

(Supervisor's Signature)

Full Name : Assoc Prof Ts Dr Hamzah Ahmad

Position : Senior Lecturer

Date : 11.2.2022

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

ahmad alif
____

(Student's Signature)

Full Name : AHMAD ALIF SULAIMAN BIN MOHD JOHARI

ID Number :EA18058

Date :13/2/2022

FUZZY BASED MOBILE ROBOT NAVIGATION ON MOTION TRACKING USING ROS

AHMAD ALIF SULAIMAN BIN MOHD JOHARI

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor Degree in Electrical Engineering (Electronic)

Faculty of Electrical & Electronics Engineering
UNIVERSITI MALAYSIA PAHANG

OCTOBER 2021

ACKNOWLEDGEMENTS

I am grateful and would like to express my sincere gratitude to my supervisor Dr Hamzah Bin Ahmad for his invaluable guidance, continuous encouragement and constant support in making this research possible. I really appreciate his guidance from the initial to the final level that enabled me to develop an understanding of this research thoroughly. Without his advice and assistance, it would be a lot tougher to completion. I also sincerely thanks for the time spent proofreading and correcting my mistakes.

My sincere thanks go to all lecturers and members of the staff of the Electrical and Electronic Engineering Department, UMP, who helped me in many ways and made my education journey at UMP pleasant and unforgettable. This four-year experience with all you guys will be remembered as important memory for me to face the new chapter of life as an engineer.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life. I am really thankful for their sacrifice, patience, and understanding that were inevitable to make this work possible. Their sacrifice had inspired me from the day I learned how to read and write until what I have become now. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to achieve my dreams.

Lastly, I would like to thanks any person which contributes to my final year project directly on indirectly. I would like to acknowledge their comments and suggestions, which was crucial for the successful completion of this study.

ABSTRAK

Dengan kemajuan teknologi, robot mudah alih menjadi popular. Menurut Karthiga (2014) negara maju mendapat bantuan robot bergerak untuk menyelamatkan manusia di kawasan bencana dan ini telah terbukti sebagai kaedah produktif untuk menghilangkan kesalahan manusia. Dalam keadaan seperti itu, robot diperlukan untuk menavigasi di persekitaran yang bermusuhan seperti runtuh bangunan atau kawasan yang terkena kebakaran. Tesis ini membentangkan terutamanya reka bentuk dan pelaksanaan pengawal berasaskan logik kabur untuk robot penghindaran halangan. Beberapa teknik, pengawal dan simulasi telah diterapkan untuk memenuhi kehendak robot. Robot dapat berinteraksi dengan persekitaran yang tidak diketahui menggunakan strategi reaktif yang ditentukan oleh perintah dari logik kabur. Fuzzy Logic Controllers (FLC) adalah teknik pintar yang terbukti menjadi salah satu pengawal yang paling dipercayai yang sesuai untuk sistem bukan linier. Reka bentuk pengawal logik kabur berdasarkan peraturan untuk navigasi robot dan menghalang penghindaran dalam persekitaran yang berantakan, berdasarkan kaedah kabur jenis Mamdani. Dengan menggunakan jenis Mamdani ini dalam pengawal kabur, data latihan pemetaan input dan output dikumpulkan. Pengawal penghindaran halangan adalah sistem dua input dan satu output. Input adalah dua ukuran jarak jarak rintangan, dan output adalah halaju kedua roda. Dalam projek ini, tiga keahlian yang mempunyai dua anggota trapezoid. Pemboleh ubah linguistik seperti "Dekat", "Sederhana" dan "Jauh" diambil untuk fungsi tiga keahlian. Teknik ini menghasilkan kawalan manuver sudut arah yang sesuai terhadap kenderaan autonomi yang digunakan oleh robot untuk mencapai keselamatan tujuannya tanpa perlanggaran di persekitaran yang berantakan. ROS memudahkan penyediaan persekitaran yang sesuai untuk menguji pengawal yang dirancang. Ia menggunakan kerangka menggunakan konsep ROS sehingga pengawal pelacak lintasan dapat dilaksanakan pada robot mudah alih yang didayakan ROS. Model navigasi robot dibina di MATLAB. Hasil simulasi menunjukkan kaedah dapat digunakan untuk robot bergerak beroda yang bergerak di lingkungan yang berantakan dengan banyak rintangan. Ia dapat mengesan rintangan hingga 3 cm hingga 300 cm. Setelah robot mengelakkan halangan, robot akan menjaga persekitaran baru. Berdasarkan isyarat ini, pengawal mengawal halaju roda kiri dan kanan sehingga menjadikan robot bergerak maju dan berpusing pada masa yang sama. Kejayaan kawalan navigasi robot mudah alih banyak bergantung pada ketepatan pengukuran mutlak kedudukannya, jarak rintangan, jarak gol dan halaju. Orientasi jarak rintangan yang berbeza juga diuji dan tindak balas robot berfungsi seperti yang diharapkan.

ABSTRACT

With the progresses of innovation, versatile robots are getting to be prevalent. Agreeing to Karthiga (2014) created nations get the help of portable robots to protect people in fiasco region and this has been demonstrated as a profitable strategy to dispense with human blunder. In such occasions robot are required to explore in antagonistic situations such as collapsing buildings or ranges influenced fire. This thesis presents essentially mark the representation and usage of fuzzy logic-based controller for obstacles avoidance robot. A few strategies, controller and simulation are connected for carry off the prerequisites of the mobile robot. The robot competent to associated for obscure surroundings employing receptive technique decided with commands from fuzzy logic. Fuzzy Logic Controllers (FLC) is a shrewdly strategy demonstrates with foremost dependable controllers that suit well for nonlinear system. The design of a rule based fuzzy-logic controller for robot route with obstacles avoidance in unknown surroundings, based on the Mamdani type fuzzy strategy. Utilizing this Mamdani-type within the fuzzy controller, information of input and output mapping is gathered. The obstacle avoidance controller may be a two input and one output framework. The inputs are position error and angular error, and the outputs are the speeds of the two wheels. For the FLC, three sorts of membership functions are taking account for. Linguistic variables such as “Near”, “Medium” and “Far” are taken for three-membership work. This strategy creates reasonable heading point navigate control for independent automobile that utilized by the robot to get to its goal point without any crash in unknown surroundings. ROS encourages significantly setting-up appropriate surroundings to test the designed controller. Its employments a system utilizing ROS concepts so that a direction following controller can be executed on any ROS-enabled mobile robot. The robot route framework and model are developed in MATLAB and ROS. Simulation comes about appear the strategy can be utilized for wheeled mobile robot moving in cluttered surroundings of many hurdles. It can identify an obstacle up to 3 cm to 300 cm. When robot avoids a hurdle, the robot will detect unused surroundings. From this signal, the controller controls the speed of left and right wheels in this way making the robot to move forward and turning at the same time. The success of mobile robot route control depends for the most part on the exactness of outright estimations of its position, goal distance and speed. Diverse obstacles distance moreover been tried and the robot response is working as anticipated.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

THIS PAGE IS INTENTIONALLY LEFT BLANK

THIS PAGE IS INTENTIONALLY LEFT BLANK

CHAPTER 1

INTRODUCTION

1.1 Project background

In recent a long time, analysis and application utilizing not relating to strategies of computing such as fuzzy logic, developmental computation, and neural systems illustrated the advantage and capability this standard in favor of cleverly control of complex framework. Besides that, fuzzy logic demonstrated a helpful apparatus with taking care of genuine world vulnerability and information representation.

Fuzzy logic-based approach is effectively to handle nonlinearity, vulnerability and complexity framework lately. Fuzzy logic system could be an incredible tool to explore mobile robots in a known or obscure environment without colliding with any inactive or energetic deterrent. ROS (Robot Operating Systems) encourages significantly setting-up appropriate environment to test the outlined controller. ROS concepts are utilized, for the trajectory tracking controller controller executed on any ROS-enabled mobile robot. Robot route model communicates with the robot through ROS[1].

The reason of this project is to aim and create an obstacle avoidance robot that utilizing fuzzy control system utilizing MATLAB Simulink and ROS for the simulation as well as experimental analysis. The analysis will primarily be focusing on the position and following mistake examination on a few versatile robot movements and natural conditions. The critical commitment anticipated for the project is an ideal independent control system which can consequently avoid the unknown objects.

1.2 Problem statements

Different areas of manufacturing are utilized mobile robot or other application and other industries are begin to think around mobile robot in their applications seeing enhancement for innovation recently. One of the improvement innovations is obstacles avoidance for mobile robot in an obscure environment. Obstacle’s avoidance system delivers security mobile robot and obstacles. It will lead to halt any costing for repair and without increment taken a toll for support or repair a mobile robot. Mobile robot without obstacle avoidance is planning of risky to obstacles and mobile robot. In basic circumstance, it will create a mishap as it were created crucial misfortunes. For the superior condition of the portable robot to the conclusion client, plan robot navigation system utilizing MATLAB Simulink and ROS based on is imperative to unravel this problem[2].

Mobile robot may be a sort of robot that can move openly since it is prepared with movement components such as wheels or legs. In directing its movement, a mobile robot is prepared with a navigation system so that it can avoid obstacle’s. Without navigation system prepared inside the mobile robot, mobile robot cannot have capability to avoid the obstacles. In arrange mobile robot to avoid the obstacle’s, plan and execution of a wheeled mobile robot utilizing fuzzy logic standards with Mamdani's fuzzy inference system is actualized within the system[3].

The concept of obstacle’s avoidance in robot way arranging clarifies capacity for robot to escape from colliding with any obstacle in dynamic surroundings. Keep away from obstacles could be an exceptionally troublesome errand since the mobile robot is dynamic and nonlinear. The adjustment for mobile robot’s states will affect adjustment of its obstacle’s states. In this manner, it is vital to examine the execution of the position and following error investigation on mobile robot motions and its environment conditions in arrange to create beyond any doubt for development of the robot is smooth and will not collide with the obstacles[4].

1.3 Project objectives

The goals for this project are:

1. To design robot navigation system using ROS that can avoid unknown or unexpected obstacles in an unknown environment.

2. To design Mamdani fuzzy logic controller and MATLAB Simulink for the obstacle avoidance robot.
3. To analyse the performance of the position on mobile robot motions and environment conditions.

1.4 Project scopes

In arrange to attain the targets of the project, there are a few scopes had been sketched out. The scopes of the project are:

1. Develop the simulation for the system.
2. MATLAB Simulink and ROS are used to detect the obstacle for the robot.
3. Fuzzy logic inference method used in this project is Mamdani Fuzzy Logic.
4. The robot able to avoid the obstacles on the flat surface, from any starting point.

1.5 Thesis outline

This thesis is partitioned into five chapters. Chapter 1 the introduction, Chapter 2 is committed for study writing on obstacles avoidance mobile robot. Chapter 3 talks about technique of an extend for bargains with the analysis of a proposed fuzzy logic technique with three sorts of membership functions to avoid the obstacles in obscure surroundings. Chapter 4 bargains with how the fuzzy logic controller taking the choice to avoid the obstacles and the simulation result of ROS and MATLAB. This chapter too comprises, conduct of the robot amid avoids obstacles and simulation results. Chapter 5 conclude of the extend with talk about thoughts for encourage work as suggested.

1.6 Current technology using mobile robot

Figure 1.1 The soccer robots of NEUIslenders group that utilize the fuzzy controller in a football game of RoboCup competition. The obtained experimental comes about fulfill the application of fuzzy controllers in control of omnidirectional robots[5].

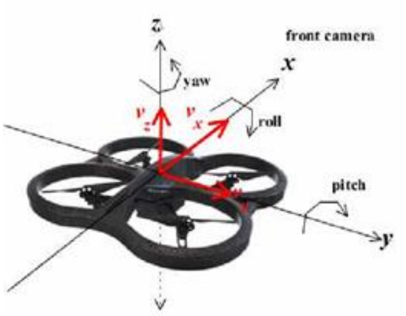


Figure 1.2 Fuzzy Logic Controller (FLC) is actualized within the AR.Drone quadrotor in arrange to form it take after a given trajectory reference. The distance between the position and angle of the AR.Drone to the reference point is utilized as the input of FLC[6].

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter is approximately the writing survey based on the existing framework or strategies that previously accepted by analysts. This incorporates considers on the term paper with respect to Local look optimization (Heuristic). The primary objective of literature review is:

- It shows the writing in an organize way.
- It studies the proficient in your chosen range of study.
- Its synthesis the data in that writing into a summary.
- It basically analyses the data accumulated by recognizing gaps in current information by appearing restrictions of hypotheses and point of see by defining zones for assist investigate and looking into area.

In arrange to plan and create an obstacles avoidance robot by utilizing fuzzy logic controller, broad investigate on the fuzzy controller that satisfied. This section will talk about previous research that fulfilled by other researchers within the same area.Description of previous method

Qing-yong, had created behaviours-based mobile robot route within obscure surroundings with utilizing fuzzy logic. Four fundamental practices utilized within mobile bot navigation that are goal behaviour, obstacle avoidance conduct, tracking behaviour, and deadlock disarming behaviour. These essential practices are executed with utilizing fuzzy logic controller. A behaviour controller is plan for decide supervising activity for a mobile robot. The simulation test comes about appeared that the proposed engineering empowers the mobile robot to securely accomplish the objective without colliding [7].

Limen, created a latest fuzzy intelligent obstacle avoidance control technique for wheeled mobile robot that are two fuzzy logic controllers and a brilliantly coordinator. Fuzzy obstacle controller will produce impediment avoidance order concurring towards goal introduction data with impediment data as robot recognizing impediments by means to it sensors. Brilliantly coordinator is planned for facilitate run-to-goal fuzzy controller and fuzzy obstacle and produce best robot control commands [8].

Mobile robot control utilizing type-2 fuzzy logic framework were created by Pisit and Supachai. 2-type fuzzy sets depicted by membership functions of fuzzy and output processor 2-type fuzzy will have: type-reduction and defuzzification. The 2-type fuzzy logic controller are handling information output to manage the course for mobile robot development. The behaviour-based control were obstacle avoidance and hallway taking after. [9] .

Capability of a robot to avoid collision with unexpected and energetic hurdles whereas go to the goal or following the way could be an imperative assignment in independent route. Route plan is according to worldwide route preparing and local way planning.

In worldwide route planning, data about the hurdles of worldwide show of surrounding are accessible for the most part Configuration space, Street map, Voronoi diagram and Potential field strategies were utilized for arrange no hurdles way towards a target. Be that as it may, in genuine surroundings a dependable map of hurdles, precise demonstrate of surroundings and exact sensor information are inaccessible because of instabilities surroundings. Whereas an effective way can stay substantial yet reaction unanticipated or energetic hurdles, there are fundamental a robot can modify its path online. Beside that, Fuzzy logic may give vigorous or dependable strategies managing of uncertain input. [10].

Zavlangas created a receptive route strategy for mobile robot utilizing fuzzy logic. Fuzzy rule-base creates inciting order for urge collision free movements of surroundings area. The fuzzy logic too gives a movable straightforward framework with the rules or physically. Seraji and Howard created the behaviour-based navigation strategy on challenging terrain utilizing fuzzy logic. A route consists of three behaviours. Nearby obstacle avoidance behaviour was comprising the set of fuzzy logic rule statements that can produces the robot velocity by hurdles gap.[11].

2.2 ROS (Robot Operating System)

2.2.1 Introduction

ROS is an open-source robot working principles. Bunch of program libraries and instruments were assisted to construct robot applications functioning over an assortment of robotic stages. Original created in 2007 at the Stanford Manufactured Insights Research facility and development proceeded at Willow Carport. Since 2013 overseen by OSRF (Open-Source Mechanical autonomy Establishment)[12]. The operating system side, which provides standard operating system services such as:

- Hardware abstraction
- Low-level device control
- Implementation of commonly used functionality
- Message-passing between processes
- Package management

A suite of user contributed packages that implement common robot functionality such as SLAM, Gazebo 11, planning, perception, vision, manipulation and etc.

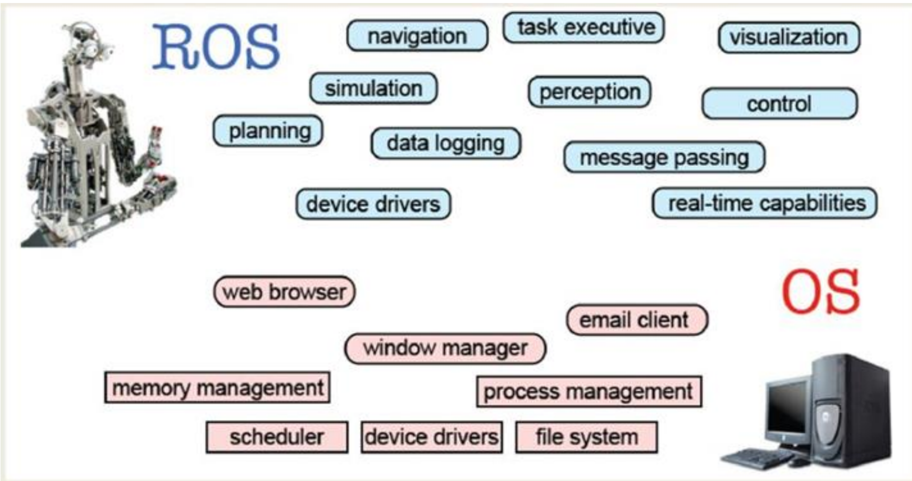


Figure 2.1 ROS main feature

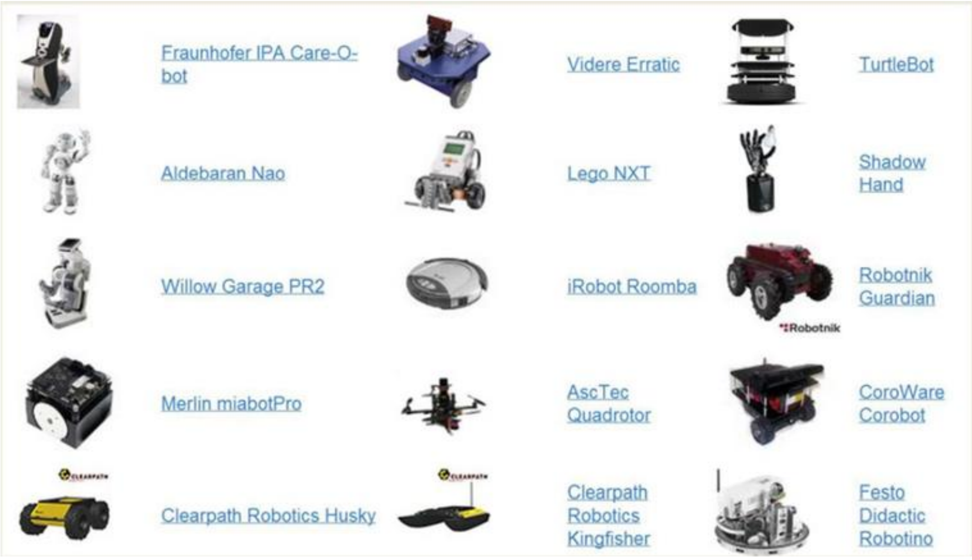


Figure 2.2 Robots using ROS

2.2.2 ROS Nodes

Single- purposed executable programs. Such an example sensor drivers, actuator drivers, mapper, planner, UI and etc. Individually compiled, executed and managed. Nodes are written using ROS client library such as roscpp C++ library and rospy python client library. Nodes can publish or subscribe to a Topic. Nodes can also provide or use a Service [13] .

2.2.3 ROS Topics

A topic is a name for a stream of messages with a defined type. Such an example data from a laser range-finder might be sent on a topic called scan, with a message type of Laser Scan. Nodes communicate with each other by publishing messages to topics [13] .

2.2.4 ROS Messages

Strictly typed data structures for inter-node communication. For example, geometry_msgs/Twist is used to express velocity commands: Vector3 linear and Vector3 angular. Vector3 is another message type composed of float64 x, float64 y and float64 z [13].

2.2.5 ROS Packages

Software in ROS is organized in packages. A package contains one or more nodes and provides a ROS interface. Most of ROS packages are hosted in GitHub [13] .

Figure 2.3 ROS Package

2.2.6 ROS Environment

ROS relies on the notion of combining spaces using the shell environment. This makes developing against different versions of ROS or against different sets of packages. After install ROS setup. *sh files in '/opt/ros/<distro>/' and also source it $source/opt/ros/indigo/setup.bash [13].

2.2.7 Advantages of using ROS

The perfect objectives of utilizing ROS are to permit simple run for apparatuses counting visualization of robot kinematics and sensor information, path planning and perception algorithms, and low-level drivers to use at sensor..

Benefits of ROS:

1. Peer-to-peer communication
2. Free and open-source
3. Multi-language

Besides that, complex robotic frameworks with different joins can has different hardware of computers (to do various assignments) associated using network. At ROS, a peer-to-peer architecture are together for buffering framework and search for framework (the title benefit called ‘master’ in ROS), empowers every single part for discourse straightforwardly to another part. Other than that, for an open-source platform gives a remain capacities given from numerous another ROS clients. Their code is provided at repositories as "stacks".[14] .

2.2.8 Method research on ROS

ROS Route: concepts and tutorial are around the most theoretical concepts and clarify utilize of ROS Navigation Stack. Typically, an effective tool compartment to way arranging with Simultaneous localization and mapping (Slam) [15].]. A Framework for Quality Assessment of ROS Repositories includes optimality with HAROS, a plug-in-driven system for source code examination of ROS repositories. HAROS’s main center was help for revealing potential flaws with absconds in ROS frameworks [16]. Verification of system-wide security properties of ROS applications bargains with strategy to naturally confirm security properties of ROS-based applications in inactive hour [17]. ROS: an open-source Robot Operating System bargains with to the existing robots program framework and briefly diagram a few of the accessible application computer program which employments ROS [12]. Simulation Environment for Mobile Robots Testing Using ROS and Gazebo is reasonable to autonomous route assignment and 3D-mapping simulation.[18].

No.	Methods	Authors	Disadvantages
1	ROS Navigation: concepts and tutorial	Rodrigo Longhi Guimaraes Andr´e Schneider de Oliveira Joao Fabro	Navigation too complex
2		André Santos	

	A Framework for Quality Assessment of ROS Repositories	Alcino Cunha Nuno Macedo	- Not have formal verification of functional properties and exploring model-based techniques
3	Verification of system-wide safety properties of ROS applications	Renato Carvalho Alcino Cunha Nuno Macedo	- Did not relying on run-time analyses - Lack of scalability of the approach
4	ROS: an open-source Robot Operating System	Morgan Quigley Brian Gerkey Ken Conley	- Complicated graph of hierarchical multi-robot control system - Research settings not enough explanation
5	Simulation Environment for Mobile Robots Testing Using ROS and Gazebo	Kenta Takaya Toshinori Asai Valeri Kroumov Florentin Smarandache	- Cause very high load on the CPU laptop

Table 1 Method research on ROS

2.3 Fuzzy Control System

Latest project utilizing Fuzzy Decision-Making Controller could be sort of fuzzy logic controller (FLC). These sorts Fuzzy Logic are utilized to take over operation such an instance plant in control engineering terminology. Benefits from it are empowers control engineers for effortlessly execute control techniques that may utilized to individual administrator [19].

Components of FLC is inference engine with linguistic IF-Then rules which encrypt to behaviour from robots. In any case, most trouble for planning a fuzzy logic controller are proficient detailing fuzzy IF-Then rules. In case there are simple for deliver predecessor fuzzy rule base, there are in any case exceptionally troublesome to create the resulting area in absence of great information [20]. Figure 2.4 appears the Fuzzy Decision-Making controller consists of three steps:

1) Fuzzification: Changes over controller inputs into data that the inference mechanism can be effectively utilize to actuate and apply rules.
2) Rule base: A set of IF-Then rules which contains a fuzzy logic measurement of the expert's verbal portrayal of how to attain great control.
3) Defuzzification: This converts the conclusions of the interface mechanism into real inputs for the process.

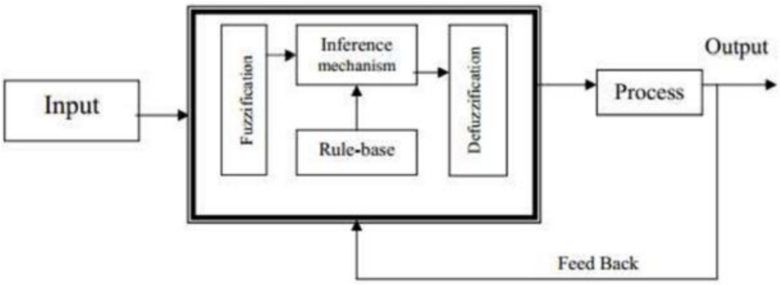


Figure 2.4 Fuzzy Logic Controller

The membership function has the magnitude of participation of each input and is represented in graphical. The roles of membership functions are to:

- Each of the input that are processed are given certain magnitude value.
- Functional overlap between inputs is to be defined.

- Output response are to be determined

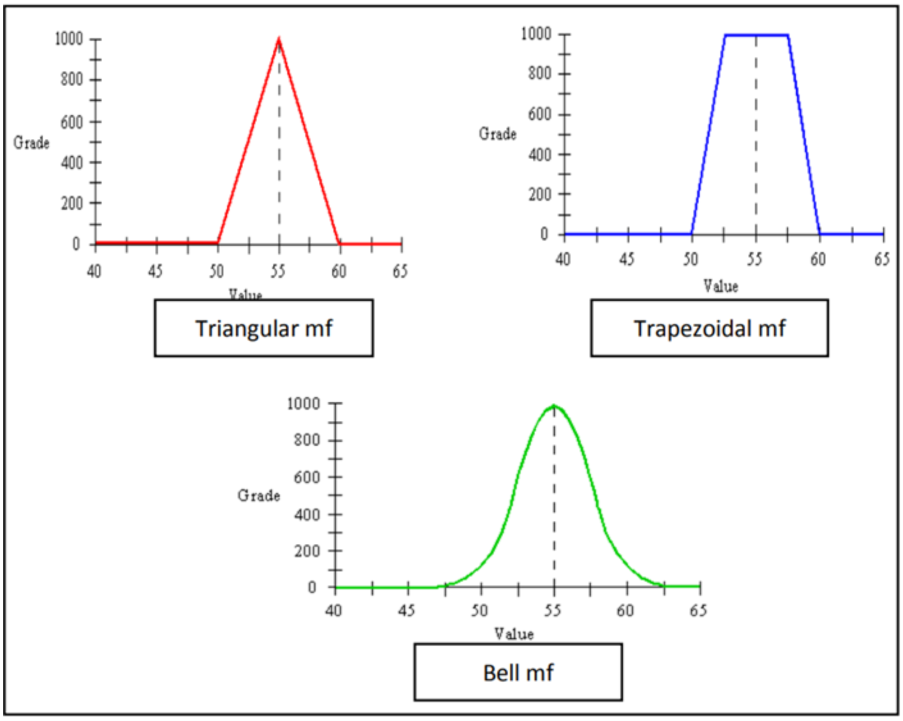


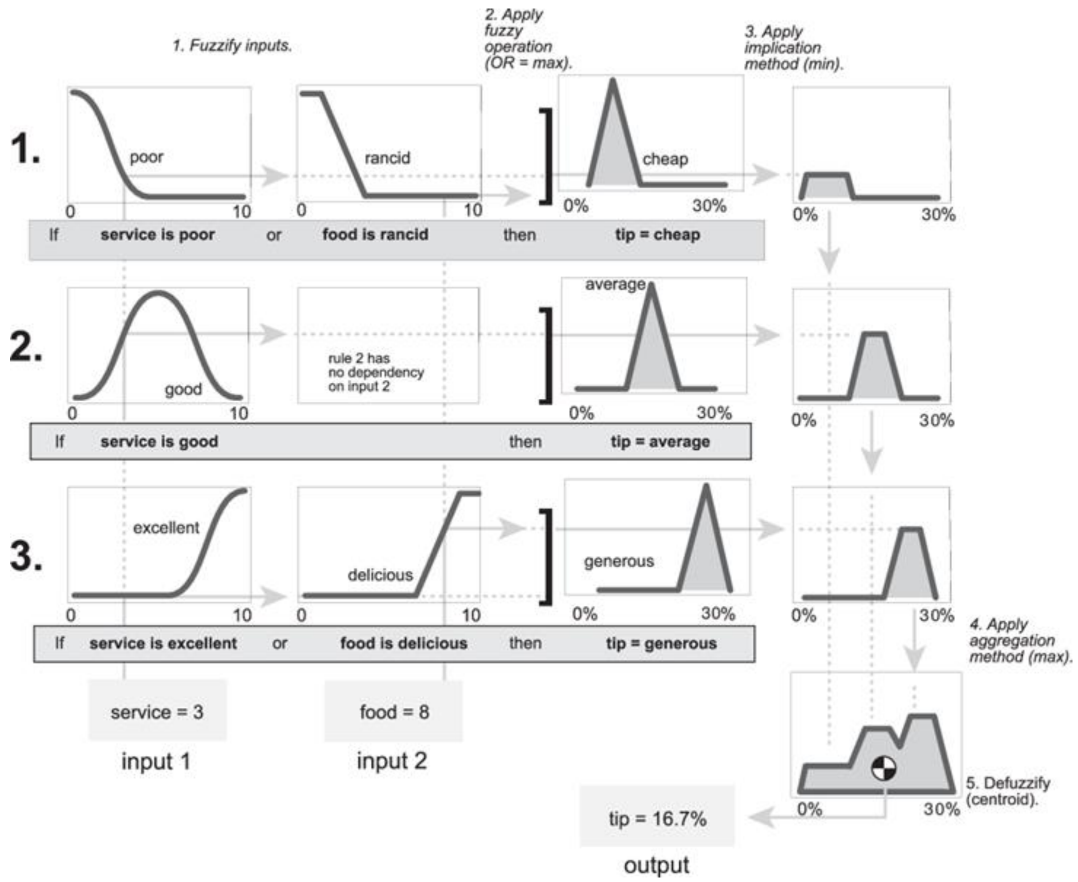
Figure 2.5 Various Types of Membership Functions

The rule utilizing input membership values as weighting components for control its affect between the fuzzy output sets with output conclusion. When capacities were concrete, expanded, and joint, they are defuzzified into a crisp output that be in charge of fuzzy structure. Addition to that, distinctive shapes memberships functions linked with each input and output response such an instance, triangular shape formed shown in Figure 2.5 [21].

Three primary fuzzy inference systems which is Mamdani, Sugeno, and Tsukamoto type. In this project, Mamdani system are apply.

Mamdani fuzzy frameworks were initially outlined to mimic the execution of man operators that controlling industrial processes (Mamdani 1974, 1976, 1977; Mamdani & Assilian 1975). A point was for conclude the operator's encounter into a set of (linguistic) IF-Then rules that may utilized by a machine to naturally manage this method [22].

Figure 2.6 The Example of Fuzzy Tipping Model Developed by Using Mamdani System



2.3.1 Method research of Fuzzy Logic

Robot navigation with obstacle avoidance in obscure surroundings are approximately the robot competent deterrents shirking in obscure environment through fuzzy logic controller [23]. MATLAB Simulation for Mobile Robot Navigation with Obstacles in Cluttered Environment Utilizing Minimum Rule Based Fuzzy Logic Controller includes the least rule based fuzzy-logic controller for robot route with obstacles refrain from [24]. Fuzzy Rationale Based Control for Independent Mobile Robot Route bargains with execution of a direction following controller utilizing fuzzy logic for mobile robot to explore indoor situations [25]. The Integration of Fuzzy Logic System for Obstacle Avoidance Behaviour of Mobile Robot bargains with mobile robot detecting its area beneath uncertain environment, real-time way controlling its controlling point and speed to reach target area [26]. Wireless Vision-Based Fuzzy Controllers for Moving Object Tracking Using a Quadcopter is suitable to present a visual servoing system for detecting and tracking a moving object using an unmanned aerial vehicle [27].

No.	Methods	Authors	Disadvantages
1	Robot Navigation with Obstacle Avoidance in Unknown Environment	Neerenda Kumar Zoltain Vamossy	Navigation model does not have any prior information about navigation environment
2	MATLAB Simulation for Mobile Robot Navigation with Hurdles in Cluttered Environment Using Minimum Rule Based Fuzzy Logic Controller	Anish Pandey Dayal Ramakrushna Parhi	Do not navigate the robot within dynamic environments simulation
3	Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation	Hajer Omrane Mohamed Slim Masmudi Mohamed Masmoudi	The detail about the MATLAB simulator did not explain well
4	The Integration of Fuzzy Logic System for Obstacle Avoidance Behaviour of Mobile Robot	Nur Ilyana Anwar Apandi Aerun Martin	Did not apply it in real world environment with robot
5	Wireless Vision-Based Fuzzy Controllers for Moving Object Tracking Using a Quadcopter	Mohammad Algabri Hassan Mathkour	Lack of optimization proposed system of the fuzzy logic controllers

		Mohamed Amine Mekhtiche	
		Mansour Alsulaiman	

Table 2 Method research on Fuzzy Logic

2.4 Artificial Intelligence

Lately, Artificial Intelligence Methods are broadly utilized in ROS and Fuzzy Logic. This strategy is the science of making intelligent computer program. There are six distinctive sorts beneath this procedure as outlined in Table 2.3. Each sort has it possess focal points of fathoming distinctive sorts of congestion issues. In this paper, research done is centering on Fuzzy Logic strategy. The point-by-point discourse is included in following subtopic[28].

Methods	Different Types	Descriptions
Artificial Intelligence Methods	Fuzzy Logic Method	Using Fuzzy Set Theory dealing with approximation rather than precision
	Artificial Neural Network	An interconnected group of neurons that uses connectionist approach to computation
	Genetic Algorithm Method	Uses theory of survival of fittest
	Evolutionary Programming	Based on metaheuristic optimization algorithm
	Ant Colony Optimization	Based on the idea of ant foraging by pheromone communication to make path
	Particle Swarm Optimization	Based on the ideas of social behaviour of organisms (animal flocking and fish schooling)

Table 3 Different types of Artificial Intelligence Methods

2.4.1 Artificial Intelligence in ROS application

Intelligent Inn ROS-based Service Robot is approximately conceiving on the intelligence hotel robot, which disentangle the check-in process [29]. ROS Based Multi-sensor Route of Intelligent Wheelchair includes ROS (Robot Working Framework) based different-sensor route of intelligent wheelchair which offer assistance the elderly and debilitated individuals [30]. Utilizing Ros for Rural Robotics bargains with utilizing the ROS middleware for improvement of agricultural robots [31]. Plan of an Independent Mobile Robot Based on ROS bargains with autonomous mobile robot (AMR) adjusted of robot working framework (ROS) that displayed with take account for the equipment design and electronic communication conventions [32]. Control of the Mobile Robots with ROS in Robotics Courses is usage of mobile robots programming handle with Robot Operating Framework (ROS) in understudy mechanical technology courses [33]

No.	Methods	Authors	Disadvantages
1	Intelligent Hotel ROS-based Service Robot	Yanyu Zhang Xiu Wang Xuan Wu	Did not reduce human labour, did not have robotic arm to press the correct floor number inside the elevator
2	ROS Based Multi-sensor Navigation of Intelligent Wheelchair	Ruijiao Li Mohammadreza A Oskoei	No navigation of the wrap of doorway passing navigator in ROS
3	Using Ros for Agricultural Robotics	Ruud Barth Yael Edan Thomas Buschmann	Does not support real-time response on the application
4		Murat Köseoğlu Orkan Murat Çelik	

	Design of an Autonomous Mobile Robot Based on ROS	Ömer Pektaş	Less obstacle avoidance for robot and mapping errors
5	Control of the Mobile Robots with ROS in Robotics Courses	Khassanov Alisher Krupenkin Alexander Borgul Alexandr	Not enough nodes and tools that developed to a student

Table 4 Method research of Artificial Intelligence in ROS application

2.5 Summary

This research review's reason is to assist the reader get it diverse viewpoints postured by the investigate on the Fuzzy Based Mobile Robot Route on Motion Tracking Using Ros. This is often critical since numerous individuals have a diverse approach for avoidance obstacles navigation framework of mobile robot application utilizing ROS and MATLAB Simulink based on Fuzzy Logic, frequently not realizing there are other focuses of view. There has been much investigate and talk conducted on these conclusions of the ROS and Fuzzy Logic, counting its application, calculation, programming coming about in python, and clashes with a lack of identity. More inquire about and simulation are required to pick up a much better understanding of Fuzzy Logic and ROS. It is imperative to conduct more considers on the comes about and reasons of the navigation of mobile robot is distinctive when Fuzzy Logic is connected.

CHAPTER 3

METHODOLOGY

3.1 Research Flowchart

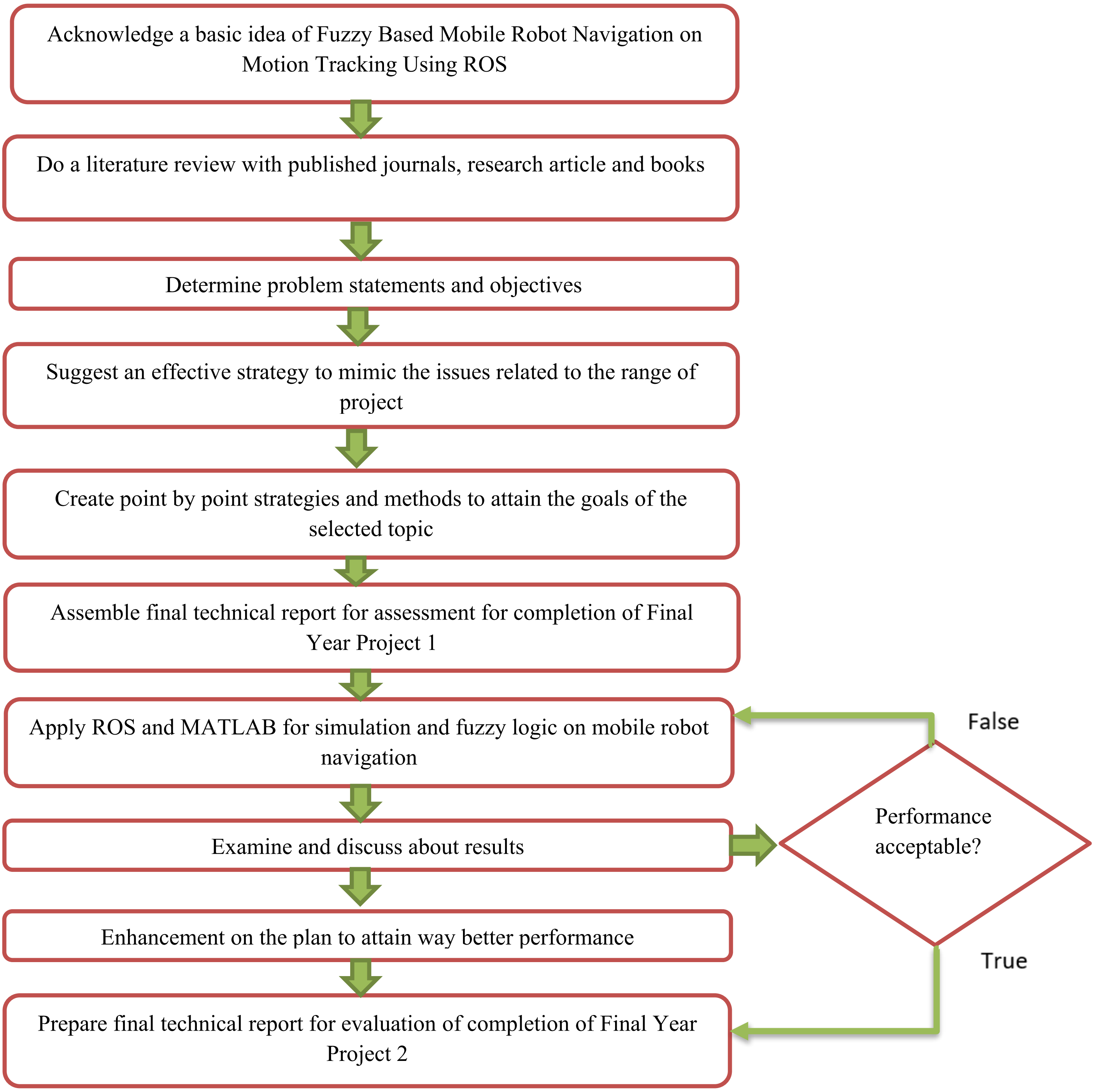
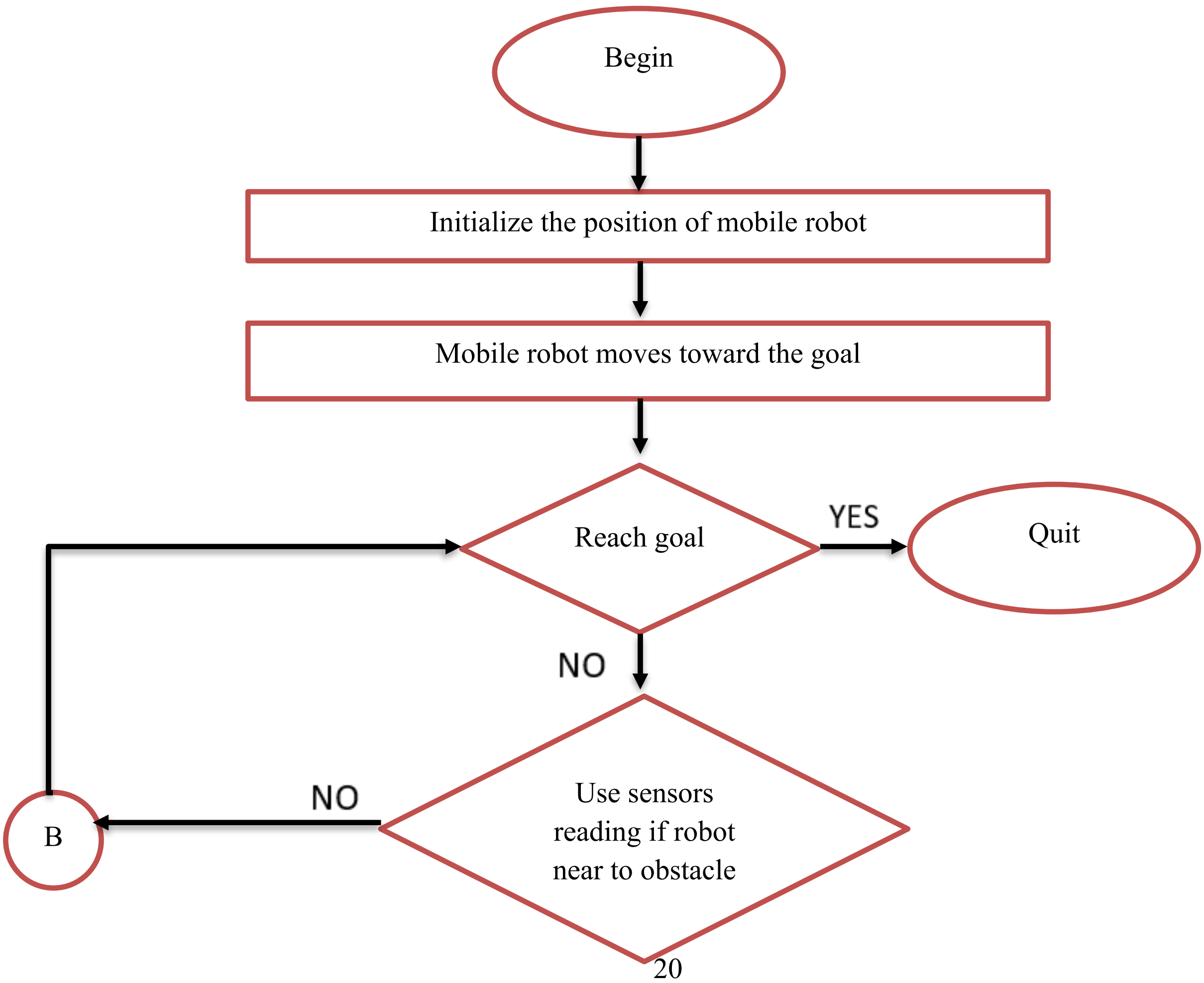


Figure 3.1 Research Flowchart

Based on the flowchart above, recognize comprehensively the elemental concept for Fuzzy Based Mobile Robot Navigation on Motion Tracking Using ROS is the first step in this research project. In this part, the understanding of the Fuzzy Logic fundamental concept must be good. Next, conducted the literature review based on published journals, research article and books must be performed. Search for an article and journal is must do for more deep understanding for this project. After that, determine problem statements and objectives for a project.

In order to do the project, the objectives and problem statement is must identify to easier to identifies the issue throughout the research and achieve the aim throughout the research. Other than that, suggest an effective strategy to mimic the issues related area of project must come up in order to do the project. Besides, the create point by point techniques and methods to realize the goals of the chosen topic.

3.2 Mobile robot navigation



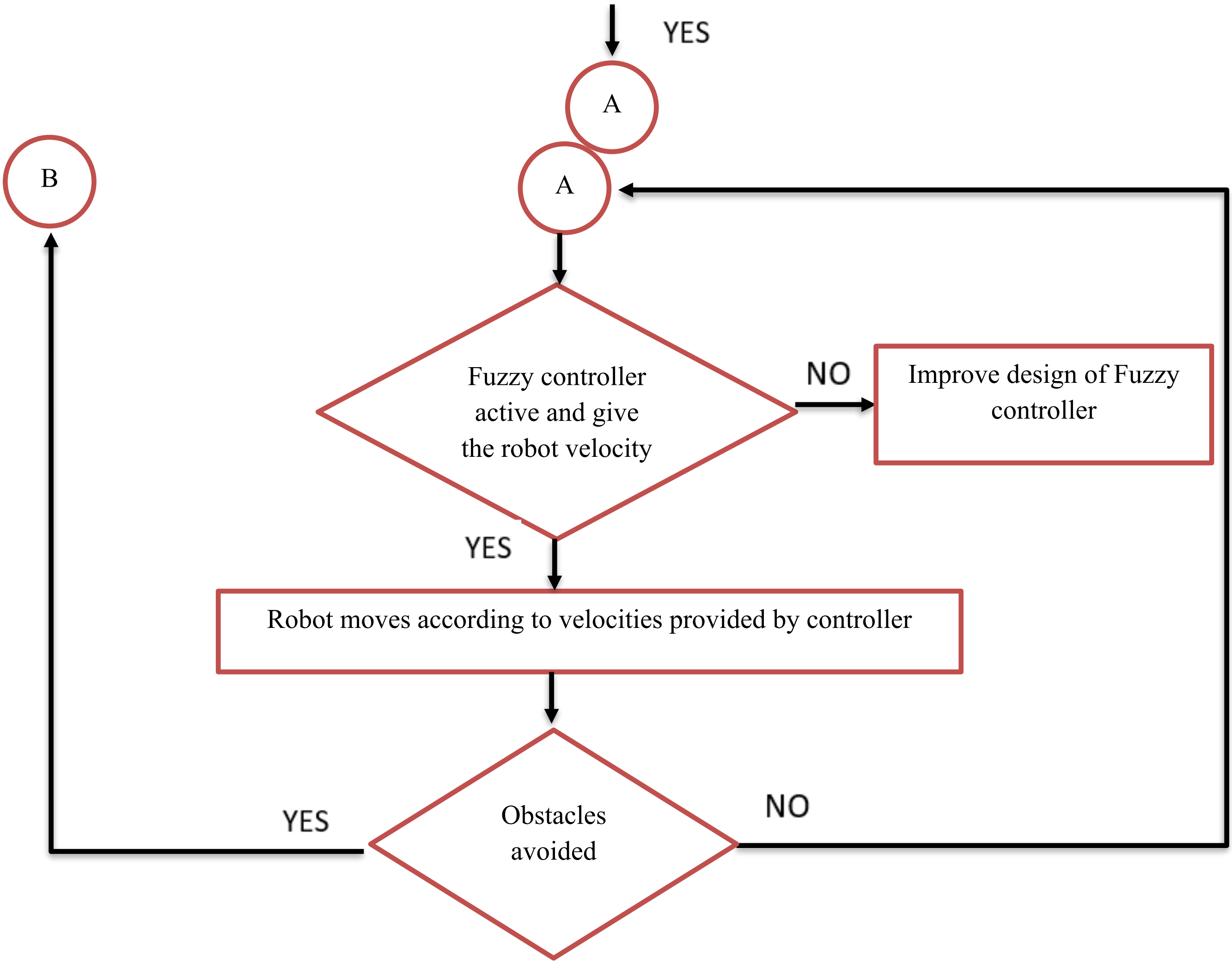


Figure 3.2 Flow chart of mobile robot navigation

Figure 3.2 shows the flow chart of mobile robot navigation. Firstly, initializes the position of mobile robot. Then the mobile robot moves toward the goal. If the mobile robot reach goal it will quit and if no, the program will examine all sensors value that connected within robot. At that point the distance value of impediment sensor was calculated. If no sensor is being read, it will quit. The program gets extending information from sensors to do fuzzy logic operations. If no fuzzy logic operation is conducted, the improving for design of Fuzzy controller will be made. Robot at that point do the turning to dodge obstacle depend for output of the fuzzy logic program. When the turning process is complete, the sensor identifies or examined once more for another input. Robot will move accordingly to velocities provided by Fuzzy controller. If there is obstacle again, the avoiding process still running until there is no obstacle. In case there's obstacle once more, the dodging process still go on until there's no obstacle. In case no obstacle is identified, the robot goes straight in quick condition. After each movement, at that point the sensor is examined once more to proceed the following movement process. Sensor reading shows the mathematical rule is connected to discover the obstacle distance and obstacle difference value [34].

3.3 Proposed solution

This is flow chart for the project. Based on flow chart, direction and heading of the project can be seen clearly. These are often an assist to makes a difference creating and making this project effective.

Research On Topic	• Conduct research on Fuzzy Based Mobile Robot Navigation on Motion Tracking Using Ros on Fuzzy Logic Approach. • Research is done by referring journals, thesis, books, conference papers, technical reports, internet and interactive media (CD ROM).
Fuzzy Logic based Mobile Robot Navigation on Motion Tracking	• Familiarization with ROS and Fuzzy Logic Theory of its definition and its equations. • Understanding the applications of Fuzzy Logic on based Mobile Robot Navigation on Motion Tracking Using ROS.
Mobile Robot and Fuzzy Logic Simulation	• Simulate mobile robot using ROS and Fuzzy Logic to control the Mobile Robot to avoid unknown or unexpected obstacles in an unknown environment.
Analysis and Improvement of Design	• If simulation does not provide fulfill output, enhancement ought to be carried out. • Design improvement includes the changes of design modelled that's made earlier to urge a better performance.
Discussion on Results	• The execution of the mobile robot based on ROS and Fuzzy Logic approach on dodge unknown or unforeseen obstacles in an obscure environment. • This new approach ought to allow way better execution as compared with the ordinary approach.

Preparations of Final Technical Report 1 & 2	• Technical report which includes 5 chapters of Introduction, Literature Review, Methodology, Results and Discussions, and Conclusion will be prepared for evaluation.

Table 5 Proposed Solution Explanation

3.4 Tools

To accomplish this project, some tools are needed either for simulation or report writing purposes. From the image below, Microsoft word 2019 is used to do a report thesis for this project. Adobe reader is function to print the report or to review the thesis report in pdf format. MATLAB purposes to do a simulation of mobile robot with the application of fuzzy logic. Robot operating system (ROS) is used for build programs for robot based on ROS based on Linux ubuntu. Fuzzy lite is for better trajectory and

Figure 3.3 Microsoft Word 2019

obstacles avoidance. Python 3 is used for build a coding. Gazebo 11 is robot simulator that can simulate the robot in complex indoor and outdoor environments.

Figure 3.4 Adobe Reader

Figure 3.5 MATLAB

Figure 3.6 ROS (Robot Operating System)

Figure 3.7 Python 3

Figure 3.8 Gazebo 11

3.5 Mobile Robot in Catkin Workspace

1. The Turtlebot 3 is used as a mobile robot.

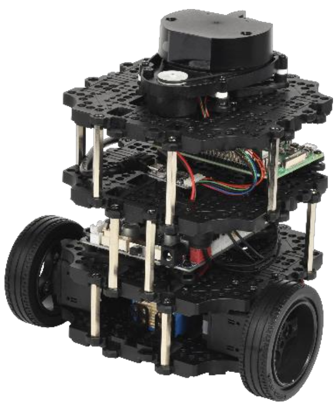


Figure 3.9 Turtlebot3

2. Then, run the catkin_make in the root of catkin workspace.

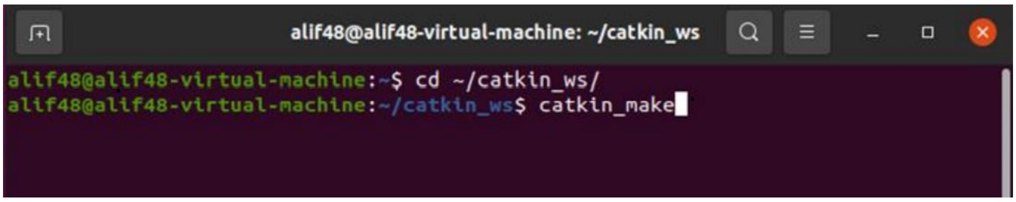


Figure 3.10 Run command catkin_make

3. Build and compile the file of catkin_ws

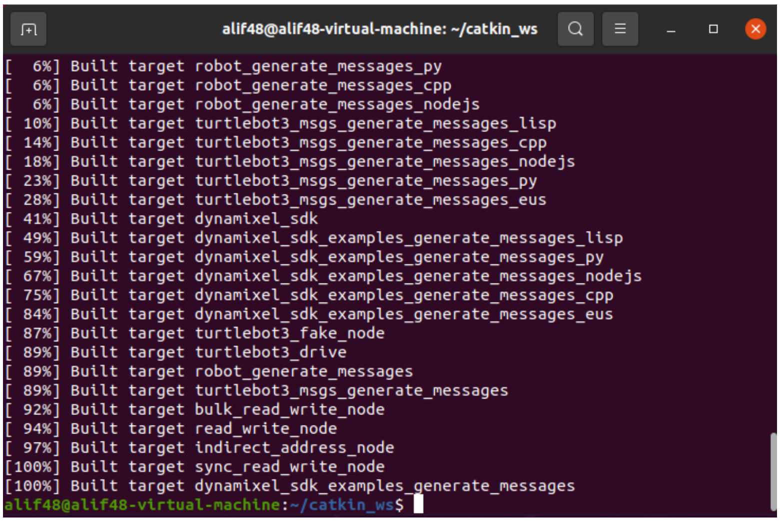


Figure 3.11 Build target of the package

3.5.1 Use command Roslaunch to launch the environment of robot simulator (Turtlebot3)

1. Use the command "roslaunch turtlebot3_gazebo turtlebot3 world. launch" to launch the environment of Turtlebot3 mobile robot using Gazebo 11 robot simulator.

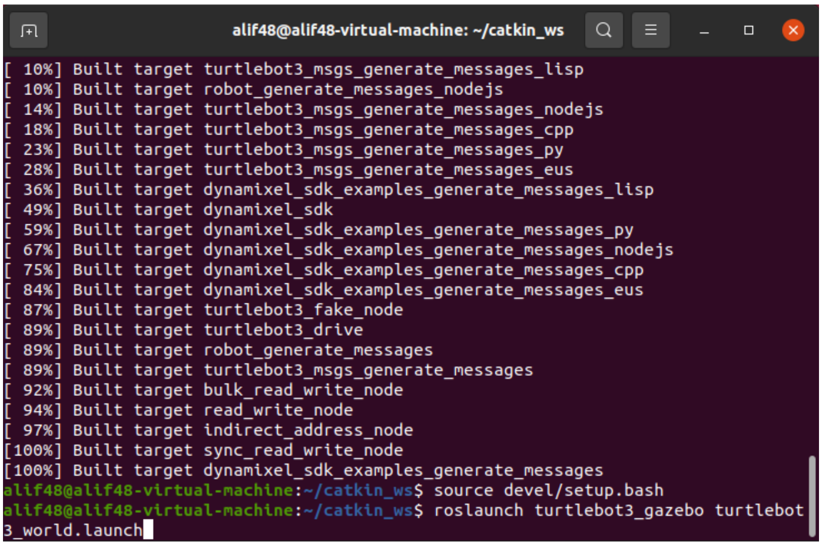


Figure 3.12 Command to launch the environment of Turtlebot3 on Gazebo 11

2. Use the command "rosrun obstacle-avoidance-turtlebot fuzzy.py" to run the python coding and to make the mobile robot move.

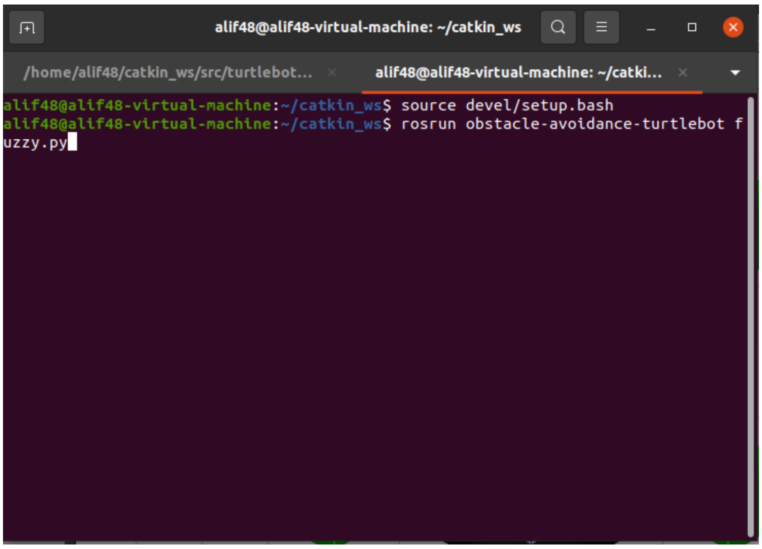


Figure 3.13 Command "rosrun obstacle-avoidance-turtlebot fuzzy.py

3.5.2 Use command Roslaunch to launch the environment of robot simulator (Rosbot)

1. Use the command "roslaunch rosbot_description rosbot_rviz_gmapping.launch" to launch the environment of Rosbot mobile robot using Gazebo 11 robot simulator.

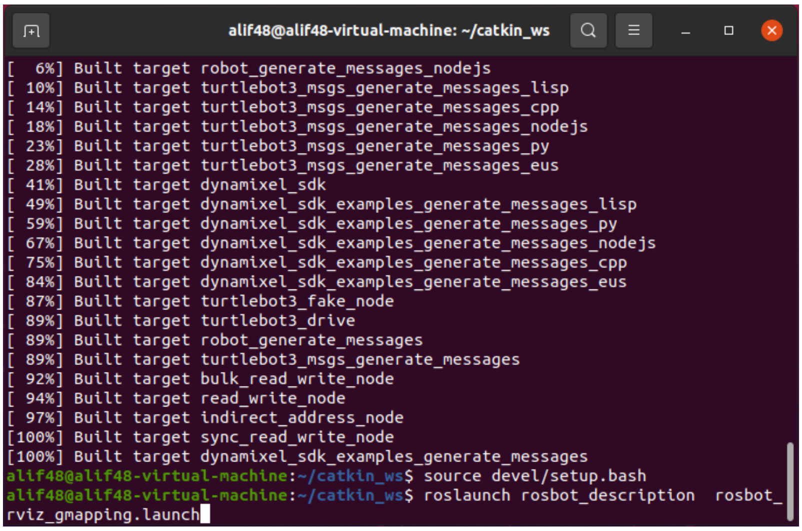
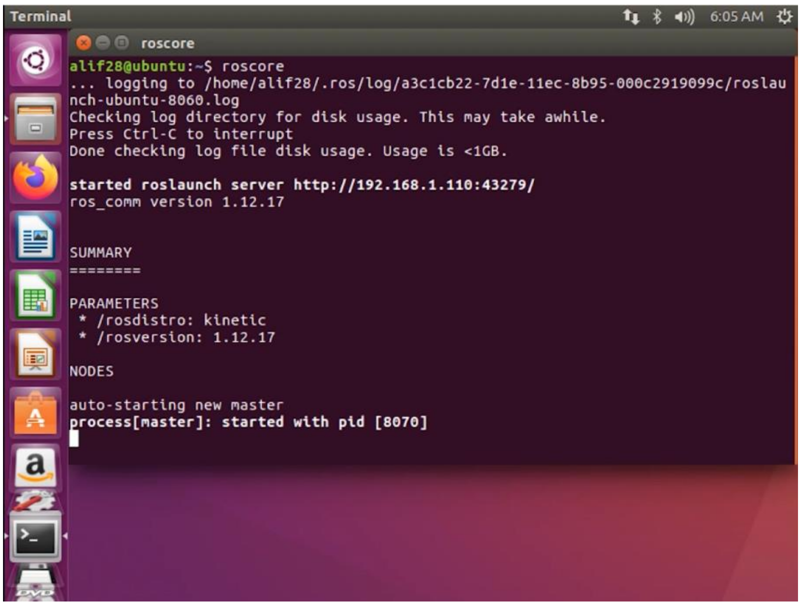


Figure 3.14 Command to launch the environement of Rosbot on Gazebo 11

3.6 Bringup TurtleBot from the remote pc

1. Run roscore from PC. The pc and the TurtleBot3 must be connect to the same ip address of wifi in order to connect to the TurtleBot3. In this case my laptop wifi ip address is 192.168.1.110 and the TurtleBot3 Resberry Pi ip address wifi is



192.168.1.112. The picture above

Figure 3.15 Command Roscore

2. Connect to Raspberry Pi with its IP address. Password default is turtlebot.

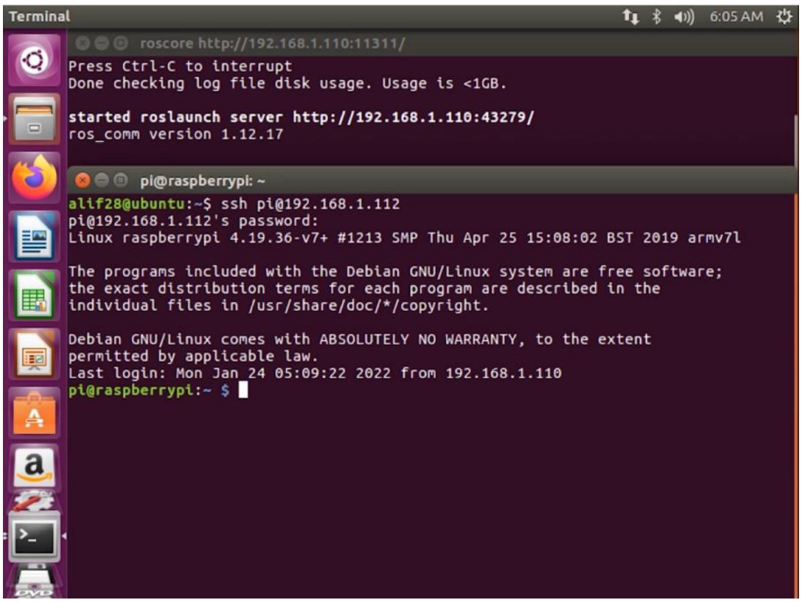


Figure 3.16 Connect to Rasberry Pi IP address

3. Bring up basic packages to start TurtleBot3 applications. Use the command “roslaunch turtlebot3_bringup turtlebot3_robot. launch”.

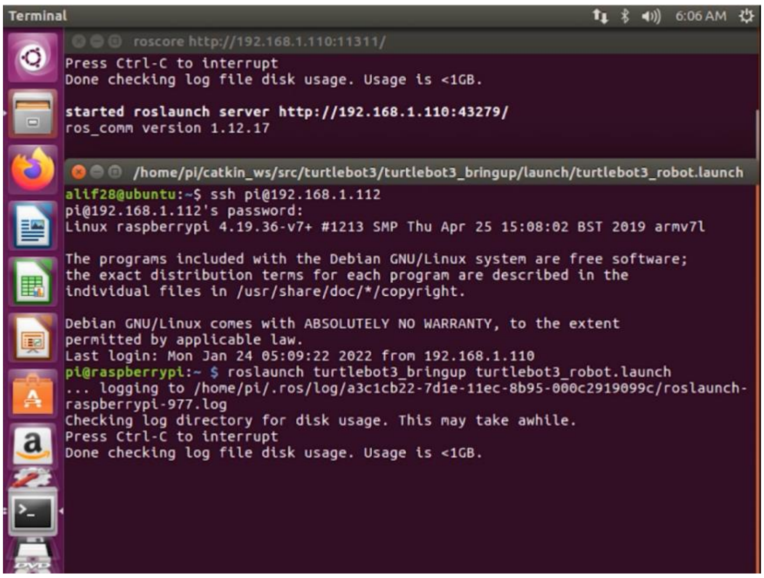


Figure 3.17 Bringup to start TurtleBot3 application

4. Successfully connected to the Rasberry Pi. If TurtleBot3 model is burger, the terminal will print above messages.

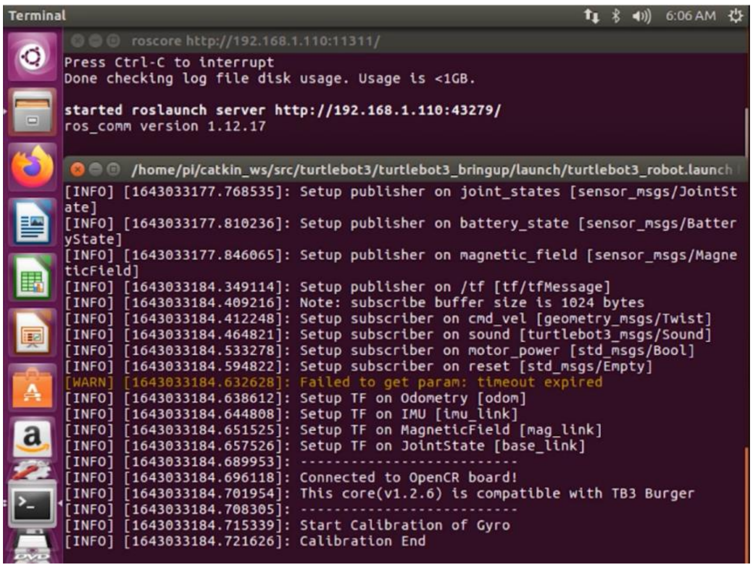


Figure 3.18 Terminal messages show successfully connected to the Rasberry Pi

5. Run the coding to move the TurtleBot3.

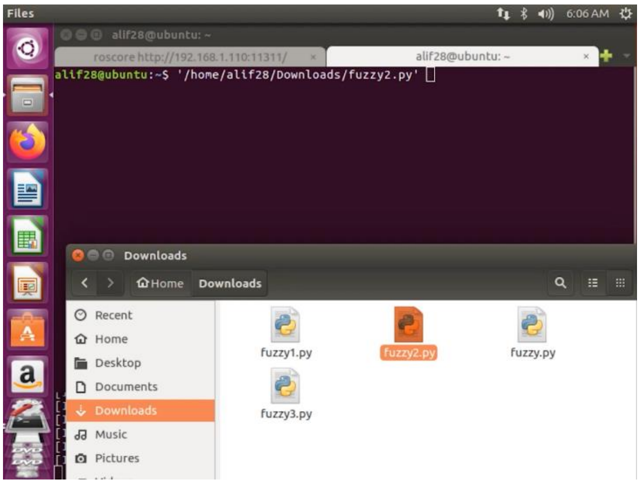


Figure 3.19 Drag the python coding to the terminal

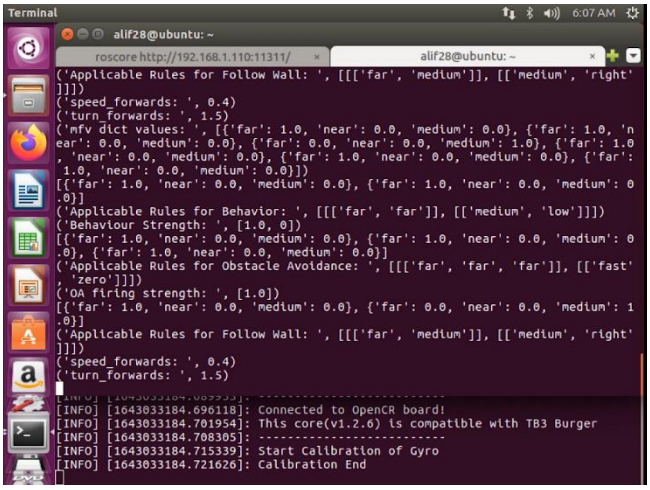
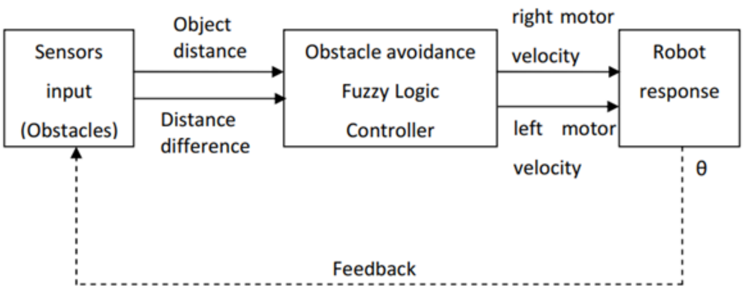


Figure 3.20 Coding is running

3.7 ROS/Design Simulation of Robot Using ROS

Fuzzy Logic Controller (FLC) planned to control the movement of the robot. There have two inputs which is Position Error and Angular Error of the robot. The one outputs are



left engine speed and right engine speed. Hence, the FLC is two inputs and one outputs framework[25].

Figure 3.21 Block diagram of fuzzy behaviour robot control architecture

The ROS interface provided by ROS Toolbox lets you:

- Connect to ROS from any operating system supported by MATLAB and Simulink.
- Leverage built-in functionality in MathWorks toolboxes for example, control systems, computer vision, machine learning, signal processing, and state machine design.
- Automatically generate standalone C++ based ROS nodes from algorithms designed in MATLAB and Simulink.

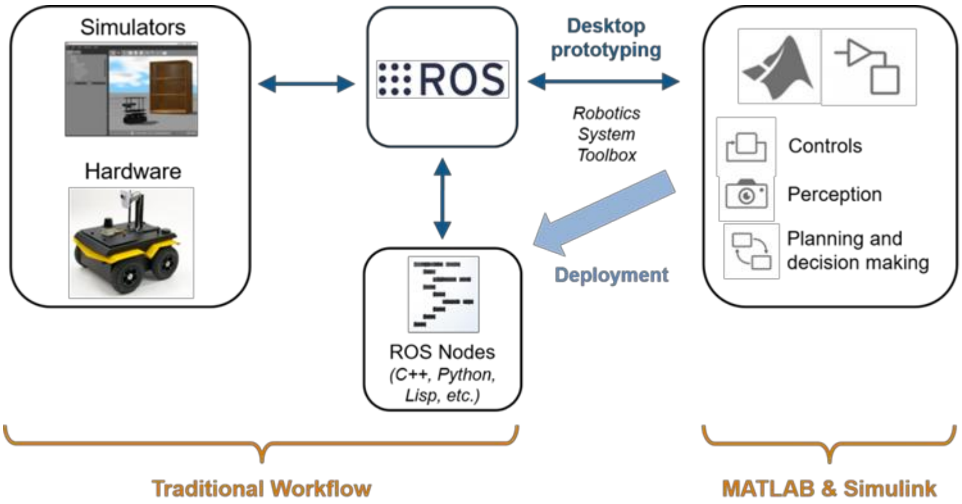


Figure 3.22 Block diagram of ROS

3.7.1 Fuzzy Control System Architectures

A fuzzy controller could be an uncommon knowledge-based framework, utilizing unpleasant thinking, in a rule chaining strategy. Able to recognize a few stages within the treatment of the rules. An ordinary graph of a fuzzy logic controller is appeared in figure below

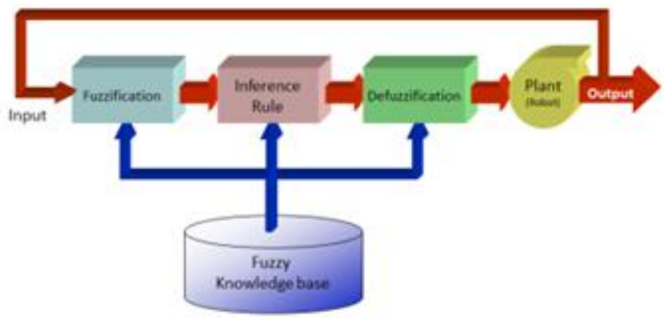


Figure 3.23 Fuzzy Control System Architectures

The primary step to recognize a fuzzy controller is fuzzification that change each value's inputs and outputs to membership for fuzzy control terms. The moment part is fuzzy inference which combines the truths obtained from the fuzzification of the rule base and conducts the fuzzy reasoning process and one of the taking after methods is generally utilized: Max-min deduction strategy (Mamdani).

When the input and the output factors and participation work are characterized a collection of rules is characterized, the rules are in "If Then" format and formally the If side is called the conditions and the at that point side is called the conclusion. The third part of fuzzy logic is defuzzification block. The objective is to convert the subsets of the outputs which are calculated by the inference engine.

3.7.2 Development of Fuzzy Logic Controller (FLC)

The design of Mamdani FLC and to navigate the robot from initial position to goal position without colliding with any obstacles. 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 the robot. The developed fuzzy logic controller (FLC) for navigation task used two inputs: the Error of position (x1) and the Angular error (x2). The outputs of the controller are the speed of the right and the left wheels.

For this project, we use a input triangular membership functions and output for gaussian membership function as for the effectiveness of our fuzzy logic controller of rapidity and accuracy.

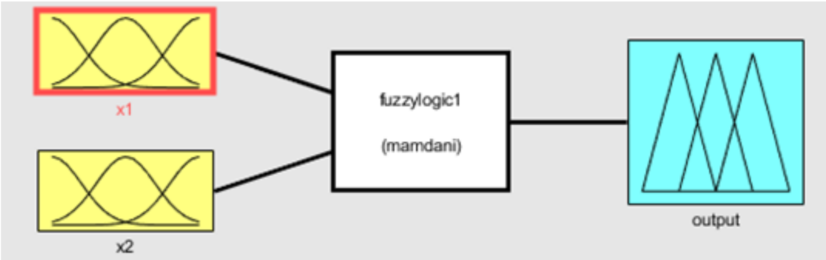


Figure 3.24 Fuzzy Logic Controller with 2 inputs and 1 output

3.7.3 Fuzzification

Fuzzification is the process of transforming crisp values into degree of membership for linguistic terms of fuzzy sets.

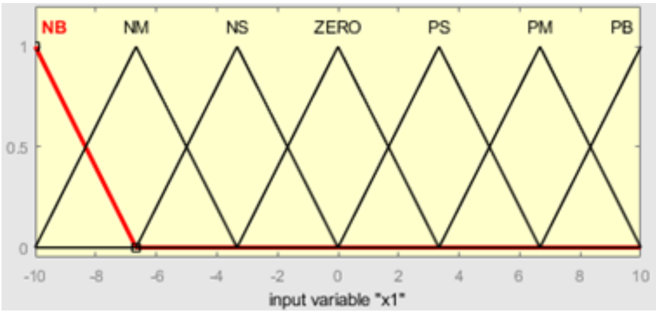


Figure 3.25 Input Triangular Membership fuzzification for x1 avoid obstacles

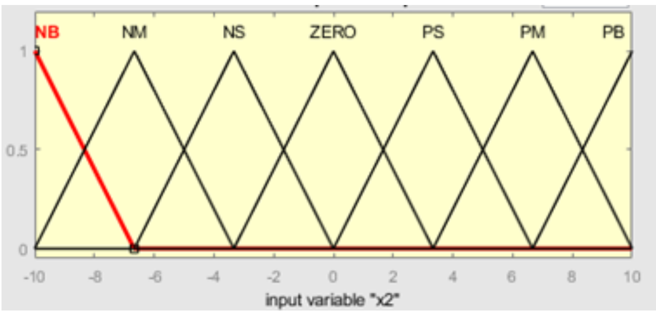
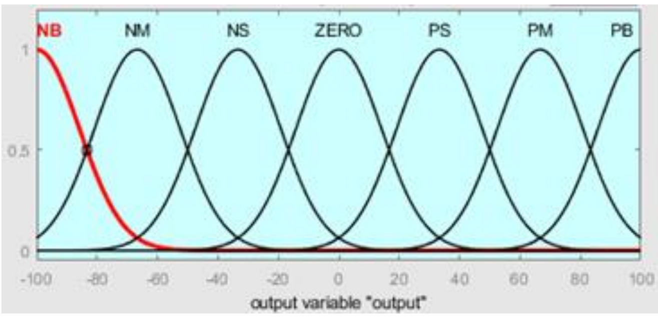


Figure 3.26 Input Triangular Membership fuzzification for x2 avoid obstacles

Figure 3.27 Output Gaussian Membership fuzzification for output avoid obstacles

3.7.4 Rule-Base Inference



After the stage of constructing membership functions, we construct the table for the rule bases, which is shown in Table 1. The expert system is developed based on the IF-THEN rules as 49 following rules to control robot tracking defined trajectory.

The following set of simple rules governs the obstacle avoidance controller:

IF X1 (Position Error) is Negative Big AND X2 (Angular Error) is Negative Big, THEN output (left and right wheel velocity of the robot) is Negative Big.

IF X1 (Position Error) is Negative Medium AND X2 (Angular Error) is Negative Big, THEN output (left and right wheel velocity of the robot) is Negative Big.

IF X1 (Position Error) is Negative Small AND X2 (Angular Error) is Negative Medium, THEN output (left and right wheel velocity of the robot) is Negative Medium.

IF X1 (Position Error) is Zero AND X2 (Angular Error) is Zero, THEN output (left and right wheel velocity of the robot) is Zero.

IF X1 (Position Error) is Positive Small AND X2 (Angular Error) is Negative Small, THEN output (left and right wheel velocity of the robot) is Zero.

		X1						
		NB	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
X2	**NB**	NB	NB	NB	NM	NM	NM	ZE
	NM	NB	NB	NM	NM	NM	ZE	PM
	NS	NB	NM	NM	NS	ZE	PM	PM
	ZE	NM	NM	NS	ZE	PS	PM	PM
	PS	NM	NM	ZE	PS	PM	PM	PB
	PM	NM	ZE	NS	PM	PM	PB	PB
	PB	ZE	PM	PM	PM	PB	PB	PB

Table 6 Rule-Base Inference

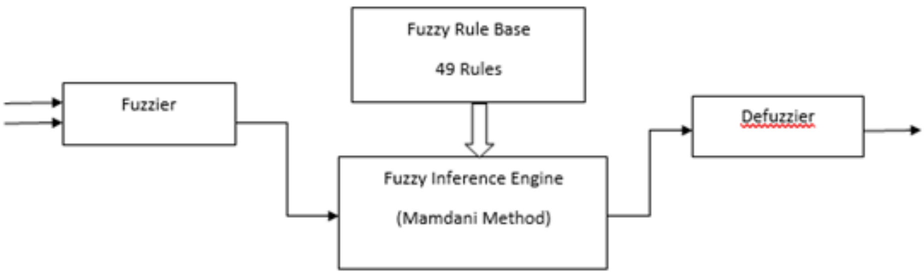


Figure 3.28 Method of the conventional tracking fuzzy control system

3.8 Key Milestones and Gantt chart

No.	Details \ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Topic Selection and Confirmation														
2	Preliminary Research Work														
3	Literature Review														
4	Extended Proposal														
5	Proposal Defence and Progress Evaluation														
6	Interim Draft Report														
7	Interim Report														

Table 7 Gantt Chart Final Year Project 1

No.	Details \ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Continuation of Final Year Project 1														
2	Simulation of Fuzzy Logic using ROS and Matlab														
3	Progress Report														
4	PSM2 Project Presentation														
5	Draft Technical Report														
6	Final technical Report Submission														

Table 8 Gantt Chart Final Year Project 2

Process

Milestone

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the results obtained and the problem encountered throughout the project development process is discussed. The robot simulation is simulated using Simulink MATLAB, ROS and Gazebo to robot to avoid the obstacles. The functionality of the robot hardware is tested in an indoor environment to ensure the robot is fully operating indoor.

4.2 Results

In this part, the project involves in both simulation part and hardware part. The hardware part is consisted of the TurtleBot3 robot base and using operating system Ubuntu 16.04 (Xenial Xerus) and ROS Kinetic. As for the simulation part, the robot will avoid the obstacles using operating system Ubuntu 20.04 (Focal Fossa) by using software MATLAB Simulink, ROS Noetic, Python, and Gazebo.

4.2.1 Hardware Development

Firstly, the material that mostly used in this project to build the robot chassis is waffle plate, plate support M3x35mm, plate support M3x45mm, PCB support, wheel, tire, ball caster and camera bracket. The two dynamixel motors are placed in the front of the robot and a two wheel is placed at the beside it. The reason that the motors are set before the robot is that the drag drive of two motor in front is superior than putting two wheel and two engines pushing at the back. Less grinding and weight for the wheel to contact the ground by comparing placing two wheel before the robot. The reason which there are as it were two motor is utilized in this project is that the two motor has the sufficient torque to bolster the robot add up to weight and spares battery power by comparing on utilizing four motor to run the robot.

Besides, the Raspberry Pi camera is introduced at the second most elevated spot of the robot chassis. This is often to ensure that the Pi camera get higher see in tracking the symbol without any obstacles from the environment and the equipment. Other than that, a 5m tall extend accuracy ultrasonic sensor is put fair underneath the Pi camera which permits the ultrasonic sensor to keep track the remove of the obstacles and the robot which fixed at slightest 1m in separate. Then two 30cm separate ultrasonic sensor is set in front of the robot which sense the front corner of the robot whether there are any obstacles in front and dodge them.

Thirdly, 11.1V LI-PO battery is set on the robot body to control up the motor driver and runs the motor. Futhermore, a save 11.1V LI-PO battery is utilized to power a up a high range ultrasonic sensor which act as the lighting for the camera to run at a low lighting surroundings. It too utilized as the supply for powering up the Raspberry Pi. 11.1V LI-PO battery is utilized to control up Raspberry Pi since the processor of the Raspberry pi needs more power to process information that collected from the picture processing. The perfect case for the control supply for powering up Raspberry Pi is 5V, 2.5A which the Raspberry Pi draws numerous current when there's a high utilization of the processing speed within the processing unit. Figure 4.2 appears the in general setup of the TurtleBot3.

Figure 4.1 Back view of TurtleBot 3

Figure 4.2 Front view of TurtleBot 3

4.3 Simulation Development

As for the programming development, it consists of MATLAB Simulink, ROS Simulation 1, ROS Simulation 2, ROS Simulation 3, ROS Simulation 4 and comparison fuzzy and without fuzzy.

4.4 MATLAB Simulink (Simulation 1)

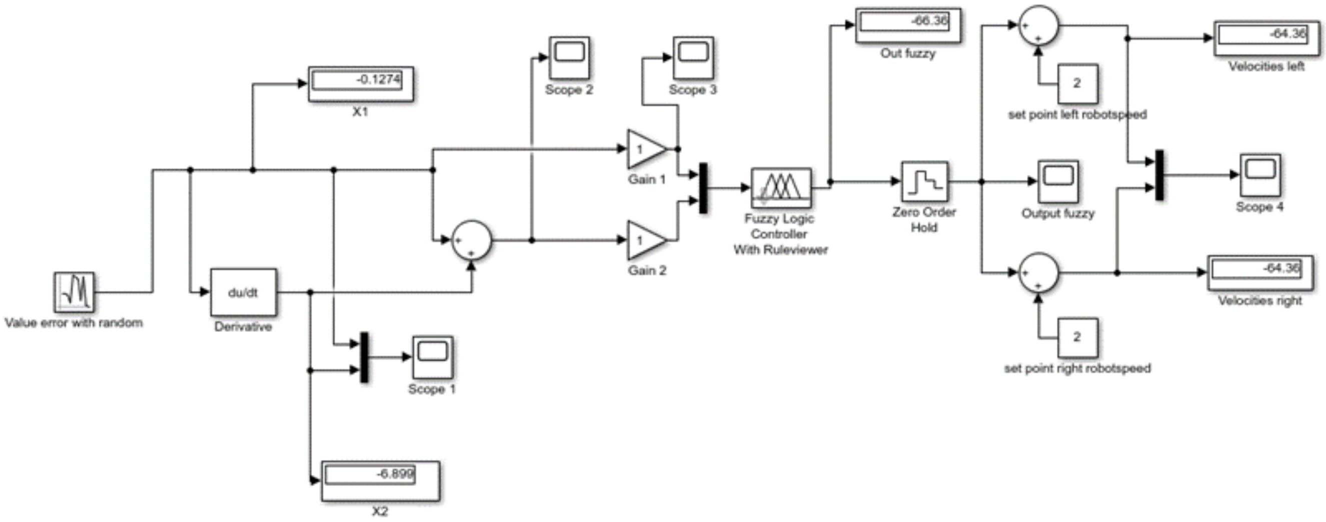


Figure 4.3 Simulink fuzzy MATLAB

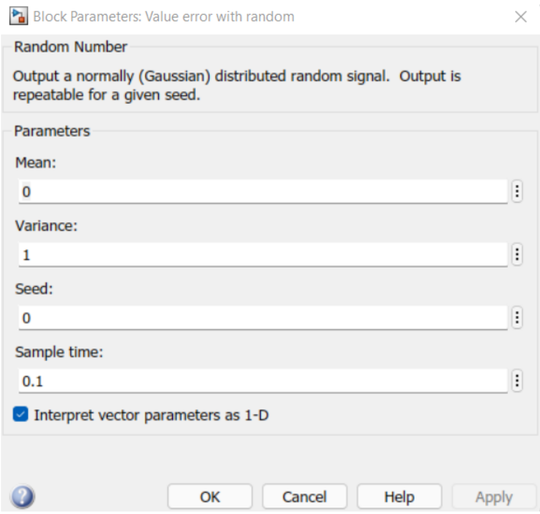


Figure 4.4 Value of mean of block value error with random

Figure 4.5 Rule Viewer

4.4.1 Discussion

For Simulink MATLAB, it starts with value with random which can enter any value to change the result of value of the error of position (x1), angular error (x2) and the value of the out fuzzy. The derivative is used to approximates the derivative of the input signal u with the respect to the simulation time t, du/dt. The gain block is used to multiplies the input by constant and it also generates its output by multiplying its input by a specified gain factor. The fuzzy logic controller with rule viewer implements a fuzzy inference system (FIS) in Simulink and displays the fuzzy inference process in the rule viewer during the simulation. To view the inference process, it must specify the input and output variables of your FIS, their corresponding membership functions, and the fuzzy rules its system. Zero order hold block is to hold its input of fuzzy logic controller for the sample periods. The set point left robot speed and set point right robot speed is to set the speed of the right and left wheel of the robot. After set up the speed, the result of the velocities of left wheel and the velocities of the right wheel will display.

The rule viewer is to shown to view the inference process for your fuzzy system. You can adjust the input values and view the corresponding output of each fuzzy rule, the aggregated output fuzzy set, and the defuzzified output value. To view the inference process, you must specify the input and output variables of your FIS, their corresponding membership functions, and the fuzzy rules for your system. To change the value of the x1 and the x2, it must change the value of the mean of the value error with random.

In the MATLAB Simulink result, the value of error of position (x1) is -0.1274 and the value of angular error (x2) is -6.899. This is because when we change the value of 0 in the mean of blocks value error with random, the value of x1 and x2 will change accordingly with the value we set at the mean value. The output of fuzzy value is -66.36 is display is same with the value of the output at rule viewer. The value of the left wheel velocities and right wheel velocities is -64.36 m/s, which is same value for both wheels. The result value of x1, x2, the output fuzzy, the velocities left and right wheel of the MATLAB Simulink and the rule viewer must be same in order to the simulation to be successful.

4.5 Fuzzy ROS (Simulation 1)

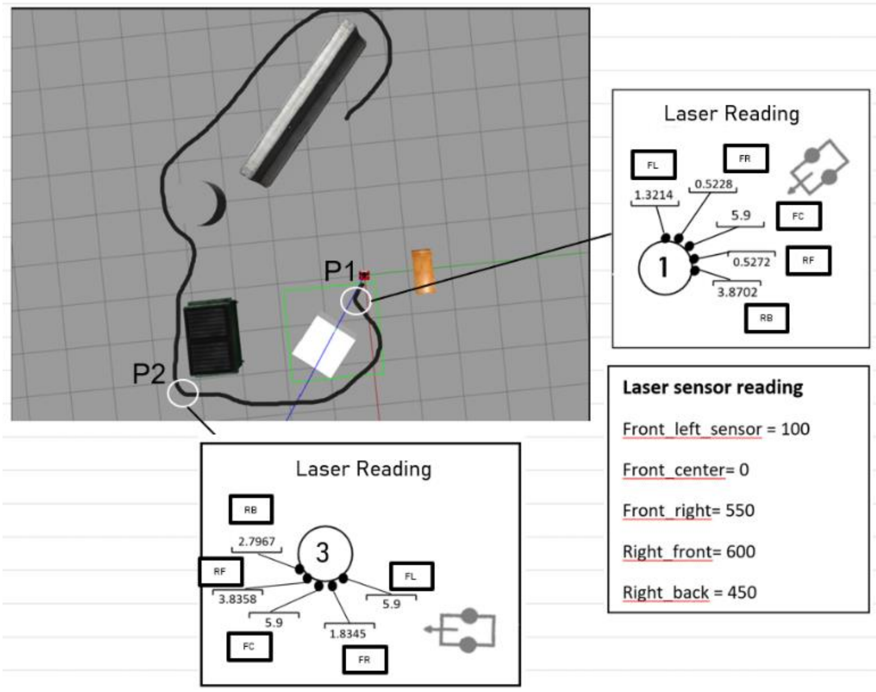


Figure 4.6 ROS Simulation 1

Figure 4.4 shows the simulation shows a robot avoids the obstacles. Four obstacles were places to see how the sensor and FLC avoid the obstacles. At the point P1, the robot sensed the obstacles 1 and avoid it. The robot will go to the left because the value reading sensor of the front left sensor (FL) and front right sensor (FR) is smaller the sensor of the right front sensor. The front center sensor (FC) reading is default which is 5.9m as it will go straight. The right back sensor (RB) will towards to the right because the higher reading. At the point P2, the robot will avoid obstacles 3. The robot will go straight because value of front left sensor (FL) and go to the left because of the front right sensor reading (FR) is smaller as avoid the obstacles at right and turn back to right due to front center sensor (FC). The laser sensor reading is indicating the distance the sensor from the obstacles. We set the value of the laser sensor reading in the coding. Front center is 0 to make a robot to go straight, front left sensor set to 100 because it moderate or near to the obstacles. Front right and right front is gap with 50 value to detect the obstacles with long range. In addition, the right back with a value of 450 is to detect the obstacles in moderate range of obstacles.

4.6 Fuzzy ROS (Simulation 2)

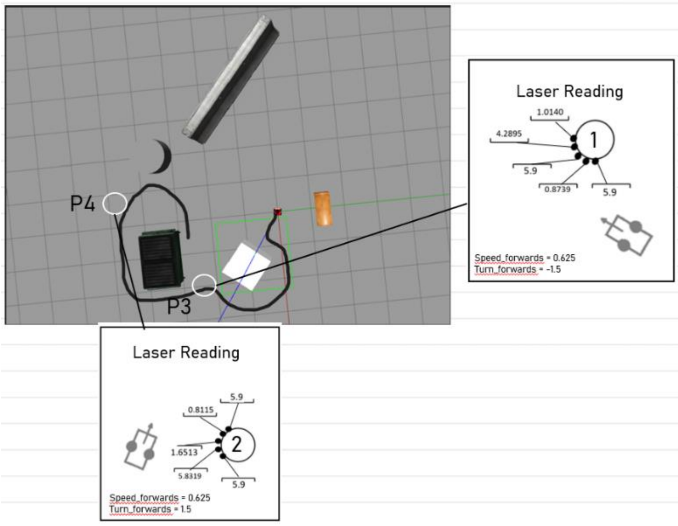


Figure 4.7 ROS Simulation 2

In this simulation, the robot will avoid three obstacles. The smaller value of the laser sensor of the robot, the more the robot intend to avoid the obstacles. The higher the value of the laser sensor, the less the robot avoid the obstacles. For the P3, the speed of the robot 0.625 and speed for the turn forward which is avoid obstacles is -1.5. For the P4, the speed of the robot 0.625 and speed for the turn forward which is avoid obstacles is 1.5.

4.7 Fuzzy ROS (Simulation 3)

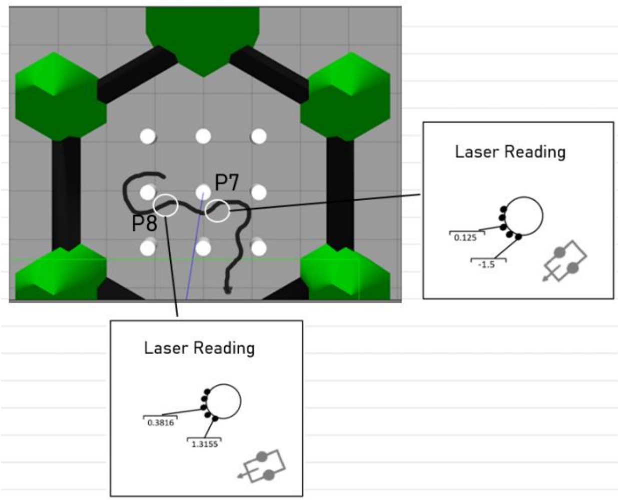


Figure 4.8 ROS Simulation 3

In this simulation, we using a Turtlebot as our robot to simulate fuzzy logic of avoidance robot obstacles. At P7, the speed of the robot is 0.125 and the speed of turn forward is -1.5. The negative value is because the robot goes to the left path and avoid the obstacles. At P8, the robot did not go to left as P7 and did not avoid the obstacles.

4.8 Implementation on Real Robot

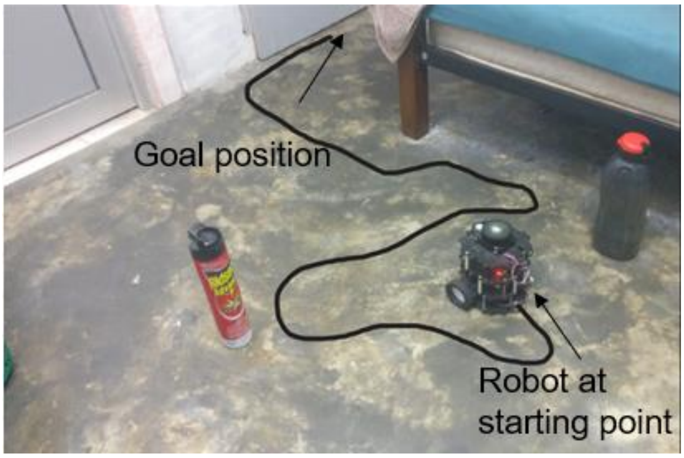


Figure 4.9 Real world environment 1 for the real robot

Figure 4.10 Real world environment 2 for real robot

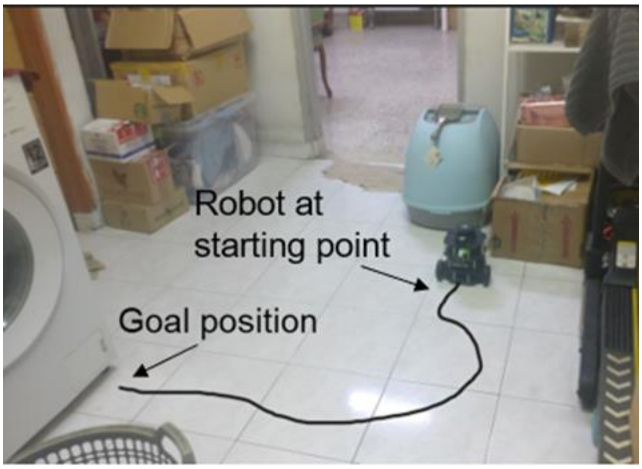


Figure 4.11 Real world environment 3 for the real robot

The proposed model for the robot navigation is implemented on real Turtlebot3 robot equipped with 360 Laser Distance Sensor LDS-01. The robot is navigated from starting position to the goal position in the environment presented by Fig 4.7, Fig 4.8, Fig 4.9. The linear velocity of the robot is taken as 0.4 meter per second. It is evident from Fig 4.7, Fig 4.8, and Fig 4.9 that the robot has to avoid obstacles on both of its sides during its drive from start to goal.

In first real world environment, the robot avoids the two obstacles. When the robot goes forward, the linear velocity of the robot is 0.4 and when it turns to the right the velocities the value becomes 1.5 to avoid the obstacles on the left side. It shows that the robot is successfully avoid the obstacles and reach the goal point.

In second real world environment, the robot uses a different path from the previous environment. In this environment, the robot avoiding three obstacles with the velocities of 0.125 and speed of the turn forward -1.5 when avoid the obstacles on the right. It takes 30s to reach the goal point.

In third real world environment, the robot is avoiding one obstacle which is chair. Its shows that, the velocities of the wheel of the robot are 0.4 and the turn forward is -1.5. It takes 10s to reach the goal point because its shortest path travel.

4.9 Comparison Fuzzy and Without Fuzzy

4.9.1 Simulation 1

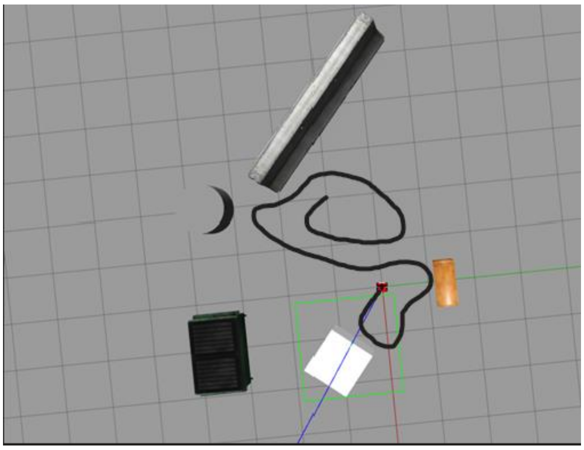


Figure 4.12 Simulation without fuzzy

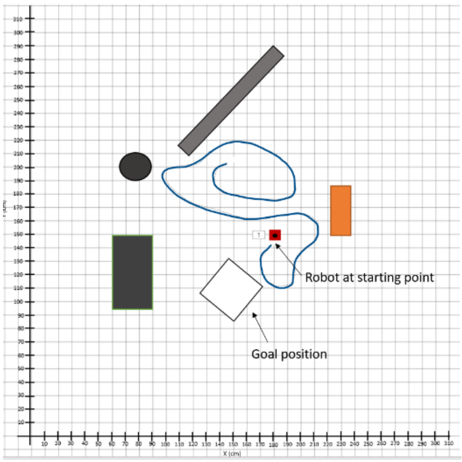


Figure 4.13 Graph simulation without fuzzy

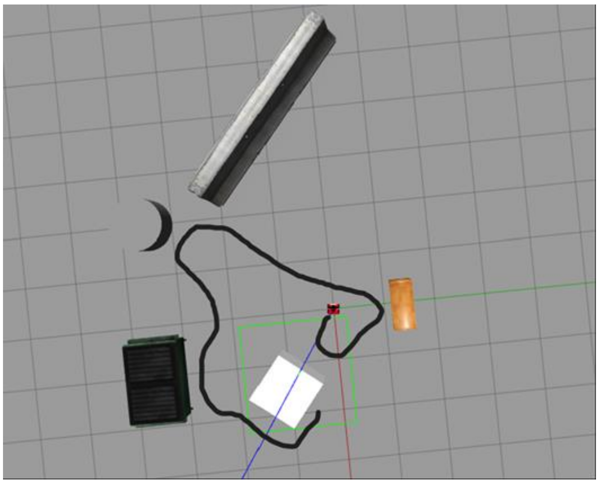


Figure 4.14 Simulation using fuzzy

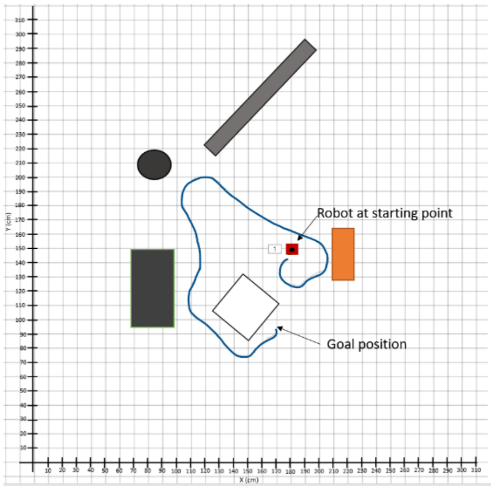


Figure 4.15 Graph simulation using fuzzy

	Without Fuzzy	Using Fuzzy
Obstacles	4	5
Time	59s	50s
Path Length	2.82m	2.12m

Table 9 Comparison Without Fuzzy and Fuzzy

Based on the Table 9 above, it shows the simulation of the mobile robot without fuzzy the time taken takes 59 second or more with the path length 2.82m or more with avoid four obstacles. It is because the path travel by the mobile robot is take a longer time to reach the goal point. Meanwhile, using fuzzy the time taken takes 50s to mobile robot to travel as it must avoid 5 obstacles with 3.12m of path length. The route of mobile robot using fuzzy is goes to the goal point, while without fuzzy it go nowhere and take a longer time to reach the goal point.

4.9.2 Simulation 2

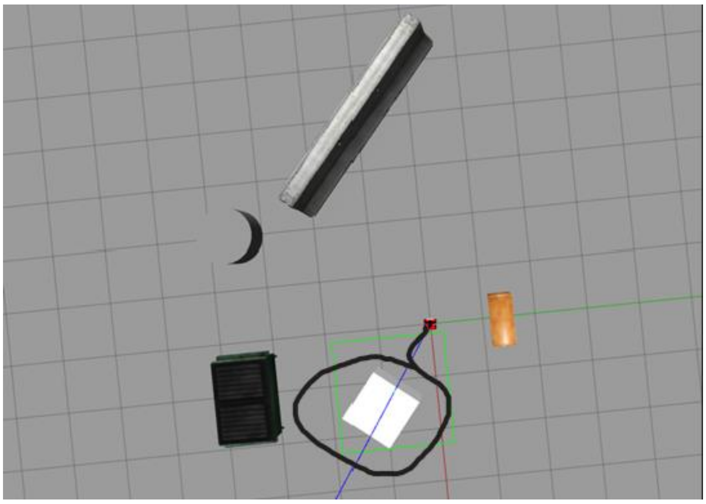


Figure 4.16 Simulation without fuzzy

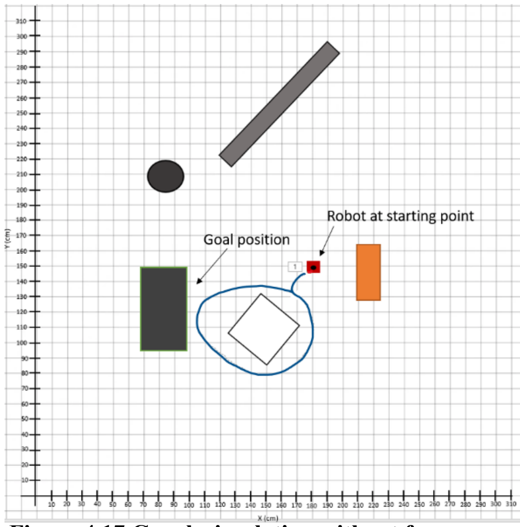


Figure 4.17 Graph simulation without fuzzy

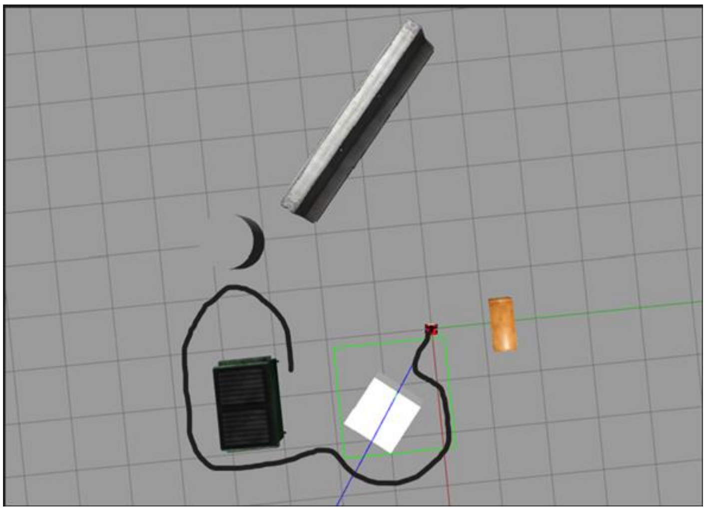


Figure 4.18 Simulation using fuzzy

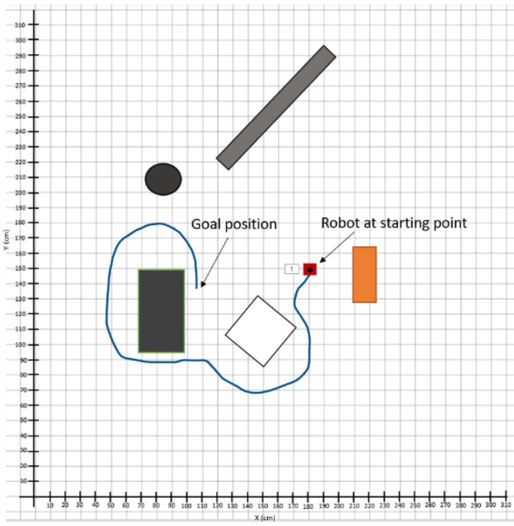


Figure 4.19 Graph simulation using fuzzy

	Without Fuzzy	Using Fuzzy
Obstacles	2	3
Time	50s	41s
Path Length	1.45m	1.58m

Table 10 Comparison Without Fuzzy and With Fuzzy

From the Table 10 above, when the mobile robot navigate the environment without fuzzy it will take 50s or sometimes more to reach the goal point. The robot avoid two obstacles with approximate 1.45m or more. It will circling the the avoidance obstcales and unable to reach the goal point. While using a fuzzy, mobile robot can reach the goal point successfully with time taken of 41 second with path length of 1.58 metre and avoid three obstacles within the path.

4.9.3 Simulation 3

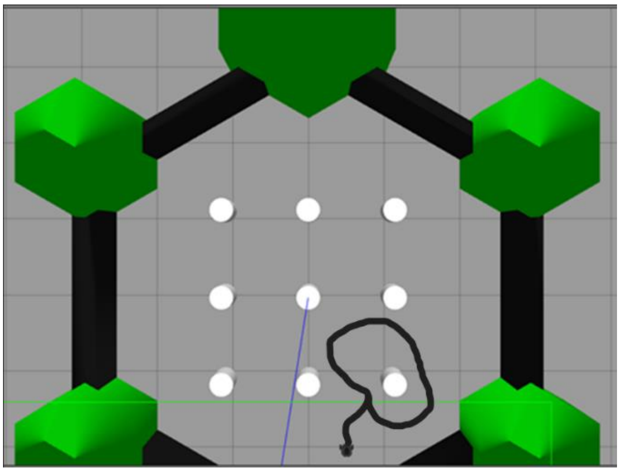


Figure 4.20 Simulation without fuzzy

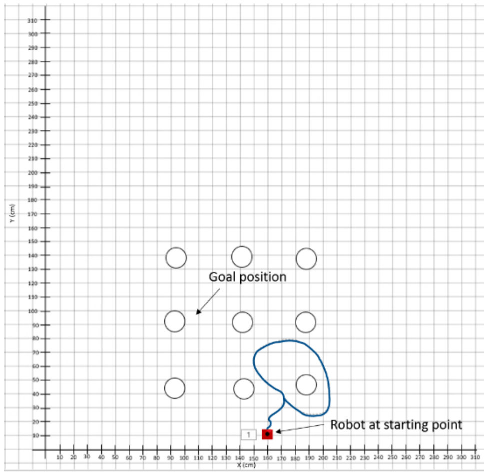


Figure 4.21 Graph simulation without fuzzy

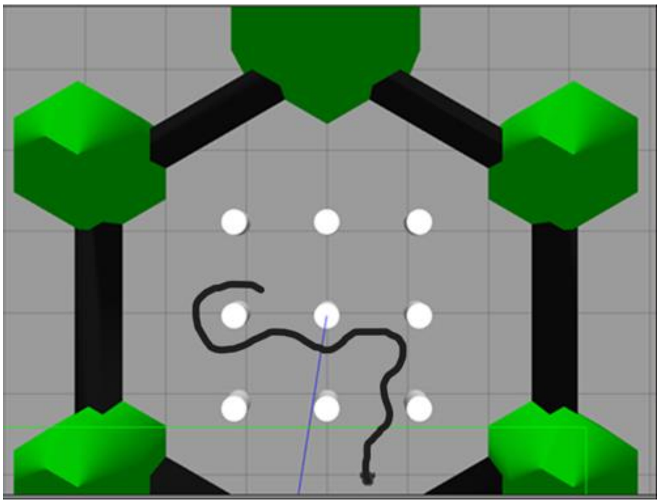


Figure 4.22 Simulation with fuzzy

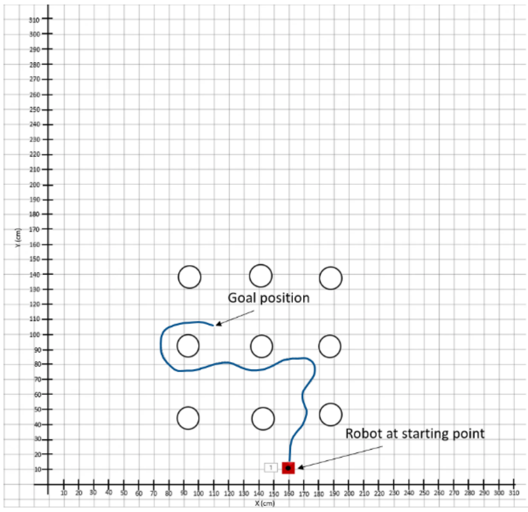


Figure 4.23 Graph simulation using fuzz

	Without Fuzzy	Using Fuzzy
Obstacles	4	6
Time	54s	50s
Path Length	1.80m	1.67m

Table 11 Comparison Without Fuzzy and With Fuzzy

Table 11 shows the comparison of obstacles avoidance of mobile robot without fuzzy and using a fuzzy. With fuzzy, the mobile robot follow the path to the goal point and avoid six obstacles equipped with robot laser sensor to avoid it. It takes 50s to reach the goal point as it has a shortest route of 1.80m. For without fuzzy, the mobile robot cannot configure the path where to go and the time taken for its reach the goal point will takes 54s with 1.80m of its path length.

4.9.4 TurtleBot3 Movement 1

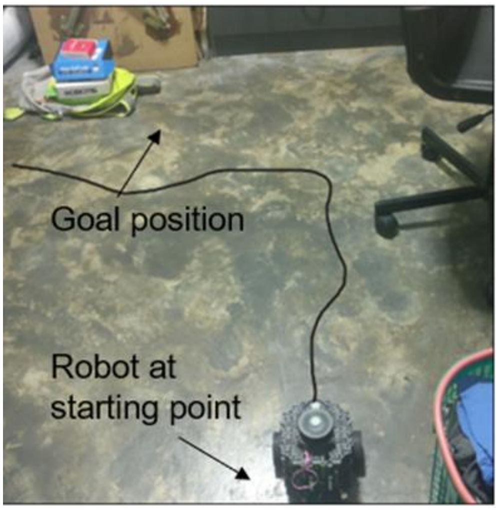


Figure 4.24 Simulation without fuzzy

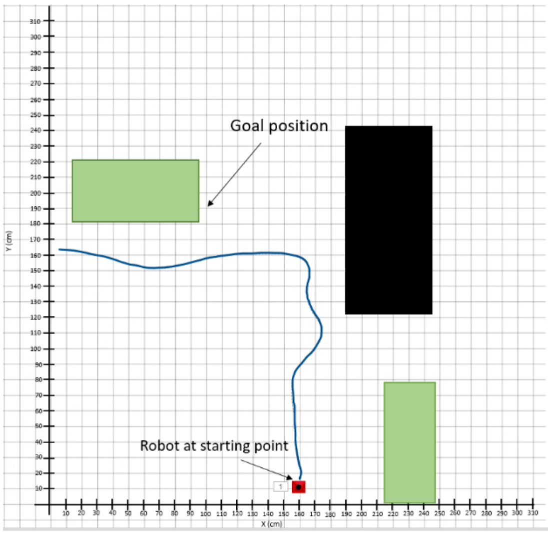


Figure 4.25 Graph simulation without fuzzy

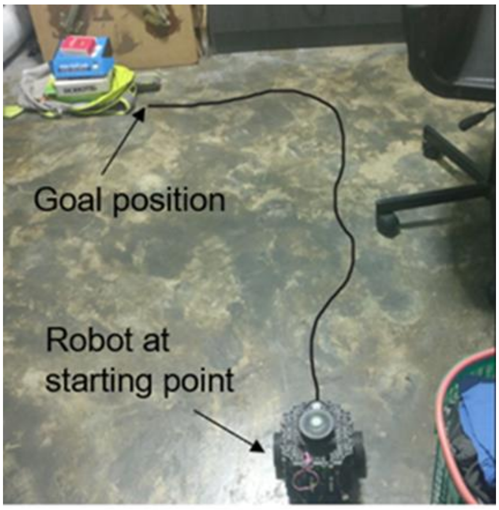


Figure 4.26 Simulation using fuzzy

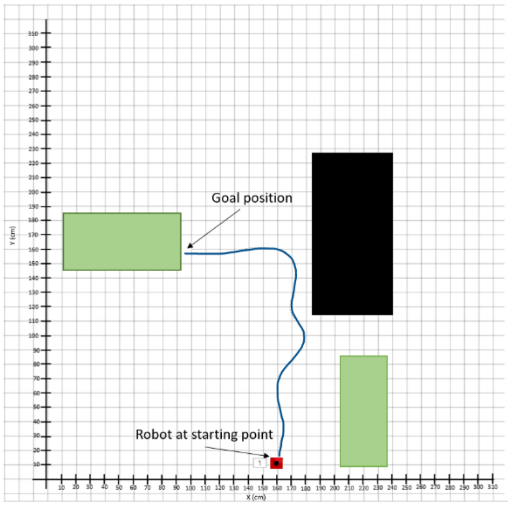


Figure 4.27 Graph simulation using fuzzy

	Without Fuzzy	Using Fuzzy
Obstacles	1	3
Time	0.17s	0.10s
Path Length	1.65m	1.58m

Table 12 Comparison Without fuzzy and using fuzzy

Table 12 showing a result of TurtleBot3 move to avoid the obstacles without fuzzy and using fuzzy. One obstacles is avoid the mobile robot when apply without fuzzy and it takes 0.17 second to reach near the goal point with the path length of 1.65m. It takes a longer time as the route of mobile robot is futher away from the starting point. With fuzzy, the mobile robot successfully reach the goal point as with shortest time of 0.10s and avoid three obstacles ahead it. The path length is shortest than the without fuzzy as it follow the path to the goal point. Three obstacles is avoiding by a mobile robot using fuzzy as it can detect the obstacles near it. The more the mobile robot avoid the obstacles the more efficiency of the mobile robot is.

4.9.5 TurtleBot Move 2

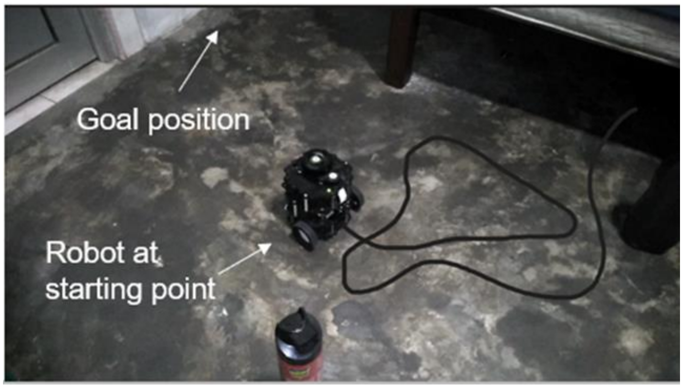


Figure 4.28 Simulation without fuzzy

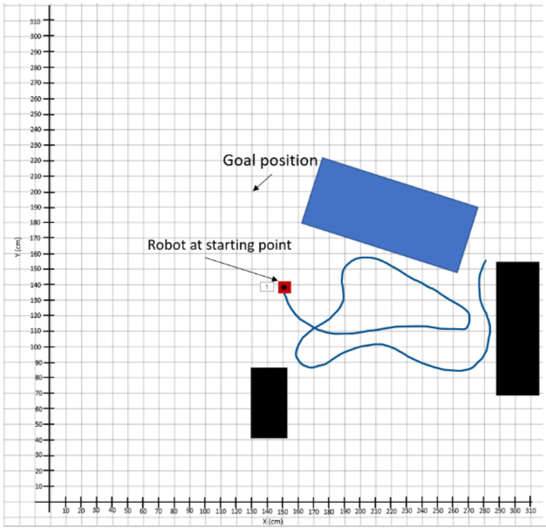


Figure 4.29 Graph simulation using fuzzy

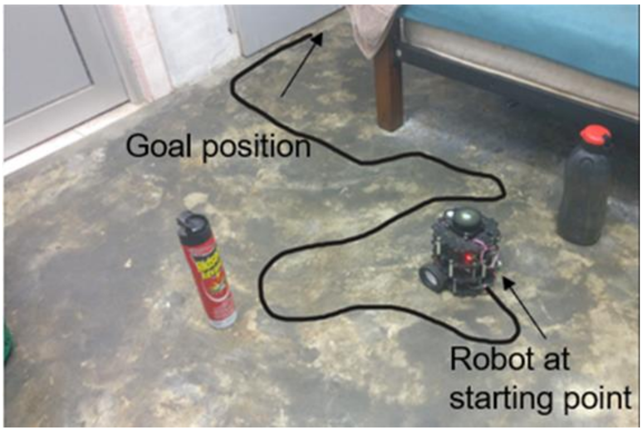


Figure 4.30 Simulation using fuzzy

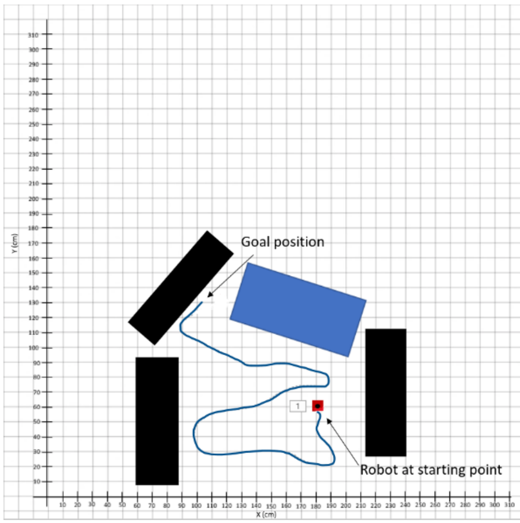


Figure 4.31 Graph simulation using fuzzy

	Without Fuzzy	Using Fuzzy
Obstacles	3	4
Time	0.40s	0.30s
Path Length	1.45m	1.38m

Table 13 Comparison Without fuzzy and using fuzzy

Figure 4.20 is showing a movement of TurtleBot3 with navigation without fuzzy. The mobile robot take 0.40s or sometimes more to reach the goal point. It avoid three obstacles that were place at the surroundings. Path length approximately 1.45m from the starting point of the robot that will it large than using fuzzy. It is because it use a different path than that were instructed. Mobile robot with apply of fuzzy is be able to avoid the four obstacles with 0.30s of time. It is a shortest time then the without fuzzy because it follow the path that embedded to it to reach the goal point.

4.9.6 TurtleBot Movement 3

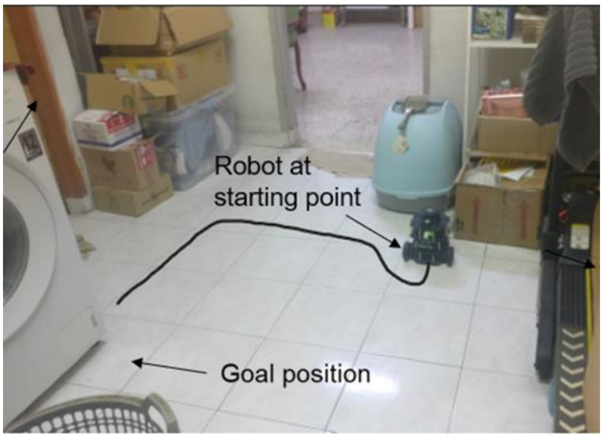


Figure 4.32 Simulation without fuzzy

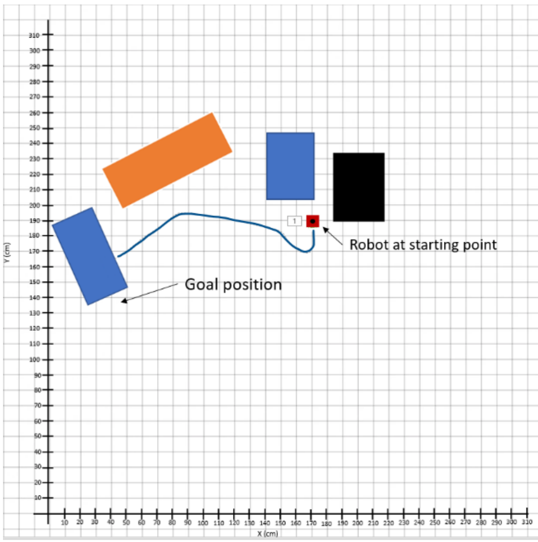


Figure 4.33 Graph simulation using fuzzy

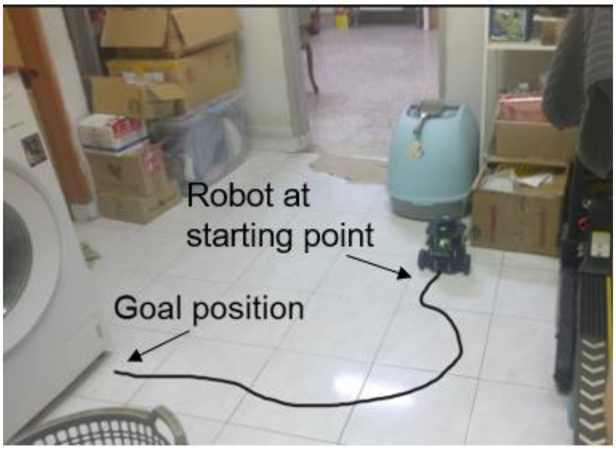


Figure 4.34 Simulation using fuzzy

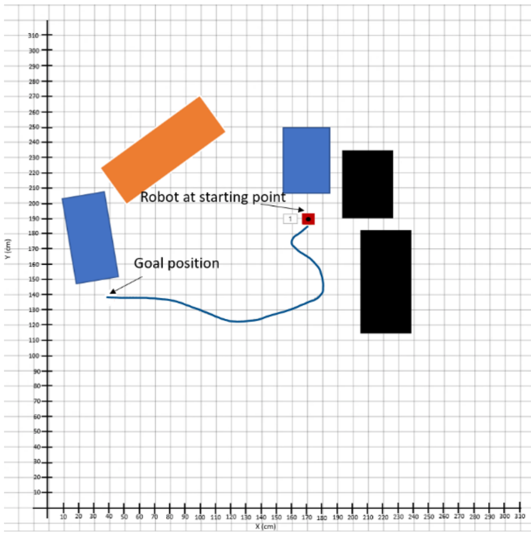


Figure 4.35 Graph simulation using fuzzy

	Without Fuzzy	**Using Fuzzy**
Obstacles	4	5
Time	0.25s	0.22s
Path Length	1.6m	1.4m

Table 14 Comparison Without fuzzy and using fuzzy

Table 14 shows the the comparison of movement of Turtlebot3 avoiding the obstacles without fuzzy and using fuzzy. When use a fuzzy, the mobile robot can go to the goal point safely and successfully after avoiding five obstacles through it with time taken of 0.22s. It was the shortest time to mobile robot to reach the goal point compare to the without fuzzy which is 0.25s of time. Mobile robot without fuzzy got a path length of 1.6m route which is more futher than mobile robot with fuzzy. The further the path length, the longer the time taken to reach the goal point.

Simulation	Wihout fuzzy logic controller			Using fuzzy logic controller		
	Obstacles	Time Taken	Path Length	Obstacles	Time Taken	Path Length
1	4	50s	2.82m	5	58s	3.12m
2	2	30s	1.45m	3	41s	1.58m
3	4	54s	1.80m	6	50s	1.67m

Table 15 Comparison of robot simulation without fuzzy and using fuzzy

TurtleBot Movement	Without fuzzy logic controller			Using fuzzy logic controller		
	Obstacles	Time Taken	Path Length	Obstacles	Time Taken	Path Length
1	1	0.17s	1.65m	3	0.10s	1.58m
2	3	0.20s	1.25m	4	0.30s	1.38m
3	4	0.25s	1.60m	5	0.22s	1.40m

Table 16 Comparison of Turtlebot movement without fuzzy and using fuzzy

Fuzzy	Without fuzzy
No affected by obstacles potential field	Robot strangely effected by obstacles potential fields
Can pass two or more obstacles	Unable to reach the goal point
Follow the path that instructed	Difference path travel by robot

Table 17 Comparison Fuzzy and Without Fuzzy

4.10 Discussion

From the results, the mobile robot has to navigate in environment with obstacles. We notice that laser scanner range sensors detect an obstacle, the robot is forced to make an adequate turn to avoid collision with the object. At the beginning of the simulation, the mobile robot starts sensing the environment for possible obstacle detection. The obstacles were placed to see how the robot sensor and FLC avoid the obstacles. At Fuzzy ROS (Simulation 1), it shows that when the value of the sensor is small, the more the robot intend to avoid that angle of obstacles. When the value of the laser sensor is bigger, the more the robot intend to go to the straight path. At Fuzzy ROS (Simulation 2), based on the simulation result, the value of the robot speed forwards is same, but the turn forwards are different. When the robot avoids the obstacles, the robot intends to go right or left to avoid the obstacles. When the robot goes to the left to avoid the obstacles, the value of turn forward will negative and when go to the right, the value of turn forward will become positive. In this simulation also, we use a different path by robot to test whether the robot successfully avoid the obstacles. It found that when more obstacles were placed in the environment, it takes more time the robot to reach the goal point. At Fuzzy ROS (Simulation 3), we use a different environment. It shows that the value of robot speed is smaller than the robot speed at Fuzzy ROS (Simulation 2). It is because the obstacles in this environment is closer than at environment Fuzzy ROS (Simulation 2). It also takes more time to reach the goal point as the robot take time to avoid the closer obstacles.

For the real robot implementation, we can see that the robot is successfully avoid the obstacles and reach the goal point. It shows that the Turtlebot take time to reach the goal point when more obstacles are added. The speed of the Turtlebot is the same, but the value of turn forward is different according the obstacles right or left. The real world environment for Fig 4.9 and Fig 4.10 is different to test the efficiency and performance of the robot avoid the obstacles.

Comparison with fuzzy and without fuzzy, if the integration of Fuzzy Logic Controller (FLC) is applied in this project, the Mamdani FLC is able to detect the obstacles and control the velocity of the left and right wheel in each environment. The FLC is able to integrate with the laser sensor to communicate and the path travelled for the mobile robot environment from the initial position to the goal position is succeed. We also can see that the when the robot with fuzzy implementation, the speed of the robot is different. With fuzzy the speed is smaller than the without fuzzy because the robot is not strangely affected by obstacles potential field and not use a different path to travel. Other than that, the FLC does not initialised in no obstacle environment. If the number of obstacles is increased, the time taken for the robot to reach the target increases for Mamdani FLC.

CHAPTER 5

CONCLUSION

In conclusion, this paper presents the plan and simulation of the mobile robot with laser sensor. To begin with, we begun with the improvement of the show of the robot framework and after that we outlined the route of the mobile robot. The simulation comes about utilizing ROS and Gazebo 11 simulation platform, have appeared the adequacy of the plan of mobile robot utilizing ROS giving great route performances. To include obstacles avoidance have we opted within the Gazebo 11 to attain the ultimate objective which to plan robot route framework utilizing ROS that can dodge obscure or unforeseen obstacles in an obscure environment. Simulation comes about utilizing Gazebo 11 stage have appeared a clear execution change of the navigation framework. ROS have appeared its adaption to the two-wheeled robot or portable robot. In reality, the TurtleBot3 is able of dodging obstacles based on an ROS.

This progress project is finished inside the 2 semesters of studies. The objective of the projects is to get it the standards and ideas of Mobile Robot Navigation on Motion Tracking. This objective is accomplished in Final Year Project 1 and Final Year Project 2 and clearly appeared in Chapter 2 of the report which is Literature Review. Another objective is to recognize conceivable approaches to oversee the robot to dodge obscure or unforeseen obstacles in an obscure environment. A few routine strategies are distinguished such as Python Programming Method, command in ROS and Fuzzy Logic method. However, the strategy that's recently utilized is Artificial Intelligence Methods. Fuzzy Logic is beneath this Artificial Intelligence Methods. At long last, the final objective of dissect the execution of mobile robot has been examined clearly in Chapter 4 of Results & Discussions. Those goals are fulfilled and accomplished in this final year project 1 and final year project 2 with firm commitment towards the methodology and planning arranged.

REFERENCES

[1] X. Li and B. J. Choi, “Design of obstacle avoidance system for mobile robot using fuzzy logic systems,” *Int. J. Smart Home*, 2013.

[2] A. shitsukane, W. Cheriuyot, C. Otieno, and M. Mvurya, “A Survey on Obstacles Avoidance Mobile Robot in Static Unknown Environment,” *Int. J. Comput.*, no. March, 2018.

[3] A. Najmurrokhman, Kusnandar, U. Komarudin, Sunubroto, A. Sadiyoko, and T. Y. Iskanto, “Mamdani based Fuzzy Logic Controller for A Wheeled Mobile Robot with Obstacle Avoidance Capability,” in *Proceedings of the 2019 International Conference on Mechatronics, Robotics and Systems Engineering, MoRSE 2019*, 2019.

[4] J. Jameson, S. N. H. Sheikh Abdullah, and K. M. Maluda, “Error Analysis in Applying Fuzzy Logic Based Obstacle Avoidance Algorithm for Robot Soccer,” in *Communications in Computer and Information Science*, 2013.

[5] R. H. Abiyev, I. S. Günsel, N. Akkaya, E. Aytac, A. Çağman, and S. Abizada, “Fuzzy control of omnidirectional robot,” in *Procedia Computer Science*, 2017.

[6] A. Prayitno, V. Indrawati, and G. Utomo, “Trajectory tracking of AR.Drone quadrotor using fuzzy logic controller,” *Telkomnika (Telecommunication Comput. Electron. Control.*, 2014.

[7] Q. Y. Bao, S. M. Li, W. Y. Shang, and M. J. An, “A fuzzy behavior-based architecture for mobile robot navigation in unknown environments,” in *2009 International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009*, 2009.

[8] L. Ren, W. Wang, and Z. Du, “A new fuzzy intelligent obstacle avoidance control strategy for wheeled mobile robot,” in *2012 IEEE International Conference on Mechatronics and Automation, ICMA 2012*, 2012.

[9] C. H. Lin, S. H. Wang, and C. J. Lin, “Interval type-2 neural fuzzy controller-based navigation of cooperative load-carrying mobile robots in unknown environments,” *Sensors (Switzerland)*, 2018.

[10] E. Masehian and M. R. Amin-Naseri, “A voronoi diagram-visibility graph-potencial field compound algorith for robot path planning,” *J. Robot. Syst.*, 2004.

[11] P. G. Zavlangas and S. G. Tzafestas, “Motion control for mobile robot obstacle avoidance and navigation: A fuzzy logic-based approach,” *Syst. Anal. Model. Simul.*, 2003.

[12] M. Quigley *et al.*, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, 2009.

[13] R. Brafman and S. Givati, “Topic in AI : Robot Programming & ROS.”

[14] B. Dieber *et al.*, *Robot Operating System-The Complete Reference (Volume 4) Several authors*, vol. 4, no. Volume 4. 2019.

[15] R. L. Guimarães, A. S. de Oliveira, J. A. Fabro, T. Becker, and V. A. Brenner, "ROS navigation: Concepts and tutorial," *Stud. Comput. Intell.*, 2016.

[16] A. Santos, A. Cunha, N. Macedo, and C. Lourenço, "A framework for quality assessment of ROS repositories," in *IEEE International Conference on Intelligent Robots and Systems*, 2016.

[17] R. Carvalho, A. Cunha, N. Macedo, and A. Santos, "Verification of system-wide safety properties of ROS applications," in *IEEE International Conference on Intelligent Robots and Systems*, 2020.

[18] K. Takaya, T. Asai, V. Kroumov, and F. Smarandache, "Simulation environment for mobile robots testing using ROS and Gazebo," in *2016 20th International Conference on System Theory, Control and Computing, ICSTCC 2016 - Joint Conference of SINTES 20, SACCS 16, SIMSIS 20 - Proceedings*, 2016.

[19] L. Ibarra and C. Webb, "Advantages of Fuzzy Control While Dealing with Complex/ Unknown Model Dynamics: A Quadcopter Example," in *New Applications of Artificial Intelligence*, 2016.

[20] "Fuzzy Logic Controllers : Optimization Issues on Design and Rule Base Reduction Algorithms," 2013.

[21] E. R. A. Safian, "• Introduction • Fuzzy Inference Systems • Examples," p. 131.

[22] S. S. Izquierdo and L. R. Izquierdo, "Mamdani fuzzy systems for modelling and simulation: A critical assessment," *JASSS*, 2018.

[23] N. Kumar and Z. Vámossy, "Robot navigation with obstacle avoidance in unknown environment," *Int. J. Eng. Technol.*, 2018.

[24] A. Pandey and D. R. Parhi, "MATLAB Simulation for Mobile Robot Navigation with Hurdles in Cluttered Environment Using Minimum Rule based Fuzzy Logic Controller," *Procedia Technol.*, 2014.

[25] H. Omrane, M. S. Masmoudi, and M. Masmoudi, "Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation," *Comput. Intell. Neurosci.*, 2016.

[26] N. I. A. Apandi and A. Martin, "The Integration of Fuzzy Logic System for Obstacle Avoidance Behavior of Mobile Robot," *Int. J. Electr. Eng. Appl. Sci.*, vol. 2, no. 1, pp. 31–38, 2019.

[27] M. Algabri *et al.*, "Wireless Vision-Based fuzzy controllers for moving object tracking using a quadcopter," *Int. J. Distrib. Sens. Networks*, 2017.

[28] B. Mrinal, "Adaptive Network based Fuzzy Inference System (ANFIS) as a Tool for System Identification with Special Emphasis on Training Data Minimization," p. 141, 2008.

[29] Y. Zhang, X. Wang, X. Wu, W. Zhang, M. Jiang, and M. Al-Khassaweneh, "Intelligent hotel ROS-based service robot," in *IEEE International Conference on Electro Information Technology*, 2019.

[30] R. Li, M. A. Oskoei, K. D. McDonald-Maier, and H. Hu, "ROS based multi-sensor navigation of intelligent wheelchair," in *Proceedings - 2013 4th International Conference on Emerging Security Technologies, EST 2013*, 2013.

[31] R. Barth *et al.*, "Using ROS for agricultural robotics : design considerations and experiences," in *Second International Conference on Robotics and associated High-technologies and Equipment for Agriculture and forestry, RHEA-2014. May 21-23, 2014 Madrid, Spain.*, 2014.

[32] M. Köseoğlu, O. M. Çelik, and Ö. Pektaş, "Design of an autonomous mobile robot based on ROS," in *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*, 2017.

[33] K. Alisher, K. Alexander, and B. Alexandr, "Control of the mobile robots with ROS in robotics courses," in *Procedia Engineering*, 2015.

[34] A. Shitsukane, W. Cheruiyot, C. Otieno, and M. Mvurya, "Fuzzy logic sensor fusion for obstacle avoidance mobile robot," in *2018 IST-Africa Week Conference, IST-Africa 2018*, 2018.