Design of an Adaptive Neuro-Fuzzy Path Tracking and Obstacle Avoidance System for a Mobile Robot

By: Dawit Nigussie

A Thesis Submitted to

The Department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of

Master's in Electrical Power and Control Engineering

(Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

March, 2022

Adama, Ethiopia

Design of an Adaptive Neuro-Fuzzy Path Tracking and Obstacle Avoidance System for a Mobile Robot

By: Dawit Nigussie

Advisor: Dr. Tafesse Asrat

A Thesis Submitted to
The Department of Electrical Power and Control Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of
Master's in Electrical Power and Control Engineering
(Control Engineering)

Office of Graduate Studies
Adama Science and Technology University

March, 2022
Adama, Ethiopia

DECLARATION

I declare here that this Master Thesis which is entitled with “Design of an Adaptive Neuro-Fuzzy Path Tracking and Obstacle Avoidance System for a Mobile Robot” is my original thesis. Which is not be given for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been acknowledged through citation.

Dawit Nigussie	____________	____________
Name of Student	Signature	Date

RECOMMENDATION

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "Design of an Adaptive Neuro-Fuzzy Path Tracking and Obstacle Avoidance System for a Mobile Robot" prepared under my/our guidance by Dawit Nigussie submitted in partial fulfillment of the requirements for the degree of Master's of Science in Electrical Power and Control Engineering (control engineering). Therefore, I/we recommend the submission of revised version of the thesis to the department following the applicable procedures.

Dr. Tafesse Asrat	________________	____________________
Major Advisor	Signature	Date
Prof. Samsun Ma	_____________	____________________
Co-advisor	Signature	Date

APPROVAL PAGE

We, the advisors of this thesis entitled "Design of an Adaptive Neuro-Fuzzy Path Tracking and Obstacle Avoidance for a Mobile Robot" and developed by **Dawit Nigussie,** hereby certify that the recommendations and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Tafesse Asrat		
Advisor	Signature	Date
Prof. Samsun Ma		
Co-advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by **Dawit Nigussie** have read and evaluated the thesis entitled "Design of an Adaptive Neuro-Fuzzy Path Tracking and Obstacle Avoidance System for a Mobile Robot" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Electrical Power and Control Engineering (Control).

Chairperson	Signature	Date
Internal Examiner	Signature	Date
Dr Beteley Teka		10-03-2022
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENTS

First of all, I want to express my gratitude to the Solicitous and Almighty God for giving me the strength, patience, and courage necessary to accomplish the thesis.

I would like to thank my advisor, **Dr. Tafesse Asrat**, for his unwavering support of my thesis, as well as his patience, motivation, and curiosity. Mainly, his advice was invaluable to me during my thesis research and writing.

I am particularly grateful to **Prof. Samsun Ma** for providing me with several suggestions and assistance during the process of producing the thesis document, as well as for the excellent ideas and discussions presented in the paper.

In addition, my heartfelt thanks go to **Prof. Gang Gyoo Jin** for his patience, helpful attitude, and sharing his work in implementing MATLAB coding.

Finally, I would like to express my gratitude to my entire family. Words can't explain how much I love, encourage, and appreciate my father and friends for all the sacrifices they've made for me.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ANFIS	Adaptive Neuro-Fuzzy Inference System
CCW	Counter Clockwise
CW	Clock Wise
CR	Center of Rotation
DC	Direct Current
DOF	Degree of Freedom
FIS	Fuzzy Inference System
FLC	Fuzzy Logic Controller
FNN	Fuzzy Neural Network
ICC	Instantaneous Center of Curvature
ICR	Instantaneous Center of Rotation
MATLAB	Matrix Laboratory
MF	Membership Function
MR	Mobile Robot
PID	Proportional Integral Derivative
WMR	Wheeled Mobile Robot

LIST OF SYMBOLS

A	The intersection of the axis of symmetry with the driving wheels axis
a	The distance between the center of mass and driving wheels axis in X-direction
C	The center of mass of the platform
D	Represent the distance between the two wheels
d_x	The distance traveled in X-direction
d_y	The distance traveled in Y-direction
e_x	The difference between the robot's position and the coordinates of the position along with the X-axis
e_y	The difference between the robot's position and the coordinates of the position along with the Y-axis
e_Θ	The input of the orientation ANFIS controller
$f_\Theta (t)$	The output of the orientation ANFIS controller
L	The distance between the two wheels
Q	Vector of independent velocity variables
R	Represent signed distance from ICC to the midpoint between the two wheels
r	The radius of each driving wheel
V	The translational velocity of the platform in the local frame
V_x	Represent the instantaneous horizontal velocity
V_y	Represent the instantaneous vertical velocity
$W_1,\ldots, W_m$	The output of the fuzzy controller
W_i^Θ	Adaptive parameter of the orientation ANFIS controller
$X_d (t)$	Represent the X-coordinates of the robot according to the reference frame
$Y_d (t)$	Represent the Y-coordinates of the robot according to the reference frame
Θ	The angle from X-axis to the robot's motion direction
$\Theta_d (t)$	Trajectories curvature at each step time t
Ω_R	The rotational velocity of the right wheel

Ω_L	The rotational velocity of the left wheel
$\Delta\Omega$	The absolute value of the angular velocity of difference between two wheels
Ω	The rotational velocity of the platform in the local and global frames
φ	Azimuth

ABSTRACT

Over the last two decades, mobile robot navigation has remained a challenge. Mobile robots must operate in unfamiliar and complex situations, and their employment in material handling has expanded dramatically in recent years. Trajectory tracking and control is often a difficult problem in mobile robot systems. An Adaptive Neuro-Fuzzy Inference System Controller is created for tracking the trajectory of a mobile robot in this proposed work. To begin, the trajectory is pre-planned with a few precise places in the work space to match the shape requirements. Adaptive Neuro Fuzzy Inference System uses these via-points as a training set to synthesize the smooth curve. Furthermore, due to the enormous number of via-points, the technique is straightforward to calculate. Finally, two examples are simulated to see how the ANFIS structure affects the produced trajectory. The simulation results show that the suggested method can generate a smooth trajectory that is a good match with the desired trajectory, even if the desired trajectory has a convoluted shape, by utilizing the appropriate structure of ANFIS.

Impediment avoidance tactics are another difficult topic in mobile robot systems. This device is meant to help wheeled mobile robots avoid front collisions by employing a fuzzy logic controller (FLC). A robust controller is installed in a mobile robot as it studies the input and output to help it navigate without colliding with any impediments. In this research, I offer a fuzzy logic controller-based obstacle avoidance control algorithm for mobile robots. With an efficient fuzzy logic look-up table, traditional approaches can be replaced in terms of efficiency. Even the output of a wheel's fuzzy logic controller can be utilized to adjust the direction and speed of obstacle avoidance. In the simulation, place a number of obstacles in the way. Allow the robot to reach the target on its own. Experiments demonstrate that the robot has an effective obstacle avoidance path.

Keywords: - Adaptive Neuro-Fuzzy Inference, Fuzzy logic controller, Wheeled Mobile robot, Obstacle avoidance, Trajectory tracking

CHAPTER 1

INTRODUCTION

1.1. Background

Robots have sparked a lot of curiosity in the past few years, and they've performed a lot of important roles in the globe. Robots, in particular, can free humans from hazardous or risky professions like nuclear waste cleanup, mining, forestry, agriculture, military service, firefighting, marine exploration, and robotic conflicts, among others [1]. In areas such as office automation and surgery, robots are beginning to have an impact on our daily lives. Robots may also assist handicapped persons in their daily routines.

A wheeled mobile robot (WMR) is a wheeled vehicle that can operate autonomously without the assistance of an external human driver thanks to an internal computer-driven motor. Because of its widespread use in mobile robots, quick moves, simple to use and easy controllers [2][3]. A differential drive, which is used by many mobile robots with wheels, is a type of drive in which each wheel is powered by its own actuator. As a result, changing the linear or angular speed of the left and right wheels can control the direction of the wheeled mobile robot. The fuzzy controller is utilized to control various paths in these approaches, and the system maintains good tracking performance.

Figure 1.1 Wheeled Mobile Robot (WMR) [4]

Mobile robots require a strong controller to allow the quick integration of input and output while navigating in unknown settings. It is challenging to create an appropriate mathematical model of the system to construct its controller due to the non-linear properties of mobile robots [5]. A differential drive is a type of drive utilized by many mobile robots with wheels, in which each wheel is powered by a separate actuator. As a result, the direction of the wheeled mobile robot can be controlled by changing the linear or angular speed of the left and right wheels. Because controlling the angular velocity requires prior information from the robot's kinematic model [6], which is composed of various and complex mathematical terms, controlling the linear velocity is a simple solution for controlling the robot's direction because it does not require any mathematical model.

Because there are often static or even dynamic obstacles in their route, obstacle avoidance is a crucial strategy for the design and implementation of mobile wheeled robots. When several robots circulate in the same space, they operate as barriers between them [7][8]. As a result, obstacle avoidance research has been a hot topic in the realm of mobile robotics, and several strategies for avoiding static and/or dynamic limitations have been developed [8-10].

Experiments on control have generally focused on tackling the following challenges to strengthen the independence of mobile robots: mobile robot placement, stabilization, trajectory tracking control, trajectory planning, and obstacle avoidance. Significant (but not absolute) progress has been made in this area. The current research focuses on the obstacle avoidance problem, which aims to provide a mobile robot with collision-free navigation through a workplace with a predetermined distribution of obstacles. The MATLAB-Simulink application was used to construct an experimental version of a WMR controller that provides collision-free navigation.

In general, planning the trajectory of non-holonomic robots in the field of robotics entails determining the path from the start point to the end point while taking into account mechanical constraints and ensuring that the robot follows the desired path. Path tracking and posture stabilization are the two basic methods of controlling mobile wheeled robots.

1.2. Statement of the Problem

The trajectory tracking case is solved using a variety of classic ways. Unfortunately, most approaches will produce a speed leap if the fault changes suddenly [11]. This thesis provides an ANFIS-based trajectory tracking controller solution to handle this problem. The adaptive neuro-fuzzy inference system (ANFIS) was established by combining neural networks and fuzzy inference systems to solve the limitations of the trajectory tracking problem for mobile robots. Mobile robots or various applications are used by people from many walks of life. Other industries have begun to contemplate adopting mobile robots with wheels in various applications as new technologies progress. The obstacle avoidance system ensures the safety of the mobile wheeled robot and its surroundings. This will save money on maintenance and will not increase the mobile robot's maintenance or repair costs. Obstacle-avoidant mobile robots are not secure against obstructions or other mobile robots. It may cause accidents and significant losses in critical situations.

1.3. Objectives of the Study

1.3.1. General Objectives

The main objective of this study is to model and design a mobile robot path tracking and obstacle avoidance system based on the neuro fuzzy technique.

1.3.2. Specific Objectives

The specific objectives of this thesis are:

- To develop mathematical models of a mobile robot.
- To develop a fuzzy system for path tracking and obstacle avoidance.
- To design and train an ANFIS system based on the designed fuzzy system.
- To verify the algorithm by using MATLAB.
- To make conclusions and recommendations for the results of the proposed system.

1.4. Scope of the Study

Using an Adaptive Neuro Fuzzy Inference System (ANFIS) controller, this research mainly focused on the analysis and simulation of path tracking and obstacle avoidance systems for mobile robots. It goes over how to describe a mobile robot mathematically, how to create an adaptive neuro-fuzzy inference system for path tracking, and how to program the controller in MATLAB.

1.5. Limitations of the Study

Because of the difficulty to get the components for the implementation of the prototype system the thesis is only shown in the simulation using MATLAB/Simulink. In addition to this it needs extra additional time and cost for real implementation.

1.6. The motivation of the Study

Medical assistance, space and marine exploration, leisure and entertainment, technology research and development, metal-mechanics and chemical industries, military research, cleaning in a variety of environments, agriculture, inspection, security, and transportation are just a few of the applications for the wheeled mobile robot. The WMR must be aimed towards the correct controller in order to gain these benefits. As a result, the research team is now focusing on creating a mobile robot controller.

1.7. Significance of the Study

A wheeled mobile robot can move independently without the assistance of a person and is suited for a variety of industries, including aerospace, construction, chemicals, and manufacturing. One example of a high-temperature environment in which mobile robots are necessary is in the manufacturing process. This proposed system allows the wheeled mobile robots to be efficient in its path to track effectively and enables the robot to avoid unnecessary collision with the obstacle. So, those wheeled mobile robots must be controlled with efficient and active controller

1.8. Thesis Organization

This thesis is arranged in to five chapters. The first chapter introduces an introduction for the research works, statement of the problems, objectives, significance, scope and limitations of the thesis. The second chapter describes the literature reviews on the path tracking and obstacle avoidance techniques. The third chapter explains the methodology and mathematical modeling of a mobile robot. The fourth chapter presents the controller design. After all the fifth chapter explains system results and discussions. At the end the last chapter is all about the conclusion and recommendation for the future work.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

There has been a lot of interest in self-sustaining industrial vehicles in recent years, as you may have seen. Such cars' control architecture must account for all potential disruptions during their missions. The control system must be able to respond fast and adapt to changing situations. Many "traditional" control system design procedures for such vehicles (optimal control methods, algebraic approaches) necessitate a description of the physical system and all of its parameters. A proper dynamic model must be created in order to accurately characterize the mobile robot system. The dynamic model allows you to take into account factors like mass, inertia, friction, and centrifugal force. This model is used to learn more about the shape and operation of a controlled electromechanical system. When a complicated system must be constructed, developing a model becomes even more critical. The required dynamic model, when prepared and used correctly, can uncover problems and inaccuracies in the description of the real system (model) as quickly as feasible. They are easier and less expensive to adapt throughout the virtual construction process than real.

It is critical to understand the dynamics of mobile robots in order to create plans that are appropriate for their trajectory. To begin, use the route planning module to create a route that includes a series of gestures. Because the mobile robot can traverse the free working space under the purposeful path, this path is generated relative to the size of the movable robot. The dynamic qualities aren't relevant in this scenario. Only geometric space is used for path tracking, and the path planning requirement is that the mobile robot does not collide with any stationary obstacles. The algorithms A* and D* [11] are common route planning techniques for these applications.

The scenario alters when moving obstacles are introduced into the mobile robot workspace. They could be a group of people, or they could be mobile robots or machines. To avoid collisions, the mobile robot must reach a specified posture within a certain amount of time. In this situation, the mobile robot can avoid colliding with moving objects that must vary their speed on a regular basis. It can only ensure that collision-free mobility in a workplace with moving impediments in this fashion [12]. During the path design step, the mobile robot's dynamic properties are used to construct a set of speed curves that the robot can safely execute. In this process, the speed limit of the mobile robot must be respected while maintaining the curvature of the trajectory [13].

"A reprogrammable multipurpose manipulator, which is arranged to transfer specialized materials, parts, tools, or equipment through various programmed movements to fulfill a significant number of jobs," according to the robot's description. Commercial robots are also employed. "Some inflexible hyperlinks related to various collaborative media handled and monitored by electronic means," according to the description [14]. Robots' primary purpose is to assist people in performing hazardous, time-consuming, and unrelenting tasks. The goal has shifted to eliminate direct human-to-human contact with hazardous commercial production processes. Robots must be able to function independently in order to attain this goal.

2.2. Related Works on Path Tracking

The path tracking controller's purpose is to regulate the mobile robot's linear and angular velocity so that it can follow a predetermined path. The error between the reference path and the current position of the robots cannot be avoided because to slippage, disturbances, noise, vehicle-terrain interaction, and measurement errors of sensors, including external and internal sensors. As a result, properly tracking a specified trajectory with a robot's movement control was a critical issue. In recent years, there has been a lot of interest in route tracking research.

2.2.1. Backstepping Kinematics Based Approaches

The dynamic work for non-holonomic mobile robots was initially proposed in [15], and it is a mix of a kinematic controller and a torque controller. Lyapunov's theory was employed to assure stability in the development of a combined kinematic/torque controller. Path following, following a reference path, and stabilizing the reference posture are all challenges that the suggested control method can solve. In this approach the control technique gives a strong technique of converting the direction changing command into the control input to the real vehicle while taking into account the specific vehicle characteristics. First, the feedback speed control input is intended to asymptotically stabilize the position error in the motion steering system. Next, create a computed torque controller to ensure that the mobile robot's speed converges to a specified speed input. Because this manipulative method may reduce to a set of simple kinematic system control velocity inputs, the same design procedure may be used to solve all three basic navigation challenges.

It is impossible to have perfect understanding of mobile robot parameters. It can be difficult to model friction in these models, for example. As a result, no prior knowledge of the mobile robot's dynamic parameters is required because the controller will learn them online.

2.2.2. Sliding Mode Based Approaches

Depending on the observer's error function, the slide mode is also known as the slip error surface or hypersurface. The error-based operation will "slide" the observation error to zero once the tracking error path reaches this hypersurface. Presented non-holonomic wheeled mobile robots with effective tracking control [16,17]. To asymptotically stabilize the mobile robot to the appropriate trajectory, a novel slip control law is developed. The results reveal that the scheme is resistant to external intervention to a certain extent.

The solution benefits from using sliding mode control since it is fast, has a good transient response, and is robust to parameter fluctuations. The robot's behavior under limited interference is described using a dynamic model of the robot. The error dynamics of the mobile robot are linearized using the computed torque approach, and the sliding mode control law is used to stabilize the robot on the reference path and correct for existing disturbances. Essentially, the suggested control strategy employs the calculated torque approach for dynamic equation feedback linearization and a sliding mode theory for robust control.

The key drawback of this control strategy is that it necessitates the use of a detailed dynamic model of the mobile robot for the required torque component. We need to know the exact robot dynamics in order to accomplish the feedback linearization of the nonlinear dynamic model of the system.

2.2.3. Adaptive Based Approaches

The trajectory of a non-holonomic wheeled mobile robot is tracked using a dual adaptive dynamic controller in [18]. The controller assures that the tracking error asymptotically converges to zero in tracking issues. Before applying tracking control, the trouble is converted into an analogous tracking problem using time-varying error feedback for stability. The developed controller assures that the stable error asymptotically returns to zero. A speed/acceleration limiter is included in the proposed control law to prevent the robot's wheels from slipping.

2.2.4. Neural Network-Based Approaches

Neural networks had been proven to be a best solution to handling a difficult control problems in recent years. In [19-22] proposes the use of neural network controllers to follow the route of mobile robots.

The author in [20] proposes a neural network path tracking controller that exploits the neural network's learning property to create smoother tracking performance. The backstepping controller is combined with compound orthogonal networks in this new control strategy, which increases performance by utilizing the neural network's learning capabilities.

The controller suggested in [21] is based mainly on the approximation properties of neural network functions, which can handle bounded unmodeled disturbance and unstructured unmodeled dynamics of mobile robots. To understand the whole dynamics of the mobile robot and transform the reversing controller's speed output into the torque input of the actual vehicle, the neural network is integrated with the reversing controller. The merit of using neural networks in this technique is that you don't need to know the robot's dynamic model and the neural network doesn't need to know the dynamics a priori.

2.3. Related Works on Obstacle Avoidance

Many scholars have recently become interested in impediment avoidance issues, resulting in the development of real-time methods for slim and congested areas. An obstacle avoidance system is designed to help a robot navigate through an unfamiliar area by avoiding collisions. The obstacle avoidance robot detects a restriction in the direction and adjusts its course accordingly.

The article in [23] describes how a mobile robot obstacle avoidance system was successfully turned into a blind navigation aid. The impediment avoidance system communicates information to the human via stereophonic warnings rather than delivering digital alerts to the robotic movement controllers. These alerts contain spatial data such as item positions in space or journey directions and speeds. The advanced concept of pre-crash sensing is discussed in [24]. Ultrasonic sensors are utilized to detect front objects and automatically apply brakes to avoid vehicle collisions during working situations.

The article in [25] depicts that a neural manipulation system in a mobile robot that can comprehend unintelligible information in order to avoid obstacles. In [26] states that to stereo vision sensors used to extend the real-time obstacle avoidance system of autonomous mobile robots in this paper. A method for detecting obstacles is proposed. The goal of this study is to see if it is possible to avoid barriers on your own in typical situations. A dependable sensor and a sufficient routing system are necessary to do this.

In [27] it shows coverage maps as a new way to depict the surroundings for a mobile robot in this article. The coverage map retains the standard occupancy grid regarding the amount of the corresponding unit covered by barriers for each unit of the subsequent specified period, and the mobile robot can accurately infer the uncertainty in the environment map.

Table 2.1 Summary of Related Works

No	Title	Description
1	Back stepping kinematics based approach	Combination of a kinematic controller and a torque controller and no need of prior information on dynamic parameters.
2	Adaptive based approaches	For tracking problems, the controller guarantees the asymptotic convergence of the tracking error to zero.
3	Collision avoidance system for automobiles using Embedded system	The microcontroller unit sends a signal to the braking unit for applying the brake automatically.
4	Obstacle avoidance of autonomous mobile robot using stereo vision sensor.	Totally Based on stereo measurement without searching any factors.
5	Neural Network based approaches	There is no need to know the dynamic model of the robot and the NN without a prior knowledge for the dynamics.

CHAPTER 3

MODELING OF A MOBILE ROBOT

Sensors such as ultrasonic sensors, infrared beams, and cameras are commonly used by robots to detect their surroundings. However, due to related concerns such as ambiguity in the processing environment, they are frequently unreliable. A fuzzy decision controller, also known as a fuzzy logic controller, is used in this project (FLC). This type of fuzzy logic can be used to control a nonlinear process, referred to as a plant in control engineering terms. FLC has the advantage of allowing control engineers to quickly create control techniques that can be used by humans.

The modeling of a differential drive non-holonomic WMR to derive the equations employed in this development is also briefly discussed in this chapter. Understanding and a proper representation of a "model" of the system are required for the construction and control of a robotic system. This is the mechanical system's fundamental research. Kinematic modeling is concerned with the system's geometric relationships and explores the mathematics of motion without taking into account the forces that create motion.

3.1. Methodologies

The method to finish the task is described in Figure 3.1.

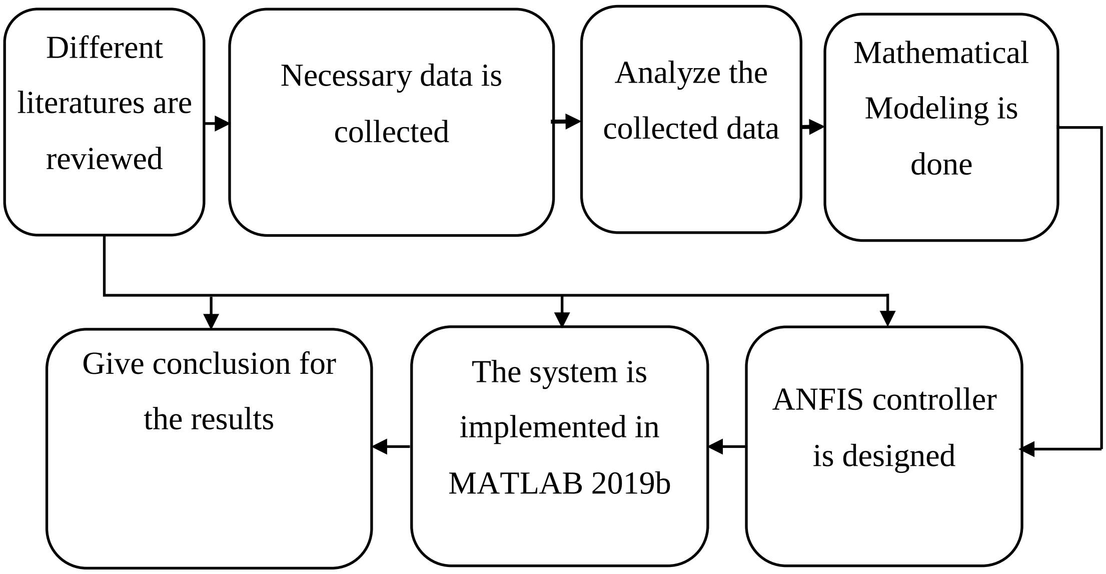


Figure 3.1 Block diagram of the system

3.2. Mathematical Modeling of a Wheeled Mobile Robot

3.2.1. Introduction

To derive the equations employed in this work, this section briefly presents differentially driven non-complete WMR modeling. Understanding and accurately representing the "blue print" of a robot system is required for its development and control. Any model is a representation of the current system that is idealized. Kinematic modeling, which is the basic research of the mechanical system, is included in the modeling of the differential drive wheeled mobile robot. Kinematic modeling is concerned with the system's geometric relationships and examines motion mathematics without taking into account the forces that generate motion.

Understanding robot mobility begins with a description of each wheel's contribution to motion. Each wheel is responsible for moving the robot as a whole. Similarly, each wheel restricts the robot's movement; for example, it won't allow it to slide sideways. The following part introduces the notation that allows robotic movement to be expressed in both a global and a local reference frame for the robot. This nomenclature is then used to explain the building of simple forward kinematic models of motion, which describe the robot's overall movement as a function of its geometry and individual wheel behavior.

3.2.2. Kinematic Modeling of the Mobile Robot

A bottom-up approach is used to create a model for the entire robot's movement. Each wheel contributes to the robot's movement while also limiting its movement. Because the wheels are coupled according to the robot chassis' geometry, their limitations combine to provide a constraint on the robot chassis' entire motion. Every wheel’s force and limitation, however, must be defined in terms of a consistent and explicit reference frame. Because of its self-contained and movable character, this is especially important for mobile robots; a precise mapping between global and local frames of reference is necessary. It begins with the formal definition of these frames of reference.

3.2.2.1. Mobile Robot Position Representation

The robot is modeled as an inflexible body on wheels that works in a horizontal plane during this examination. The robot chassis on the plane has three degrees of freedom (DOF), two of which are used for locations in the plane and one for alignment along a vertical axis perpendicular to the plane. Of course, there are extra degrees of freedom and flexibility due to the axles, wheel steering joints, and castor joints. The robot chassis, on the other hand, simply refers to the robot's inflexible body, ignoring the internal joints and degrees of freedom and flexibility. The robot chassis, on the other hand, simply refers to the robot's rigid body, ignoring the robot's internal joints and degrees of freedom, as well as its wheels. The initial stage in kinematic modeling analysis is to choose the coordinate systems.

i. Coordinate Systems

The primary advantage of the coordinate is to represent the robot's position. There are two types of coordinates for the derivation model: Cartesian and Polar coordinates. As a result of Cartesian coordinates' ubiquity, modeling is done in Cartesian coordinates.

a) Global Inertial Frame $\{X_I, Y_I\}$: It defines any inertial basis in the plane as a global reference frame from a certain origin O.

b) Local Robot Frame $\{X_R, Y_R\}$: This coordinate is connected to the robot and defines two axes related to P on the robot chassis.

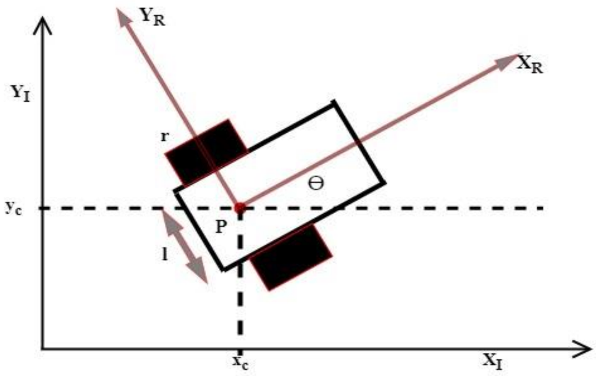


Figure 3. 2 The global reference frame and the robot local frame coordinate systems [28]

As illustrated in Figure 3.2, a relationship is created between the aircraft's global reference system and the robot's local reference system in order to identify the robot's position in the plane. Select point P on the robot chassis as your position reference point to specify the robot's position. P's position in the global reference system is provided by coordinates, but the angular difference between the global and local reference systems is given by. These three elements can be put together to make a vector that depicts the robot's pose.

Note, the use of the index I and R to clarify the basis of this pose as an inertial reference frame and local reference frame of the robot respectively:

$$q_I = [x_I \; y_I \; \theta_I]^T \tag{3.1}$$

$$q_R = [x_R \; y_R \; \theta_R]^T \tag{3.2}$$

It is necessary to map the movement along the axis of the global reference frame with the movement along the axis of the robot's local reference frame in order to describe the robot's movement with the movement of the components. The typical orthogonal rotation transformation is used to put these two frames together:

$$q = R(\theta) q_l \tag{3.3}$$

Where,

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

This matrix is called the orthogonal rotation matrix and can be used to map the motion in the global reference system according to the motion of the local reference system.

3.2.2.2. Differential Drive Kinematics

A differential drive, which consists of two drive wheels mounted on a shared shaft and each of which can be moved forward or backward independently, is used by many mobile robots. A chassis with two fixed wheels is one of the simplest ways to create mobile robots. Understanding this design aids in the understanding of some basic robot kinematics. Differential drives usually need mobile robots to have an additional castor wheel for stability. Rollerballs are sometimes used, however there are no variations in calculations in terms of kinematics.

The wheel calculation can be omitted because it has no effect on the robot's kinematics because it can rotate freely in all directions. In the case of differential drive, the mobile robot must rotate around a point that is the common axis of the two driving wheels in order to avoid skidding and only have a pure rolling motion. The instantaneous center of rotation (ICR) or instantaneous center of curvature is the name given to this place (ICC).

By changing the speed of the two wheels, the ICR will move and follow a different path.

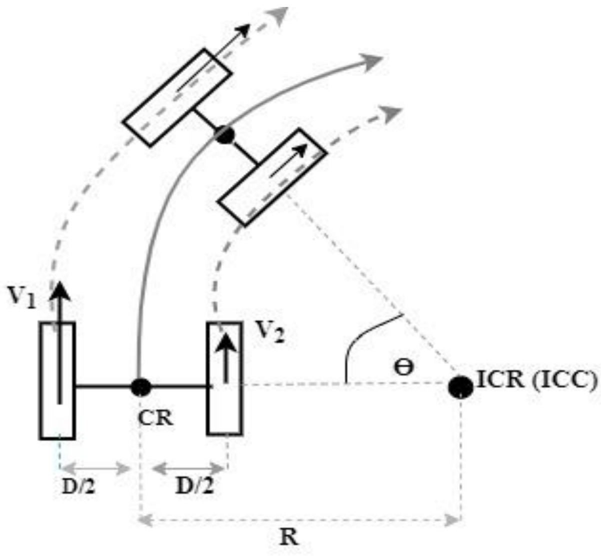


Figure 3. 3 The differential drive motion example [29]

At every moment in time, the left and right wheels follow a path (Figure 3.3) that moves around the ICR with the same angular rate ω = dθ/dt. Because the rate of rotation ω about the ICR must be the same for both wheels, the following equations can be written as:

$$V_{CR} = \omega R \tag{3.5}$$

Therefore, based on equation (3.5), the following equations can be derived as:

$$V_1 = \omega(R + \frac{D}{2}) = (V_{CR} + \frac{D}{2}\omega) \tag{3.6}$$

$$V_2 = \omega(R - \frac{D}{2}) = (V_{CR} - \frac{D}{2}\omega) \tag{3.7}$$

Where $V1$ and $V2$ are the linear speeds of the left and right wheels respectively; VCR is the linear velocity at the CR point of the platform; ω is the rotation speed of the platform; D is the distance between the two wheels, and R is the signed distance from the ICC to the midpoint between the two wheels.

Note that: V_1, V_2, and R are all functions of time.

The linear speed at point CR, that is, the midpoint between the two wheels, can be calculated as the average of the speeds V1 and V2.

$$V_{CR} = \frac{V_1 + V_2}{2} = \frac{R}{2}(\Omega_1 + \Omega_2) \tag{3.8}$$

At any moment in time ω and R can be solved as:

$$\omega = \frac{V_2 - V_1}{D} = \frac{R}{D}(\Omega_2 - \Omega_1) \text{ and } R = \frac{D}{2}\frac{V_2 + V_1}{V_2 - V_1} \tag{3.9}$$

Therefore, from equations (3.8) and (3.9), the angular wheel speed can be:

$$\Omega_L = \frac{1}{R}\left(V_{cr} - \frac{D}{2}\omega\right) \tag{3.10}$$

$$\Omega_R = \frac{1}{R}\left(V_{cr} + \frac{D}{2}\omega\right) \tag{3.11}$$

Where Ω_L and Ω_R are the rotational velocities of the left and right wheels respectively. There are three interesting cases with these kinds of drives. If $V_1 = V_2$, then we have forward linear motion in a straight line. R becomes infinite, and there is effectively no rotation, i.e., ω is zero (Figure 3.4a). For different values of V_1 and V_2, the mobile robot does not move in a straight line, but moves along a curved path along a point with a distance of R from CR (Figure 3.4b and 3.4c around a rotating wheel), changing the position of the robot and direction. If $V_1 = -V_2$, then $R = 0$, and we have a rotation about the midpoint of the wheel axis (Figure 3.4d).

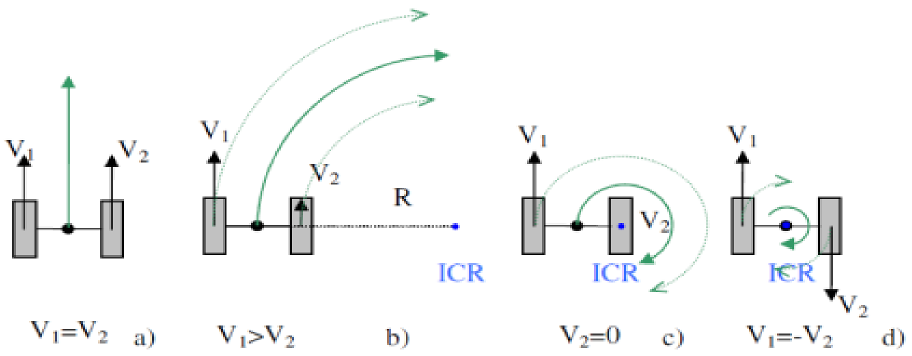


Figure 3. 4 The different moving possibilities for a differential drive [29]

a) Motion forward or backward, R is infinite, c) Turning R=D/2,

b) Turning R>D/2, d) Turning R=0;

The differential drive mobile robot is very sensitive to the relative speed of the two wheels. Small errors in the relative speed between the wheels can affect the trajectory of the robot. They are also very sensitive to small changes in ground level and may require additional wheels for balance.

i. The Differential Drive Odometry

Let's try to find formulas using which is possible to compute the actual position of the robot.

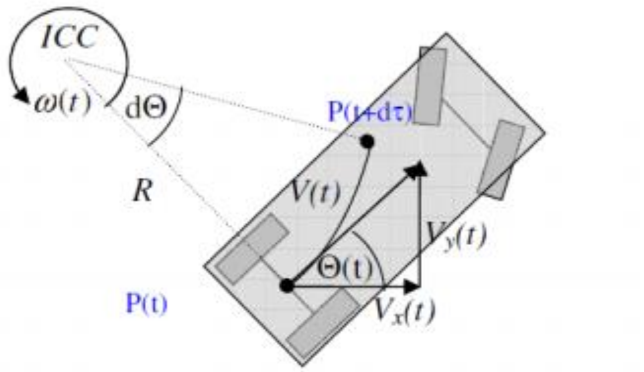


Figure 3. 5 Odometry for a differential drive [29].

Consider a differential drive robot revolving at an angular velocity (t) around the point ICC. The robot will travel the distance from point P (t) to point P (t+dt) with a linear velocity V(t) during the infinite short period. V (t) has two perpendicular components, one on the X-axis (Vx (t)) and one on the Y-axis (Vy) (t). The robot moves along a straight-line tangent to the real robot trajectory at point P (t) for an endlessly brief time. The traveled distance in either direction can be estimated using the two components of the velocity V (t) as follows [29]:

$$d_x = V_x(t)d\tau \qquad (3.12)$$

$$d_y = V_y(t)d\tau \qquad (3.13)$$

Where,

$$V_x(t) = V(t)\cos\theta(t) \qquad (3.14)$$

$$V_y(t) = V(t)\sin\theta(t) \qquad (3.15)$$

Similarly, the angle of the rotation can be obtained as follows:

$$d\theta = \omega(t)d\tau \qquad (3.16)$$

By integrating equations (3.12), (3.13) and (3.16), the time domain can be:

$$x(t) = \int_0^t V_x(\tau)d\tau + x_0$$

$$y(t) = \int_0^t V_y(\tau)d\tau + y_o \qquad (3.17)$$

$$\theta(t) = \int_0^t \omega(\tau)d\tau + \theta_o$$

Where, (x_0, y_0, θ_0) are the initial pose.

Using equations (3.14) and (3.15), the equation (3.17) can be rewritten as:

$$x(t) = \int_0^t V(\tau)\cos\theta(\tau)d\tau + x_0$$
$$y(t) = \int_0^t V(\tau)\sin\theta(\tau)d\tau + y_0 \quad (3.18)$$
$$\theta(t) = \int_0^t \omega(\tau)d\tau + \theta_0$$

Formulas in equation (3.18) are valid for all robots capable of moving in a particular direction (t) at a given velocity V (t). For the special case of differential drive robot, based on equations (3.8) and (3.9), equation (3.18) can be rewritten as:

$$x(t) = \frac{1}{2}\int_0^t \left[V_1(\tau) + V_2(\tau)\right]\cos\theta(\tau)d\tau + x_0 \quad (3.19)$$

$$y(t) = \frac{1}{2}\int_0^t \left[V_1(\tau) + V_2(\tau)\right]\sin\theta(\tau)d\tau + y_0 \quad (3.20)$$

$$\theta(t) = \frac{1}{D}\int_0^t \left[V_2(\tau) - V_1(\tau)\right]d\tau + \theta_0 \quad (3.21)$$

3.2.2.3. Forward Kinematic Models

In the simplest case, the mapping defined by equation (3.3) is sufficient to generate a formula to capture the direct kinematics of the mobile robot. More formally, consider the example shown in Figure 3.6. The goal of modeling the robot's advanced kinematics is mainly to determine the robot's speed as a function of the wheel speed.

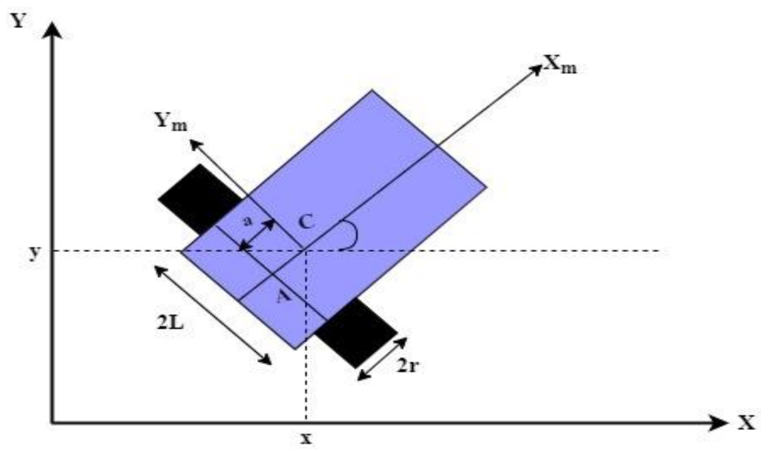


Figure 3. 6 The differential drive mobile robot model

The forward kinematic problem can be described as the problem of finding the robots overall speed in the global reference frame as the following function:

$$q_I = \begin{bmatrix} x & y & \theta \end{bmatrix}^T = f\,(wheels\ \text{speed,geometric parameters}) \\ q_I = \begin{bmatrix} x & y & \theta \end{bmatrix}^T = f\,(\Omega_R, \Omega_L, L, r, \theta) \tag{3.22}$$

First compute the contribution of each wheels in the local reference $\dot{q}R$. First, consider the contribution of each wheel's spinning speed to the translation speed at A in the direction of +xm. If one wheel turns and the other wheel makes no contribution and stands still, because A is in the middle of the two wheels, it will immediately move at half the speed: $x_{mR} = \frac{r}{2}\Omega_R$ and $x_{mL} = \frac{r}{2}\Omega_L$. In a differential drive robot, you can simply add these two contributions to calculate the component $\dot{x}_R$ of $\dot{q}R$. Neither wheel can move laterally in the robot's frame of reference, so $\dot{y}R$ is always zero.

Consider the right wheel, the forward spin of this wheel results in counterclockwise (CCW) rotation at point A. If the right wheel turns on its own, the robot turns around the left wheel. The rotational speed at A can be calculated ω_R, because the wheel moves instantaneously along an arc with a radius of 2L: $\omega_R = \frac{r}{2L}\Omega_R$. The same calculation applies to the left wheel, the difference is that the forward rotation will give resulting in a clockwise (CW) rotation at point A: $\omega_L = \frac{r}{2L}\Omega_L$.

Therefore, by combining these individual formulas yields the translational speed and the rotational velocity in the robot reference frame as follows:

$$x_R = \frac{r}{2}\left(\Omega_R + \Omega_L\right) \tag{3.23}$$

$$y_R = 0 \tag{3.24}$$

$$\theta_R = \frac{r}{2}\left(\Omega_R - \Omega_L\right) \tag{3.25}$$

From equation (3.3), it is known that the robot's motion in the global reference frame from motion in its local reference frame can be computed as: $q_I = R(\theta)^{-1} q_R$.

Therefore, the forward kinematic model for the differential-drive mobile robot in the inertial frame is:

$$q_I = R(\theta)^{-1} \begin{bmatrix} x_R \\ y_R \\ \theta_R \end{bmatrix} \tag{3.26}$$

$$q_I = R(\theta)^{-1} \frac{r}{2} \begin{bmatrix} \Omega_R - \Omega_L \\ 0 \\ \frac{\Omega_R - \Omega_L}{L} \end{bmatrix} \tag{3.27}$$

Where,

$$R(\theta)^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.28}$$

Therefore, the robot velocity in the global inertial frame is:

$$q_I = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \frac{r}{2} \begin{bmatrix} \Omega_R + \Omega_L \\ 0 \\ \frac{\Omega_R - \Omega_L}{L} \end{bmatrix} \tag{3.29}$$

Finally, rewriting Equation (3.29) yields

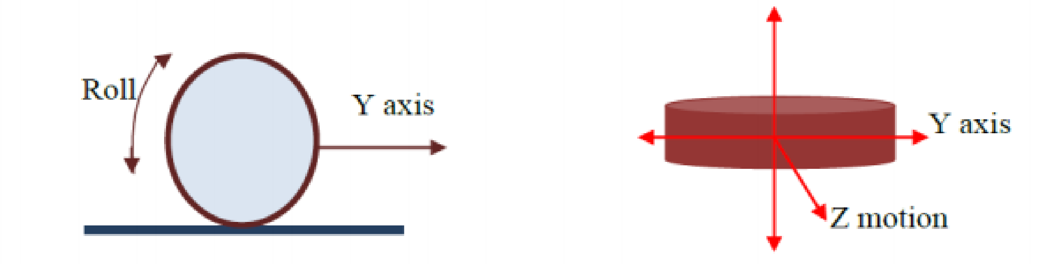
$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\theta & \frac{r}{2}\cos\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \\ \frac{r}{2L} & -\frac{r}{2L} \end{bmatrix} \begin{bmatrix} \Omega_R \\ \Omega_L \end{bmatrix} \tag{3.30}$$

In simple circumstances, this kinematic modeling method can provide information about the robot's movement based on the speed of the wheels that make it up. To calculate the possible movement space of each robot chassis design, however, one must go a step further and formally explain the limits placed on the robot's mobility by each wheel.

3.2.2.4. Types of Wheels

The motors that are powered by the output of a microprocessor which is based on sensor information can travel without the assistance of external human drivers. When the wheeled robot moves, there are two constraints for each type of wheel.

As demonstrated in Figure 3.7, the main limitation strengthens the rolling contact idea (a). As seen in Figure 3.7, the second constraint strengthens the concept of no lateral slip by requiring the wheel to not slide perpendicular to the wheel's plane (b). The constraints on the motion of a single wheel are expressed in the first stage of the robot kinematics model. As a result, by combining the movements of a single wheel, the movement of the complete robot may be estimated.



a) Rolling Motion b) Lateral slip

Figure 3. 7 Types of wheels

The wheels can be classified into five categories based on geometric constraints: fixed standard wheels, standard steering wheels, swivel wheels, Swedish wheels, and spherical wheels. Because the internal wheels can move freely in all of these scenarios, kinematic limits on the robot chassis do not apply to castor wheels, Swedish wheels, or spherical wheels. Only constant standard wheels and steerable standard wheels have an impact on robotic chassis kinematics; hence they must be considered when computing robotic kinematic constraints.

3.2.2.5. Wheel Kinematic Constraints

The primary stage in the robot kinematics model is to represent the constraints on each wheel's movement. Mechanical mobile robots, often known as "two-wheel differential drive mobile robots," usually have at least three wheels.

A shared horizontal axis connects the two separately controlled drive wheels, which is fixed during robot operation (relative to your body). These "drive wheels" ensure the mobile robot's mobility due to their angular velocity. The robot's equilibrium is maintained by the flywheel (a passive rotating wheel). It is mounted separately on the vertical axis rather than on the mobile robot body's driving axis. As a result of the force provided by merely the two "driving wheels" [31], the rotating wheels will align themselves on the course automatically and freely.

The vehicle rotates around the center of the axle due to the difference in speed between the two independently driven coaxial wheels, while the wheels work together to create forward or reverse movement. Because of the angular velocity of the two wheels are equal and opposite, those robots can rotate without changing the midpoint between the wheels. Mobile robots move at a slow pace and presume there is no vertical motion.

However, the analysis will be explained by a few key assumptions. Assume that the wheels remain upright throughout the robot's movement and that the ground plane's contact point with the wheels does not slip. This means that the wheel only revolves around the vertical axis through the contact point under pure rolling conditions.

There is no vertical rotation axis or steering axis on a typical fixed wheel. Because the angle between the chassis and the wheel axle remains constant, the robot can only move back and forth along the plane of the wheel and spin around its point of contact with the plane of the wheel.

The robot will have a specific form of a non-holonomic constraint, which will take into consideration the preceding assumptions about wheel motion. Non-holonomic restrictions are limits on an object's possible speed. It can simply mean that a differential drive mobile robot can move ongoing and backward directions while not in sideways, as shown in Figure 3.8.

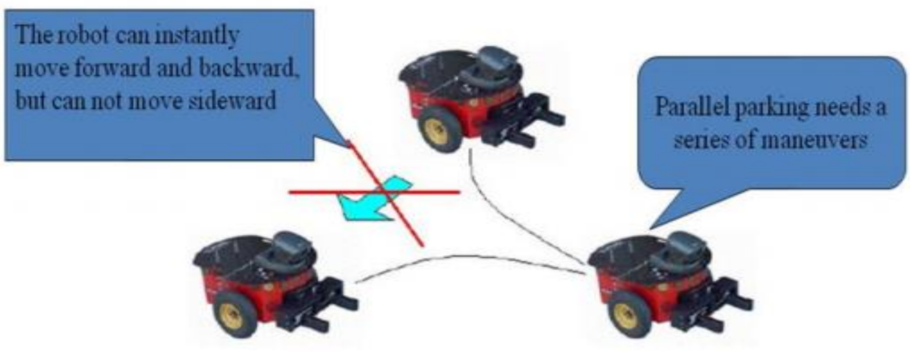


Figure 3. 8 The non-holonomic robot motion

3.2.3. Modeling of the Power Actuator Drive

3.2.3.1. Modeling of the Electrical Part

The study of the movement that shapes these forces, as well as the energy and speed associated with these movements, is known as dynamics. The DC motor is an important transducer that converts electrical energy into mechanical energy. A DC motor is the polar opposite of a generator, yet it can function as both a generator and a motor.

The input voltage Vin (s) is related to the output angular velocity (s) by the open loop transfer function of a DC motor without any additional load, which is given by the equation (3.39). Equations supply the total equivalent inertia J and total equivalent damping b in the motor armature (3.33). The mobile robot can be thought of as having a cuboid shape for the sake of simplicity [32].

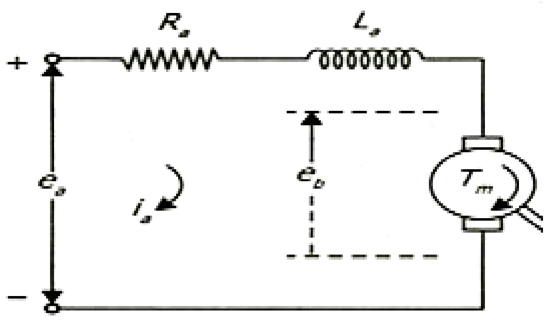


Figure 3. 9 DC Motor Electric Circuit

The relationship b/n the torque and armature current expressed as:

$$T(t) = k_t * I(t) \tag{3.31}$$

Where k_t is the motor torque constant in Nm/A

The back electromotive force $Vemf$ is proportional to the angular velocity of DC Motor

$$V_{emf}(t) = e(t) = K_b * w(t) = K_b * \frac{d\theta(t)}{dt} \tag{3.32}$$

Where, K_b is the back emf constant; $w(t)$ is the angular velocity; $\theta(t)$ is the angular displacement

$$T(t) = K_t(t) * I(t) = b * w(t) + J * \frac{dw(t)}{dt} \tag{3.33}$$

Where, b is the damping inertia of the mechanical system

$$u(t) = e(t) + R_a * I(t) + L_a * \frac{dI(t)}{dt} \tag{3.34}$$

Where, u(t) is the input voltage; R_a is the Resistance; L_a is the inductance of armature; J is a moment of inertia.

Appling Laplace transform for Eq. (3.33) and Eq. (3.34)

$$K_t * I(s) = b * w(s) + JS * w(s) \tag{3.35}$$

$$u(s) = e(s) + R_a * I(s) + L_a S * I(s) \tag{3.36}$$

Rewrite Eq. (3.35)

$$I(s) = \frac{b * w(s) + JS * w(s)}{K_t} \tag{3.37}$$

$$u(s) = e(s) + R_a * I(s) + L_a S * I(s), \quad e(s) = K_b w(s)$$

$$u(s) = K_b * w(s) + R_a * I(s) + L_a S * I(s) \tag{3.38}$$

Substitute Eq. (3.37) into Eq. (3.38)

$$u(s) = K_b * w(s) + \frac{R_a * w(s)(b + JS)}{K_t} + \frac{L_a S * w(s)(b + JS)}{K_t}$$

$$\frac{u(s)}{w(s)} = \frac{K_t * K_b + (R_a + L_a S)(b + JS)}{K_t}$$

$$Gv(s) = \frac{w(s)}{u(s)} = \frac{K_t}{K_t * K_b + (R_a + L_a S)(b + JS)} \tag{3.39}$$

$$Gv(s) = \frac{w(s)}{u(s)} = \frac{K_t}{K_t * K_b + (R_a b + R_a J)S} \tag{3.40}$$

Equation 3.40 is the transfer function of the motor dynamics.

To obtain maximum accuracy while developing the mathematical model of the mobile platform system, it is required to observe and evaluate the dynamics between the road, wheels, and platform, as well as take into account all forces acting on the mobile platform system [32]. Various forces act on the mobile platform during operations. The balance between the forces acting on the platform during operation and the forces classified as road loads and traction forces is included in the dynamic modeling of the mobile platform system.

3.2.3.2. Modeling of the Driving Part

Mobile platforms are typically supported by two driving wheels (e.g., rear wheels) plus one or two front caster wheels for added stability (s). Because their torque-speed characteristics can be achieved with different electrical configurations, and their speeds can be smoothly controlled and, in most cases, are reversible, direct current (DC) motors that are coupled directly or indirectly (via velocity reduction gears) with the driving wheels are typically used to actuate mobile platforms. This section gives a full physical model of a DC motor linked directly to a driving wheel and identifies all of its parameters. A circuit diagram of actuator drive is shown below.

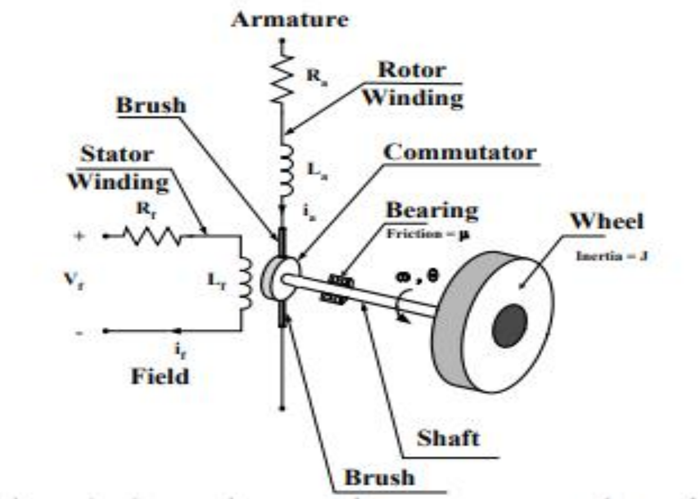


Figure 3.10 Circuit diagram of actuator drive

The stator and rotor windings are the two major components of a dc motor. The purpose of these components is to transform electrical energy into mechanical energy. There are two types of DC motors: induction and synchronous. First, armature-controlled dc motors with a constant field voltage (Vf) or a permanent magnate in place of the stator winding. Variation of the armature voltage (Va) or current is used to control them (ia). Second, field-controlled dc motors, in which the armature voltage (Va) is constant and the field voltage (Vf) or current is varies to regulate the armature voltage (Va) (if).

To achieve a linear model for DC motors, the hysteresis and voltage drop across the motor brushes are ignored, and the motor input voltage is supplied to the field or armature terminals [33]. As follows, the air gap flux (φ) is proportional to the field current (if):

$$\varphi = K_f i_f(t) \tag{3.41}$$

Where, K_f is the proportionality and t is time.

The developed torque (T_m) is assumed to be linearly related with φ and the armature current (i_a) (K_L is the proportionality constant) and substituting for air gap flux (φ) from Equation 3.41 yields:

$$T_m(t) = K_L \varphi i_a(t) = K_L K_f i_f(t) i_a(t) \tag{3.42}$$

Equation 3.42 shows that to have a linear dc motor, if or ia have to be maintain constant while the other current becomes the input current controlled motor.

Case 1: Field current controlled DC motor

In this case, the armature current (i_a) is constant and the field current (i_f) varies with time (t) yields:

$$T_m(t) = K_L \varphi i_a = (K_L K_f i_a) i_f(t) = K_m i_f(t) \tag{3.43}$$

Where: ($K_m = K_L K_f i_a$) is called the dc motor constant.

Applying Kirchhoff's law and using Laplace notation yields:

$$V_f(S) = (R_f + L_f s) I_f(S) \tag{3.44}$$

Where, S is the Laplace notation, V_f is input voltage, Rf is resistance, L_f is inductance, and $I_f = i_f$ is the stator current, see Figure 3.11. The motor torque **(**Equation 3.43) equals the torque delivered to the wheel and rotor (T_R) that can be modeled by:

$$T_R(s) = Js^2\theta(s) + \mu s\theta(s) \tag{3.45}$$

Where T_R is the torque, J is the inertia.

Equating Equation 3.43 with Equation 3.45, and substituting for $I_f(s)$ which yields the transfer function, G(s), of a field current controlled dc motor connected directly to a wheel:

$$G(s) = \frac{\theta(s)}{V_f(s)} = \frac{K}{s(\tau_f s + 1)(\tau_R s + 1)} \tag{3.46}$$

With, $$K = \frac{K_m}{\mu R_f} = \frac{K_L K_f I_a}{\mu R_f}, \tau_f = \frac{L_f}{R_f} \text{ and } \tau_R = \frac{J}{\mu}$$

Where: K is the system, τ_f is electrical time constant, and τ_R is the mechanical time constant.

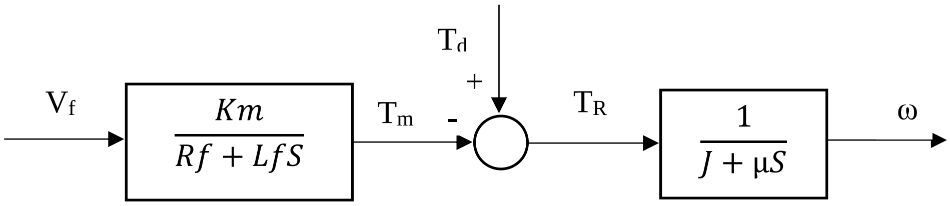


Figure 3.11 Block diagram of a field controlled drive

Where, T_m generated torque; T_d disturbing torque; T_R is driving torque; ω is the wheel's angular velocity.

Case 2: Armature current controlled DC motor

The field current (if) is constant and the armature current (ia) varies with time (t), yielding:

$$T_m(t) = K_L \varphi i_a = (K_L K_f i_f) i_a(t) = K_m i_a(t) \qquad (3.47)$$

Applying Kirchhoff's voltage law and using Laplace notation yields:

$$V_a(S) = (R_a + L_a S) I_a(S) + V_b(S) \qquad (3.48)$$

Where, V_a is input voltage, V_b is the back electromotive-force.

$$V_b(t) = K_b \omega(t)$$

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_m}{(R_a + L_a s)(Js + \mu) + K_b K_m} \qquad (3.49)$$

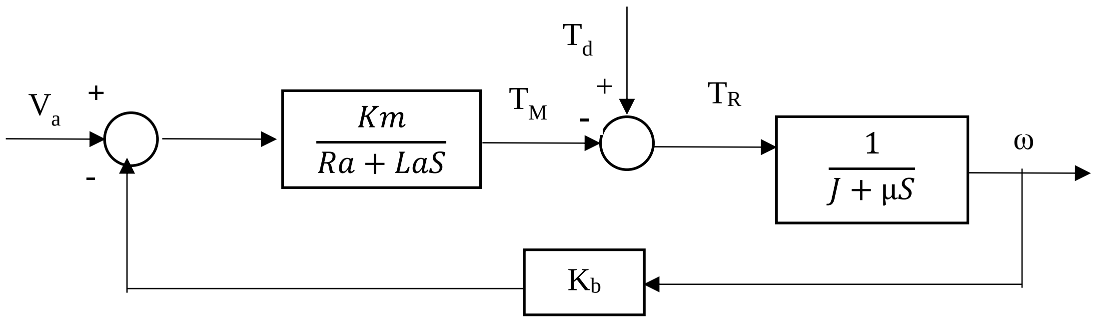


Figure 3.12 Block diagram of armature control drive

However, in most cases, the induction term *L* is negligible and if *J* is small, then Equation (3.49) can be simply rewritten as

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_m}{R_a \mu + K_b K_m} \qquad (3.50)$$

CHAPTER FOUR

CONTROLLER DESIGN

This chapter explains how to keep track of the road and stay on track without colliding with an obstacle. The robot model gained in Chapter 3 is used to create a fuzzy logic controller, which is then used to create an Adaptive Neuro Fuzzy Inference System (ANFIS).

4.1. Proposed Control System

A two-wheeled mobile obstacle avoidance robot is designed to maneuver in its surroundings. For distance information, object detection, collision avoidance, and navigation, three and eight proximity sensor models are employed to create input variables. The left and right motors are controlled by two of the output variables. The mobile robot features two wheels, one of which is utilized for robot stability and is propelled and controlled by a pulse width modulator to control a DC motor [33]. Using MATLAB software, simulations are run to assess the viability and effectiveness of the proposed control system. The study is based on the amount of time it takes to go from one spot to another.

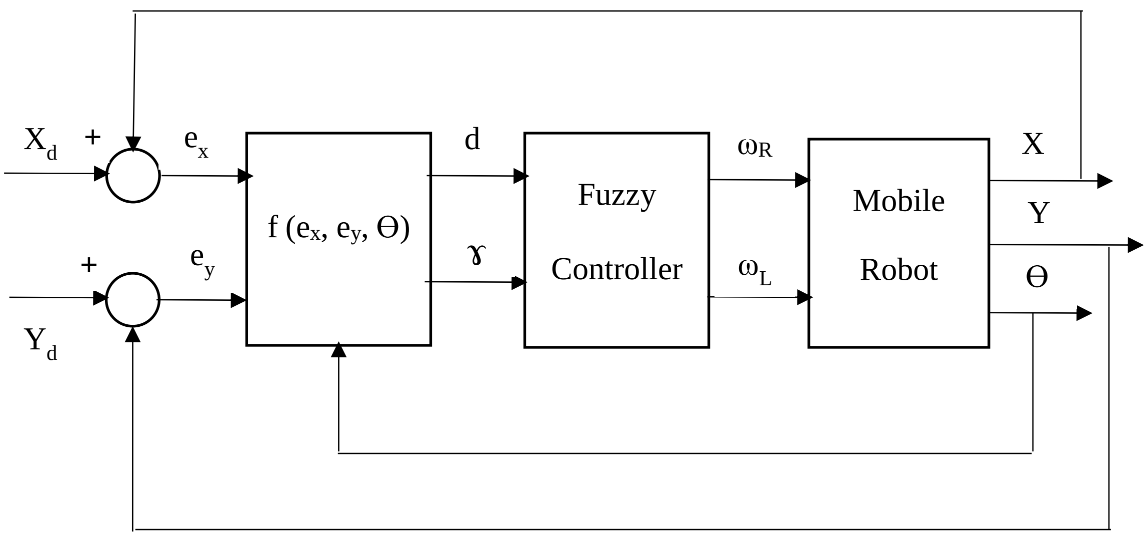


Figure 4. 1 Block Diagram of the System

4.2. Overview of Fuzzy Logic Controller

Here, we deal with the fuzzy controller needed to develop the path tracking and obstacle avoidance controller, which is the core of this study.

Fuzzification, knowledge base, fuzzy reasoning, and defuzzification are the main components of a traditional fuzzy logic system. Fuzzification changes the right input variables into fuzzy variables, which are input grades. The knowledge database stores relevant data and control rules. The knowledge base and the reasoning engine yield fuzzy outcomes when using fuzzy reasoning. Defuzzification is the process of converting ambiguous variables into precise output variables. Figure 4.2 depicts a typical architecture.

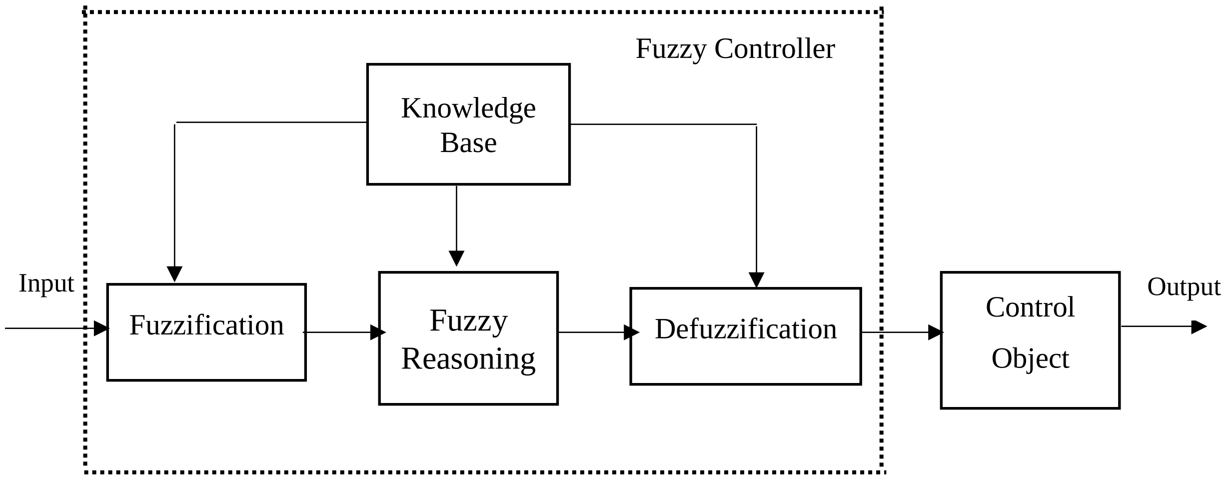


Figure 4.2 Typical Structure of a fuzzy logic system

The fuzzifier, inference engine, rule base, and fuzzified are the four essential components of fuzzy logic controllers. The Fuzzifier conducts input variable measurement, scale mapping, and defuzzification (input signals). All detected signals are scaled, and fuzzification refers to the conversion of the measured signal (clear input quantity) to a diffuse quantity (linguistic variable). Membership functions are used to accomplish this conversion.

The membership function describes the degree to which a quantity belongs to the fuzzy set and has a value between 0 and 1. Its value is 1 if the number is unconditionally decided to fit the fuzzy set, but it is 0 if the set is determined not to fit. To regulate the speed and steering angle of the robot's driving wheels, the trigger rule's output is mixed with fuzzy logic operations. These are represented by the left and right motors, which represent the speed of the robot's left and right wheels, respectively.

Fuzzy controllers can make use of information gathered from human operators. This is crucial in control issues where exact mathematical models are difficult or impossible to develop, and the acquired models are difficult or expensive to utilize. Inherent nonlinearities, the time-varying nature of the process to be regulated, massive unpredictable environmental disturbances, degrading sensors, and a variety of other factors all contribute to this challenge.

An experienced human operator's knowledge can be employed instead of a detailed model of the regulated process. The language description is made up of a series of fuzzy proposition-based control rules. The rule, which exemplifies the form of these rules,

If the temperature is very high

And the pressure is slightly low

Then the heat change should be slightly negative.

Where temperature and pressure are the process's observed state variables, and heat change is the controller's action to take. The fuzzy sets defined by the universes of discourse of temperature values, pressure values, and heat change values, respectively, can represent the vague terms very high, slightly low, and slightly negative. The fuzzy controller was built on the basis of this type of language rule.

4.2.1. Fuzzification Module

The input information is converted by the fuzzification module. The data is transformed into a format that the machine can search in its knowledge base for. This is accomplished by breaking the phrases down into simpler terms and extracting the essential terms, which are then passed to the inference engine to be processed further.

Fuzzification is the process of turning a precise number into a fuzzier one. This is accomplished by realizing that many quantities that are thought to be crisp, classical, and deterministic are not in fact deterministic. They do, in fact, carry a significant degree of risk. If the source of the uncertainty is ambiguity, vagueness, or imprecision, the variable is most likely fuzzy and can always be represented by a membership function.

4.2.2. Rule Base

A rule basis is present in every Artificial Intelligence system. Because the Fuzzy logic system is an AI-based system, it has its own rule base in which all of the information and data for the agent's reference is stored. A rule basis refers to a set of rules. The most common syntax for these rules is IF-THEN. The antecedent, premise, or condition of the rule is called the antecedent, while the THEN side is called the consequent, or conclusion.

The rules of the fuzzy logic set theory are kept in the fuzzy logic system's knowledge base. Their rules are represented by an if-then ladder. As a result, anytime the system tries to solve an issue, this if-then ladder is executed, and the system then works on the rule it obtains from the matched condition.

4.2.3. Inference Engine

The essential component of the fuzzy logic system is the inference engine. When compared to computer components, our inference engine is identical to the computer's processor. It is where all of the information processing takes place. The inference engine's job is to analyze and conclude all of the information it receives from the fuzzification module in order to produce a correct result. This is done again by referencing to the knowledge base's rules and preceding information. The final conclusions are then given to the defuzzification module for additional tweaking.

4.2.4. Defuzzification Module

Defuzzification is the process of converting a fuzzy amount to a precise number, while fuzzification is the process of converting a precise number to a fuzzy number. The inference engine sends the processed data to the defuzzification module. The conclusion is contained in this data, but it is not in the format in which it was received, i.e., in a user-friendly manner. As a result, the defuzzification module re-converts this data into a format that the user can understand. Now I'll try to describe each of the main processes needed in the design of a fuzzy controller using this simple control problem as an example.

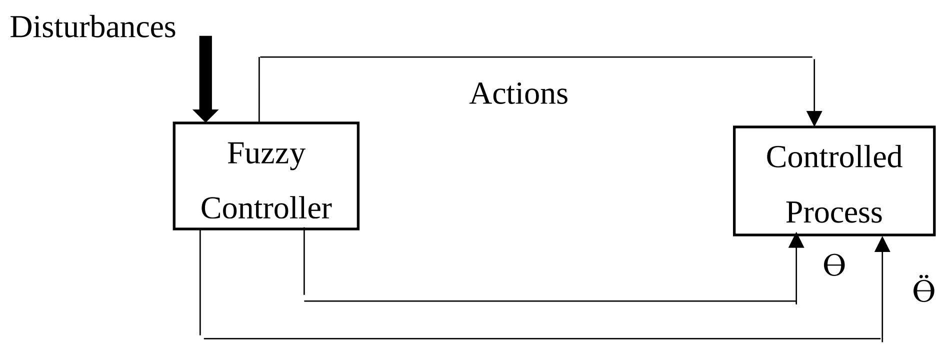


Figure 4.3 Design steps of fuzzy logic controller

The design involves the following steps:

Step1: After determining the controller's important input and output variables, as well as the range of values for each, we must choose linguistic states for each variables and represent them in the appropriate fuzzy set. These fuzzy sets are, in most situations, fuzzy numbers that represent linguistic categories like nearly zero, positive small, negative small, positive medium, and so on.

Now in order to explain this step, assume that the ranges of the input variables e and $\dot{e}$ are [-a, a] and [-b, b] respectively and the range of the output variables v is [-c, c]. let the following seven linguistic states are selected for each of the three variables.

NL-Negative Large	PL-Positive Large
NM-Negative Medium	PM-Positive Medium
NS-Negative Small	PS-Positive Small
AZ-Approximately Zero	

On representing these linguistic states by triangular shape fuzzy numbers that are equally spread over each range, we will get the fuzzy quantization exemplified for variable e in this figure. For variables $\dot{e}$ and v, variable a is replaced with values b and c respectively.

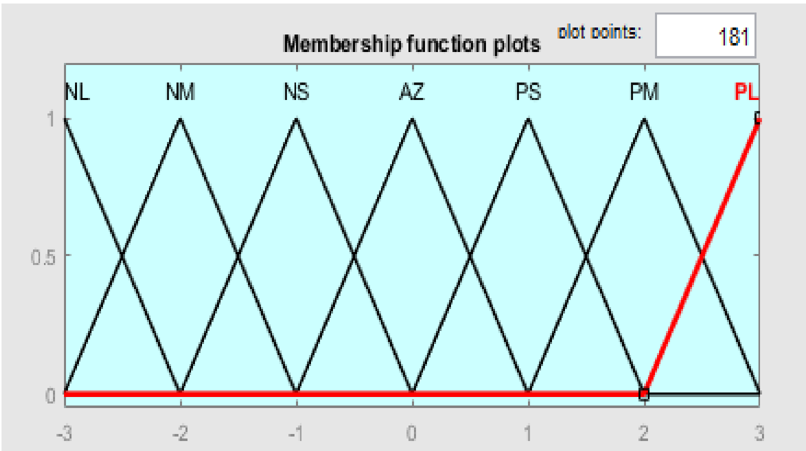


Figure 4.4 Membership function output

Possible fuzzy quantization of the range [-3, 3] by triangular shaped fuzzy numbers. Other shapes of membership functions are preferable to triangular shape for specific applications. The shapes need not be symmetric and need not be equally spread.

Step2: To reflect the measurement uncertainty associated with each input variable, a fuzzification function is introduced in this stage. The fuzzification function interprets measurements of input variables expressed as real numbers or more realistic fuzzy approximations of real values. Consider the variable e with the fuzzification function f_e applied. The fuzzification function then takes the following shape:

$$f_e = [-3, 3] \rightarrow R$$

Where R is the set of numbers of all fuzzy and $f_e(x_o)$ is number of fuzzy chosen by f_e as a fuzzy approximation of the measurement $e = x_o$.

Step3: The knowledge relating to the specified control problem is expressed in terms of a set of fuzzy inference rules in this stage. There are two main methods for determining closely similar inference rules. One method is to obtain them from skilled human operators. The alternative option is to derive them from empirical (experimental) data using an appropriate learning process and neural networks.

In the concerned example with variables e, $\dot{e}$ and v the inference rules have the canonical form of e=A and $\dot{e}$=B, then V=C.

Table 4.1 Example of fuzzy rule base

	$\dot{e}$							
	V	NL	NM	NS	AZ	PS	PM	PL
	NL	PL	PL	PL	PL	PM	AZ	AZ
	NM	PL	PL	PL	PL	PM	AZ	AZ
e	NS	PM	PM	PM	PS	AZ	NM	NM
	AZ	PM	PM	PS	AZ	NS	NM	NM
	PS	PM	PM	AZ	NS	NM	NM	NM
	PM	AZ	AZ	NM	NL	NL	NL	NL
	PL	AZ	AZ	NM	NL	NL	NL	NL

Step4: To create inferences about the output variables, measurements of the input variables of a fuzzy controller must be properly paired with applicable fuzzy information rules. This is what an inference engine is for. Some of the materials, i.e., knowledge acquired previously, can be directly utilized in the design of inference engines.

Now proceed with variables e, $\dot{e}$ and v as follows. Firstly, convert given fuzzy inference rules in to equivalent simple fuzzy conditional propositions of the form,

If (e, $\dot{e}$) is AxB, then v is C.

Where, [AxB] (x, y) =min[A(x), B(y)] for all x $\in$[-a, a] and all y $\in$[-b, b]. similarly, the fuzzified input measurements $f_e(x_o)$ and $f_e(y_o)$ can be expressed as a single joint measurement $(e_o, \dot{e}_o) = f_e(x_o) \times f_e(y_o)$.

When the fuzzy rule base consists of n fuzzy inference rules, the reasoning schema has the form:

Rule 1: If $(e, \dot{e})$ is A_1 x B_1, then v is C_1

Rule 2: If $(e, \dot{e})$ is A_2 x B_2, then v is C_2

Rule n: If $(e, \dot{e})$ is A_n x B_n, then v is C_n

Step5: The designer of a fuzzy controller must choose a suitable defuzzification method in the final step of the design process. Defuzzification is the process of converting each inference engine conclusion, which is expressed in terms of a fuzzy set, into a single real number. The resultant number, which defines the fuzzy controller's action, is not random. Any input measurements are included in the collection to be defuzzified which is,

$e = x_o$ and $e = y_o$ and C is defined by $C = U_j [f_e(x_o) \times f_e(y_o)] i_o R_j$.

4.3. Design of the Path Tracking Controller Using Fuzzy Logic Techniques

The acronym ANFIS is derived from the neuro-fuzzy adaptive inference system. The ANFIS controller is a combination of neural network and fuzzy logic. Many inputs are applied to the neural network, depending on the input that the neural network has some standard output. Input membership functions (MF) are assigned to input characteristics, input MF to an if-then rule set, rules to an output characteristic set, output characteristics to output MF, and output MF to a single-valued output or a decision using ANFIS. ANFIS combines the neural network and fuzzy systems approaches. You can get good reasoning in terms of quality and quantity if you mix these two strategies. To put it another way, fuzzy inference and network computing will be available concurrently.

The goal of control is to keep the robot on track with the specified reference by inputting proper wheel speeds (both left and right). To converge the difference between the robot's track P (x (t), y (t), q (t)) and the reference Pd (xd (t), yd (t), qd (t)) to zero (0). In actual life, getting precise parameters from a robot is tough. The neuro-purpose fuzzy's is to decrease the error between desired and real positions (xd-x, yd-y), as well as the error between the desired and real orientations (qd-q).

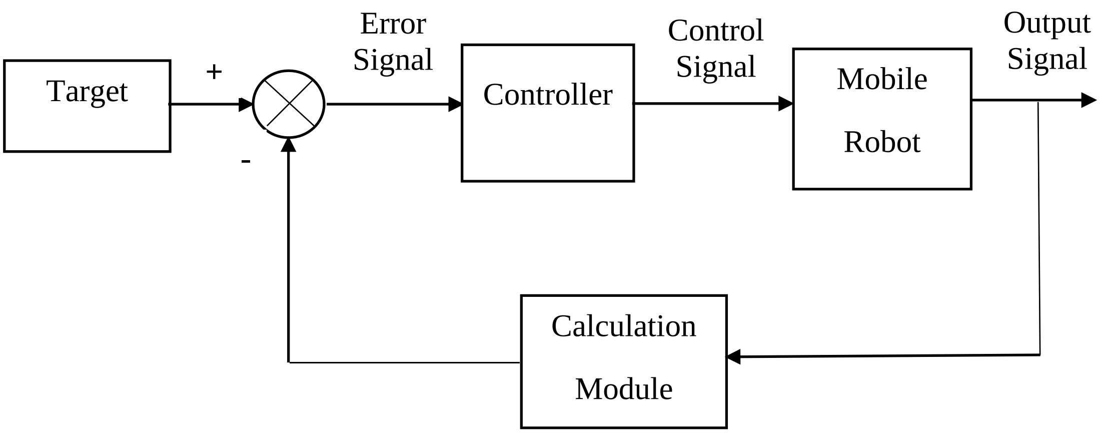


Figure 4.5 Control architecture

The feedback signal compares the real robot coordinates to the target coordinates and calculates the intended orientation as well as the angle error between the actual present orientation and the desired orientation, as shown in the equations below.

$$\theta_T = \tan^{-1}\left(\frac{Y_T - Y}{X_T - X}\right) \tag{4.1}$$

$$angle\ error = \theta_T - \theta \tag{4.2}$$

The real position of the mobile robot to the target position is calculated by:

$$D = \sqrt{(X_T - X)^2 + (Y_T - Y)^2}$$

The vector $\begin{pmatrix} X \\ Y \\ \theta \end{pmatrix}$ and $\begin{pmatrix} X_T \\ Y_T \\ \theta_T \end{pmatrix}$ represents the current position and the desired position of the mobile robot respectively.

For control problems at a given required position, reduce the distance and the deviation angle to zero to achieve the purpose of position control.

4.3.1. Position Control

The mobile robot can use this control approach to follow the target points (xd (t), yd (t)) along the target trajectory. The robots must go from their starting place to their final destination without changing their orientation. The controller receives two inputs, which are the disparities between the intended and target positions of the robot.

The position control block diagram is as follow:

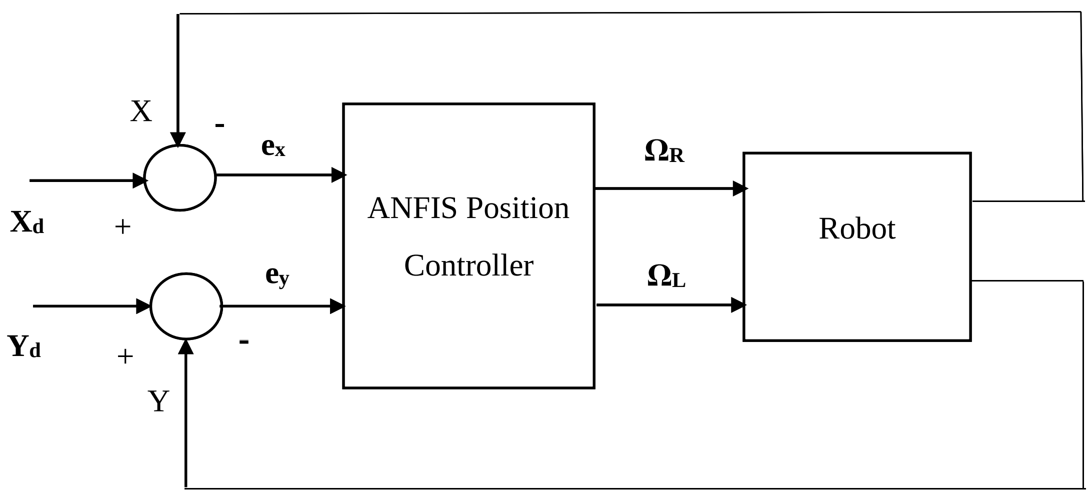


Figure 4.6 Position control block diagram

In the position control, the controller takes two inputs variables ex and ey given by equations (4.27) and (4.28) respectively. The position control diagram is shown in Figure 4.6 and the single output is expressed in equation (4.29).

$$e_x(t) = x_d(t) - x(t) \tag{4.27}$$

$$e_y(t) = y_d(t) - y(t) \tag{4.28}$$

Where,

x(t) and y(t) are the coordinates of the robot;

xd(t) and yd(t) are the target coordinates of the robot.

The neuro fuzzy have only one output $f_p(t)$:

$$f_p(t) = \frac{\sum_{i=1}^{m} w_i^p f_i^p}{\sum_{i=1}^{m} w_i^p} \tag{4.29}$$

4.3.2. Orientation Control

This control approach, like position control, allows the robot to spin around itself by following the target angle. The difference between the desired angle θd of the robot and the direction θ is the ANFIS's single input eθ. The angular velocity is the single output of fuzzy inference, neural networks, and defuzzification. The input is shown in equation (4.30) and the output is shown in equation (4.31).

The orientation control block diagram is described as follow:

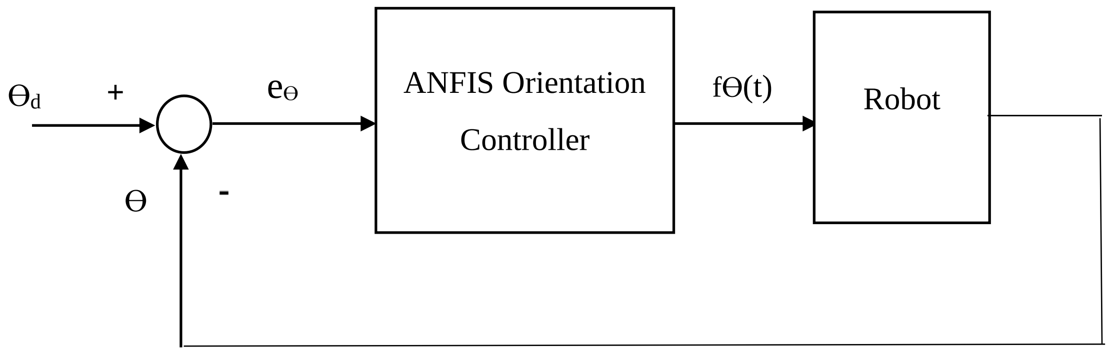


Figure 4.7 Orientation control block diagram

$$e_\theta(t) = \theta_d(t) - \theta(t) \tag{4.30}$$

$$f_o(t) = \frac{\sum_{i=1}^{m} w_i^\theta f_i^\theta}{\sum_{i=1}^{m} w_i^\theta} \tag{4.31}$$

4.3.3. Trajectory Control Strategy

The combination of position and orientation controls creates the trajectory control that allows the robot to follow the specified path. The controller's inputs ex, ey, and e_θ determine the difference between the robot's actual and desired positions.

The linguistic variable "distance" is obtained by using equation (4.33).

$$distance = \sqrt{e_x^2 + e_y^2} \tag{4.33}$$

The linguistic variable "alpha" which is the angle of orientation that the MR forms between the straight line of actual and target position are calculated by using equation (4.34).

$$alpha = \tan^{-1}\left(\frac{e_y}{e_x}\right) - \theta \tag{4.34}$$

v and w are from ANFIS position control, and w is from ANFIS orientation control. In this case, the angular velocity of the right and left wheels (ΩR and ΩL) is obtained by equations (4.35) and (4.36).

$$\Omega_R = \frac{1}{r}(v + Lw) \tag{4.35}$$

$$\Omega_L = \frac{1}{r}(v - Lw) \tag{4.36}$$

The mobile robot's orientation from the real position and orientation to the desired position, regardless of ultimate orientation, is the method implemented using ANFIS1. The mobile robot moves to the target point and rotates the (alpha) angle in this phase. Using ANFIS2, the mobile robot is oriented according to the goal pose once it has arrived at the target point.

The following diagram shows the global trajectory control:

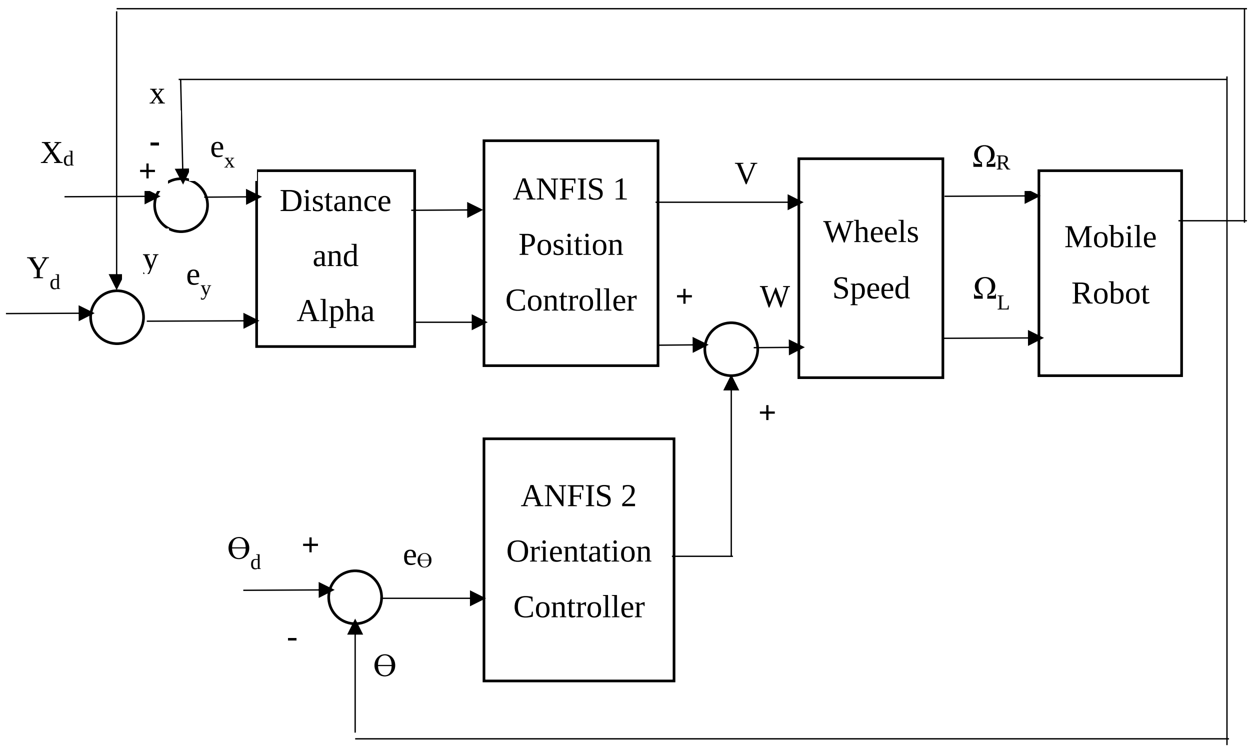


Figure 4.8 Trajectory control block diagram

4.4. Implementation of the Path Tracking Controller Using ANFIS

4.4.1. Adaptive Neuro-Fuzzy Inference System (ANFIS)

In recent years, the notion of fuzzy logic and artificial neural networks for control problems has become one of the most popular analysis subjects. The reason for this is because in traditional control theory, the controller is usually created using a mathematical model. The controller's effectiveness is generally degraded by the inaccuracy of mathematical modeling of the plants, particularly for nonlinear and complicated control difficulties. The introduction of fuzzy logic controllers (FLC) and neural controllers based on multilayered neural networks has sparked a new wave of possibilities for better, more efficient control.

It has been proposed and developed in recent years to combine fuzzy logic and neural networks, notably fuzzy neural networks (FNN). ANFIS is a word used to describe a system that combines fuzzy logic with neural networks (Adaptive Neuro Fuzzy Inference System). The neural system has a number of inputs and outputs. Fuzzy logic, on the other hand, has a single output and several inputs. ANFIS is a nonlinear application that uses the combination of these two [34].

Conventional PID controllers in various combinations are commonly used for industrial processes due to their simplicity and efficacy for linear systems, particularly initial and second-order systems. Proportional integral derivative (PID) controllers are well known for being useful for linear systems, but they are often ineffective for higher order and nonlinear systems. For non-linear systems, other controllers, such as PID, proved ineffective. This ANFIS controller, on the other hand, is more accurate than other controllers and is beneficial for nonlinear systems. This is the controller's most significant benefit.

The T-S fuzzy model was proposed by Takagi and Sugeno in 1985. It was dubbed the Sugeno fuzzy model by students. A nonlinear model is the Sugeno fuzzy model. It's a good way to express the dynamic properties of complicated systems. In addition, the fuzzy inference model is the most commonly used.

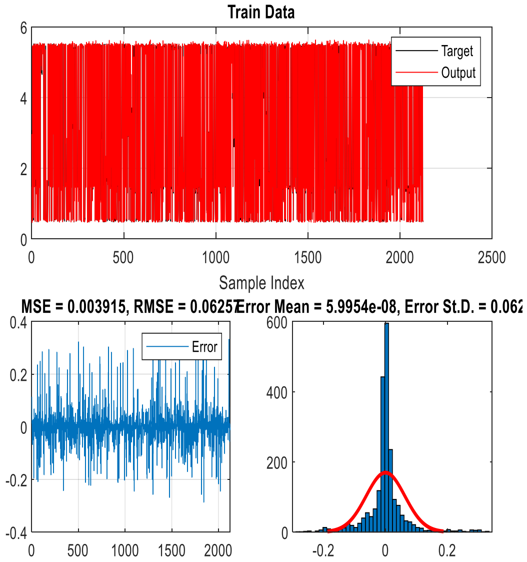
The ANFIS Controller is a neural network and fuzzy logic hybrid. The neural network receives a large number of inputs. The neural network produces a standard output based on the inputs. As a result, the neural network is trained based on the inputs and outputs. After the neural network has been trained, the output is sent into fuzzy logic, which provides IF THEN rules and membership functions. MATLAB is used for this.

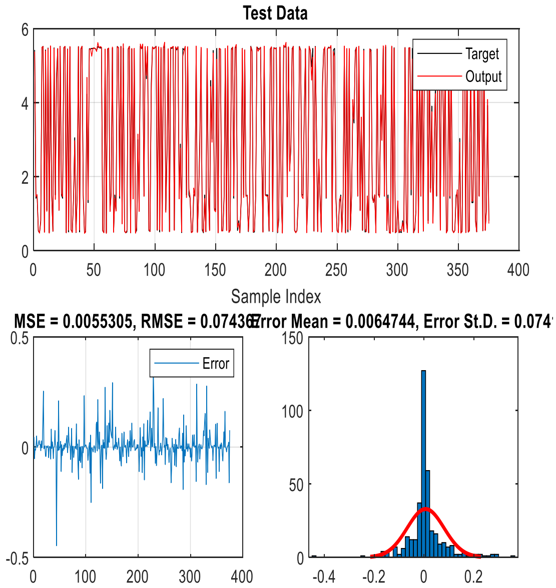
This ANFIS controller is commonly used for nonlinear system control. In comparison to traditional PID controllers and other controllers, this is the best controller. This controller is used to control the temperature of the water bath. In addition, this controller is utilized to operate planes. Intelligent planes that learn on their own and take off are currently being researched.

4.4.1.1 Trained Data Using ANFIS

Trained data is information that instructs the computer on how to process the data. It mimics the human brain's ability to take several inputs, interpret them, and trigger activations in brain cells. With neural network and machine learning programs, artificial neurons can duplicate a major portion of this process. These programs create extremely precise representations of the human thought process.

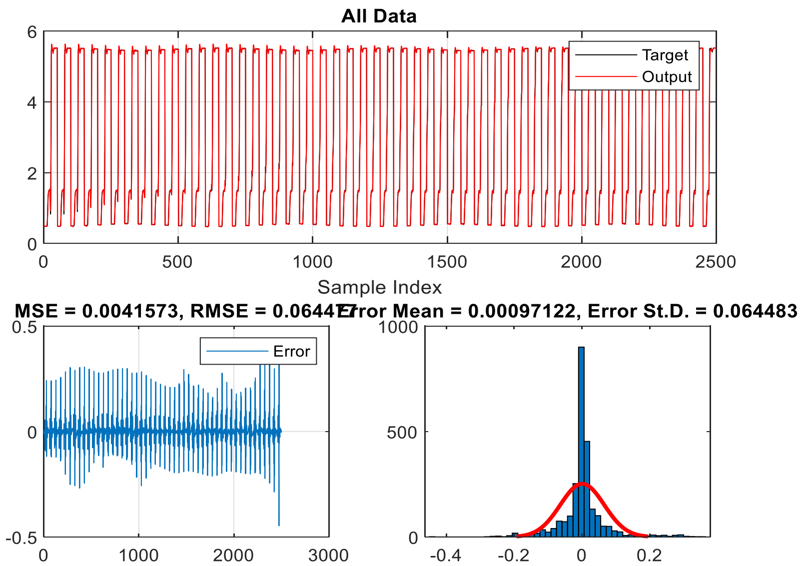
Training data is a form of data that is used to train a new application, model, or system using a variety of approaches, depending on the practicality and requirements of the project. Artificial intelligence or machine learning training data differs differently in that it is labeled or annotated using specific procedures to make it recognizable to computers, which aids machines in understanding the objects. The amount of data needed to train the software is determined by the problem's complexity level and an algorithm. Inadequate data might influence the model's forecast accuracy, and managing large amounts of data can be problematic, even if it can produce appropriate results. As a result, the amount of data required at first is determined by the problem's complexity.





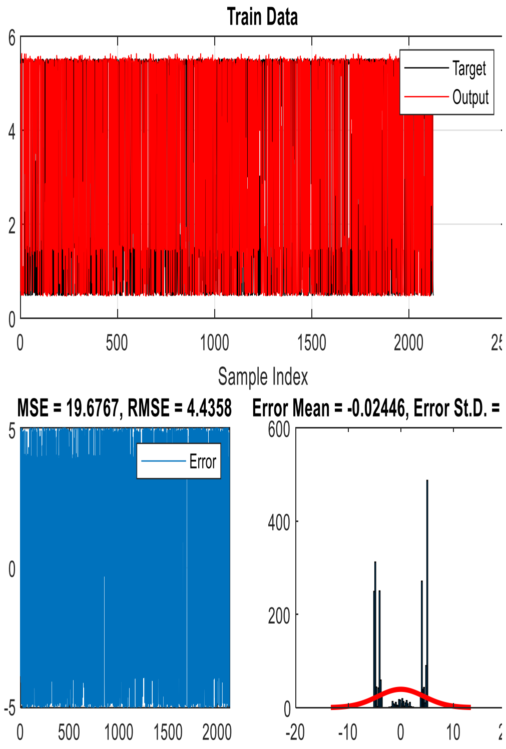
a) Train Data

b) Test Data

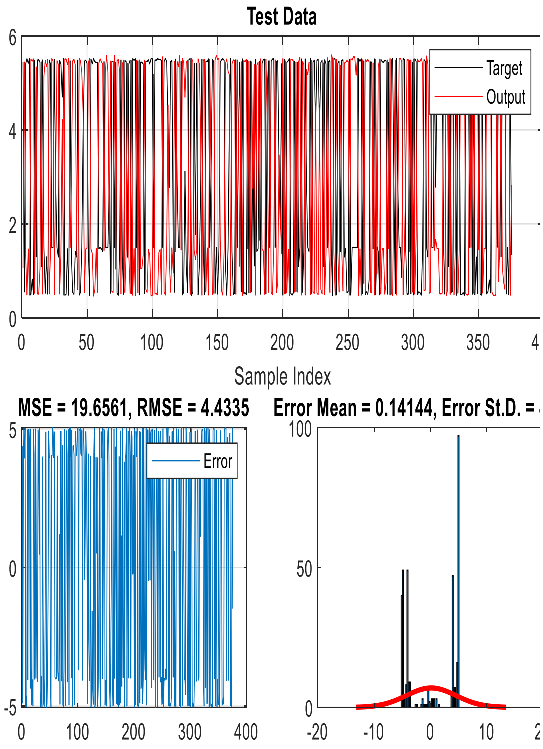


c) All Datas

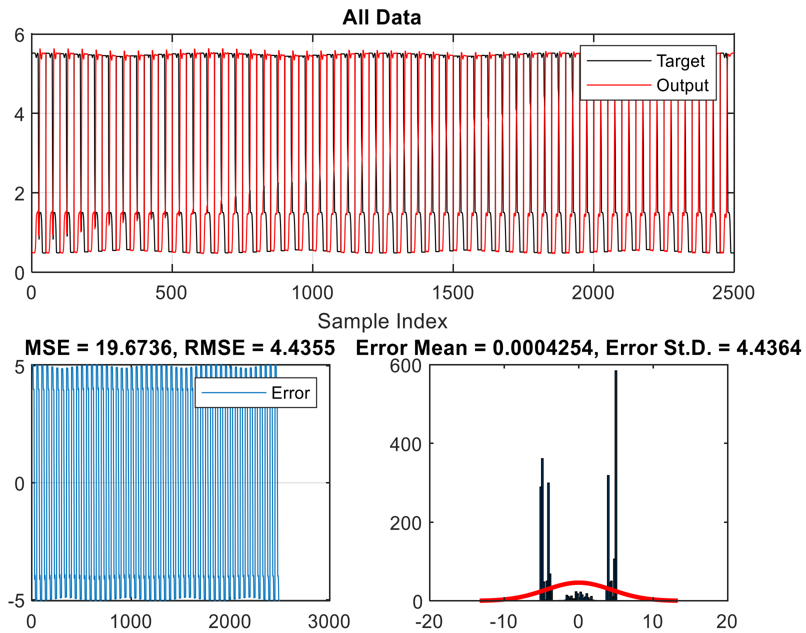
Figure 4.9 Train and Test Data for Position Control



a) Train Data



b) Test Data



C) All Datas

Figure 4.10 Train and Test Data for Orientation Control

4.4.2. Path Tracking Algorithm

The author describes a path tracking control system for autonomous wheeled mobile robots. In traditional systems, the path is usually designed using smooth curves with curvature-continuity, and a path tracking controller is constructed separately to adjust for path error that arises during navigation. Path tracking aims to reduce the distance and rotational error between a robot and a geometric reference path. An autonomous robot's control system consists of multiple interconnected functions, such as vision, path planning, and path tracking.

The ability of a mobile robot to track a path is determined by the way's properties, therefore the path generator must satisfy basic requirements such as position, direction, and curvature. However, because of the relationship between the characteristics of the required steering action and the properties of the vehicle, smooth variation in curvature is still required. The goal is to generate control commands for the robot so that it can follow a path despite environmental perturbations when it is tracking a previously determined path. As a result, it must take into account the robot's current position as well as the limits imposed by its lower motion controllers.

The method simplifies the path by calculating the path's main corners and establishing a new trajectory as a list of evenly spaced points. The path tracking method uses the pure pursuit algorithm to calculate steering commands in order to follow the path. The objective point is set a predetermined distance ahead of the robot in the intended path.

4.5. Development of a Strategy for Obstacle Avoidance

The fuzzy controller is mainly for controlling the robot's movement in this thesis. The robot's distance from the objective and the angle between the robot and the obstacles are also inputs. Left motor velocity and right motor velocity are the two outputs. As a result, the FLC has two inputs and two outputs.

4.5.1. Constructing Membership Functions

There are five membership functions which are triangular members. Zero (Z), Near (N), Medium (M), Far (F), Very Far (VF), and Big Negative (BN) are the languages for distance, while Negative (N), Zero (Z), Positive (P), and Positive Big (PB) are the languages for angle. Very Slow (VS), Slow (S), Medium (M), Fast (F), Very Fast (VF) linguistic variables are specified for L, R. In Table 4.2, the parameters that define the functions are listed.

Table 4.2 Input and Output Parameters

r (m)	Z [0 0 2.5]	N [0 2.5 5]	M [2.5 5 7.5]	F [5 7.5 10]	VF [7.5 10 10]
Ɵ (rad)	BN [-3.14 -3.14 -1.75 -0.75]	N [-1.5 -0.5 0]	Z [-0.3 0 0.3]	P [0 0.5 1.5]	BP [0.75 1.75 3.14 3.14]
ω_R	VS [0 0 1.5]	S [0 1.5 3]	M [1.5 3 4.5]	F [3 4.5 6]	VF [4.5 6 6]
ω_L	VS [0 0 1.5]	S [0 1.5 3]	M [1.5 3 4.5]	F [3 4.5 6]	VF [4.5 6 6]

4.5.2. Constructing the Rule Base

It is important to create a fuzzy logic control for the real velocities of the mobile robot that uses linguistic variables in the inputs and outputs in order to meet the control target. The expert system was built using the IF-THEN rules, which are a set of 25 rules for controlling robots on a predetermined path. This rule will help the robot change directions while striving to keep track of the target position.

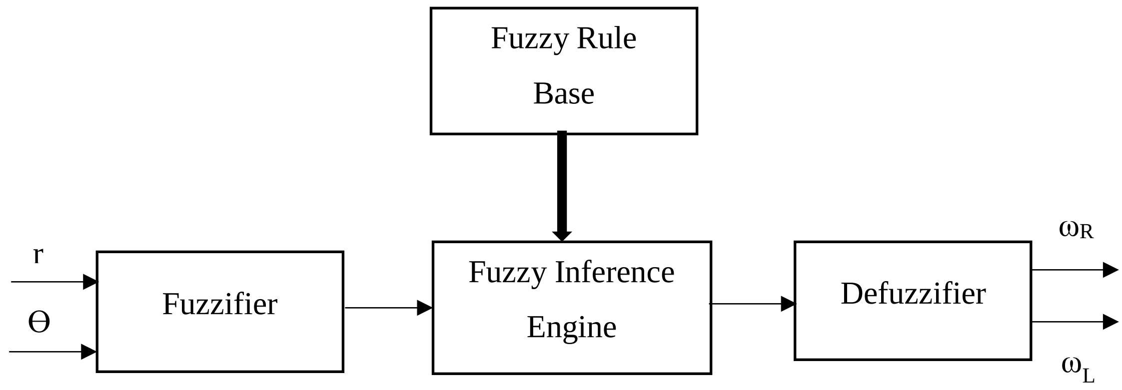


Figure 4.11 Conventional fuzzy system

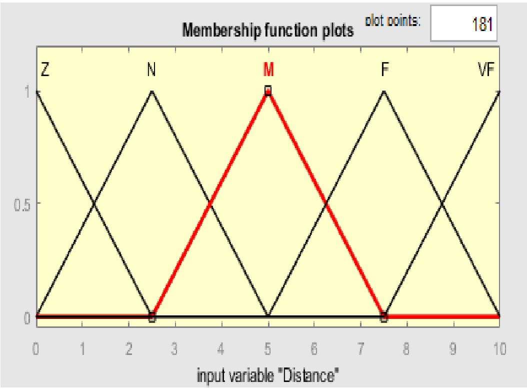
Table 4.3 Right Wheel Velocity

			X2=ϴ		
X	**BN**	**N**	**Z**	**P**	**BP**
Z	VS	S	VS	VF	VF
N	VS	VS	S	VF	VF
M	VS	S	M	VF	VF
F	VS	S	F	VF	VF
VF	VS	S	VF	VF	VF

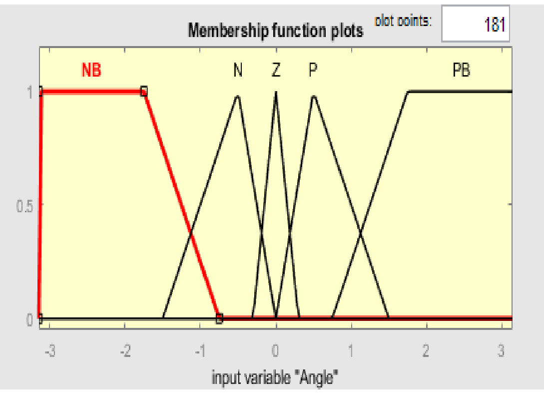
Table 4.4 Left Wheel Velocity

			X1=r		
X	**BN**	**N**	**Z**	**P**	**BP**
Z	VF	VF	VS	S	VS
N	VF	VF	S	S	VS
M	VF	VF	M	S	VS
F	VF	VF	F	S	VS
VF	VF	VF	VF	S	VS

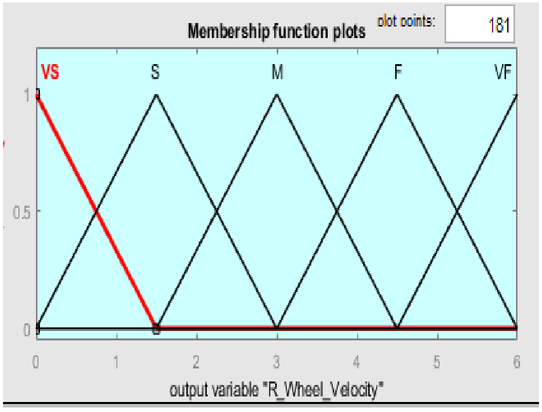
A membership function (MF) is a curve that describes how each point in the input space is transformed to a membership value (degree of membership) that ranges from 0 to 1. The universe of discourse is a term used to describe the input space. The only requirement for a membership function is that it must fluctuate between 0 and 1.



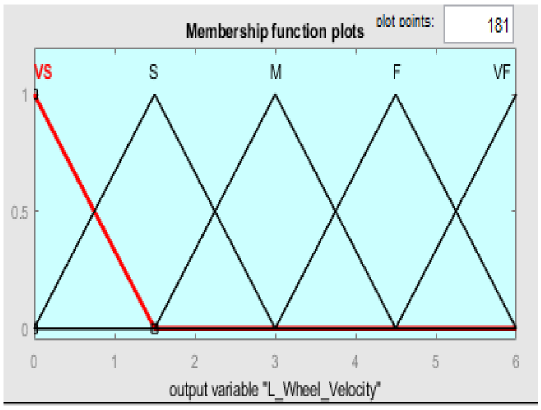
(a) Input Variable Distance



(b) Input Variable Angle



(c) Right Wheel Velocity



(d) Left Wheel Velocity

Figure 4.14 Membership function of fuzzy logic system

Depending on the number of linguistic terms employed, fuzzy membership functions can comprise several fuzzy sets. A linguistic term is represented by each fuzzy set. A degree of membership is a number that shows how much of a crisp value can be a member of each fuzzy collection. One crisp value can be translated into a number of fuzzy sets, but the degree of membership in each fuzzy set can vary.

CHAPTER FIVE

RESULTS AND DISCUSSIONS

The results for simulation of the problem stated in the previous parts are presented in this section. Simulation can be taken as a significant research tool since the early twentieth century, and it has now evolved into a strong visualization, planning, and strategy tool in a variety of sectors of research and development. In robotics, simulation is also quite significant. The controller was created using the R2019b version of the MATLAB/Simulink software.

Fuzzy logic, unlike classical logic, accepts error, uncertainty, and partial truth. The advantage of the fuzzy logic-based system in the control of mobile robots is that it can actualize the intuition of sensor-based navigation and readily translate linguistic information into control signals. The trajectory tracking of wheeled mobile robots is mostly explored using fuzzy logic control, which allows the mobile robot to move to a specified posture while following any reference trajectory.

5.1. Specification of a Mobile Robot

One of the most distinguishing characteristics of robots is their ability to interact with the outside world, altering it through their activities and responding to external events. Robots assist humans by doing important activities, extending their skills, and reducing our hazards when working in hazardous areas. To develop the correct robot for each application, robotics experts must be knowledgeable with logic, microprocessors, and computer programming. They must also develop specifications for the robot's capabilities in relation to the workplace. Robotics engineers are also in charge of cost-effective design, cost-price computations, and quality assurance.

Mobile robotics is a fast-evolving, solution-oriented business in which robotics engineers play an increasingly important role. Mobile robotics is a significant aspect of the business, with applications in manufacturing, agriculture, aerospace, mining, and medical, among others. A robotics engineer works in offices, factories, or laboratories, designing, maintaining, developing new applications, and conducting research to help robots reach their full potential. The job starts with a laser-like focus on a single business challenge in a certain industry. In manufacturing, for example, there may be a desire to enhance capacity by developing robots for activities that can be automated.

Table 5.1 Parameters of the WMR system [35]

Parameter		Value	Unit
r	Wheel Radius	0.04	M
m_g	Mass of the Rotation system	0.18	Kg
J_R	Moment of inertia of the rotation system	0.000025	Kgm^2
F_R	Friction Coefficient of the rotation system	0.2	$Nrad^{-1}s^{-1}$
l	Length measured from the wheel axis and mass center of the main body	0.08	m
m_B	Mass of the main body	1.16	Kg
J_B	Moment of inertia of the main body	0.0015	Kgm^2
F_B	Friction Coefficient of the main body	0.002	$Nrad^{-1}s^{-1}$
g	Gravitational acceleration of the Earth	9.807	ms^{-2}

5.2. Scenario for Path Tracking

Experiments were carried out on concrete terrain in an outdoor environment in this study, with various scenarios:

Scenario 1 (Seesaw shape): In this case, the robot is programmed to follow a sinusoidal course. The program specifies the vertex coordinates.

Scenario 2 (Circular shape): In this scenario, the robot follows a 10cm-radius circular trajectory with initial positions of (5,5).

Scenario 3 (Rectangle Shape): The robot is meant to follow a rectangular trajectory in this scenario. The software defines all of the vertex points.

Scenario 4 (Twisted Shape): In this scenario, the robot follows a common-point route. The entire point.

5.3. Path Tracking Result

As seen in the Fig.5.1 below the robot trajectory follows the reference trajectory in sinusoidal shape form.

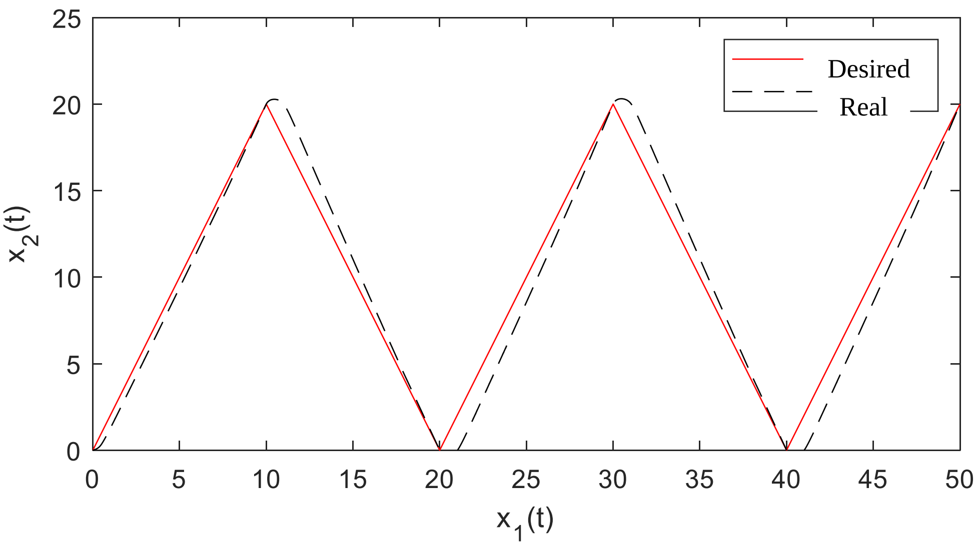


Figure 5.1 Tracking the trajectory of a mobile robot in the shape of a seesaw

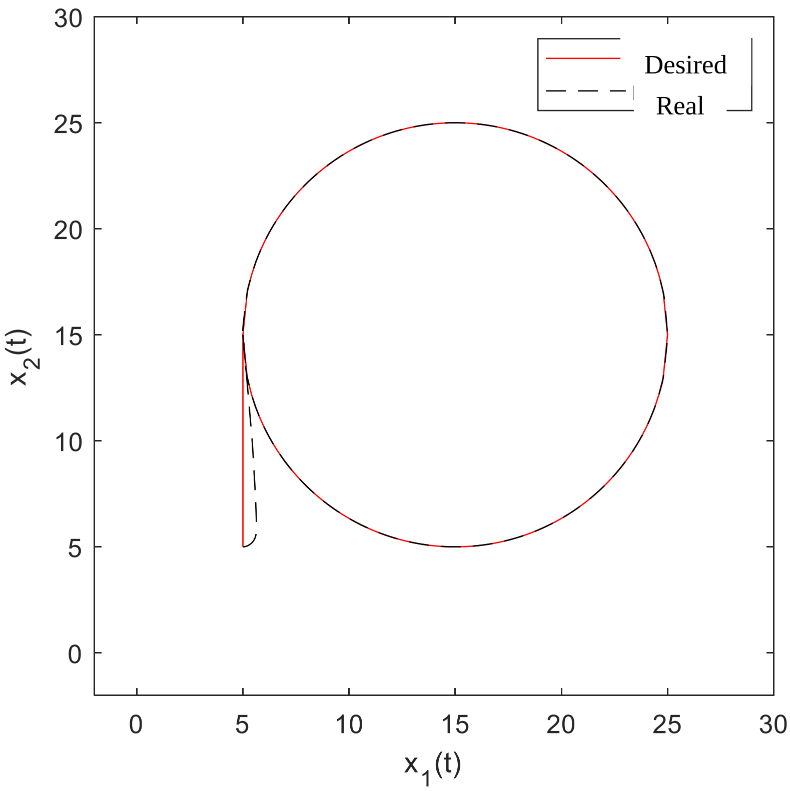


Figure 5.2 Tracking the circular path of a mobile robot.

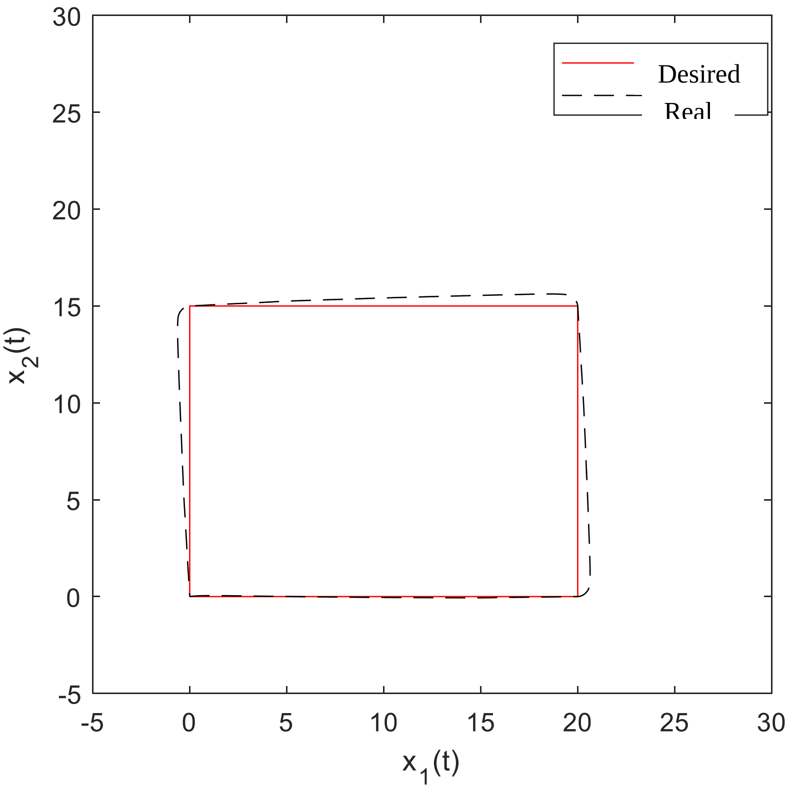


Figure 5.3 A mobile robot's rectangular trajectory tracking

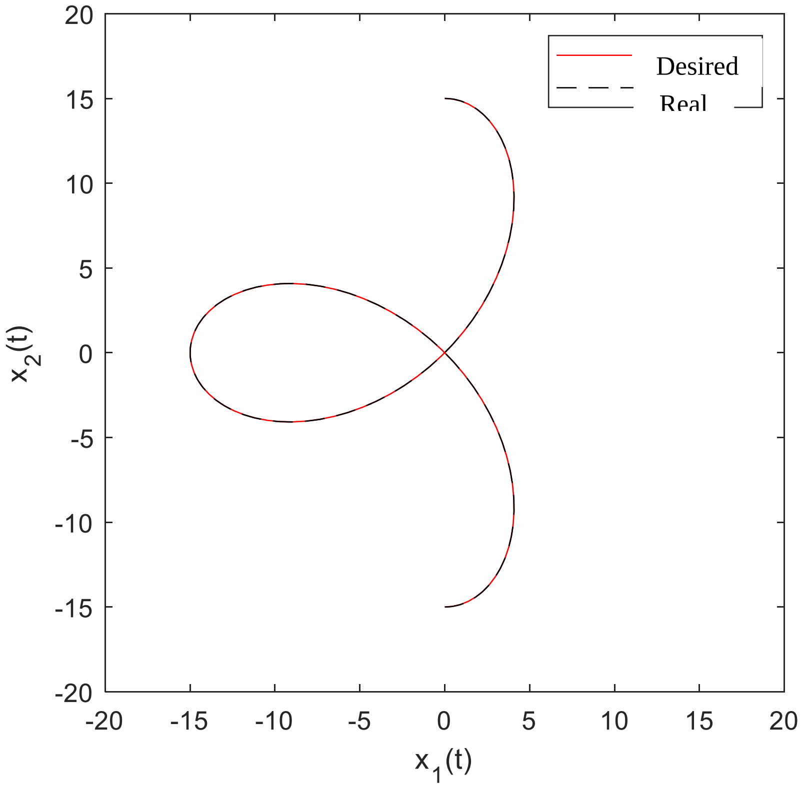


Figure 5.4 A mobile robot's tracking with overlapped points

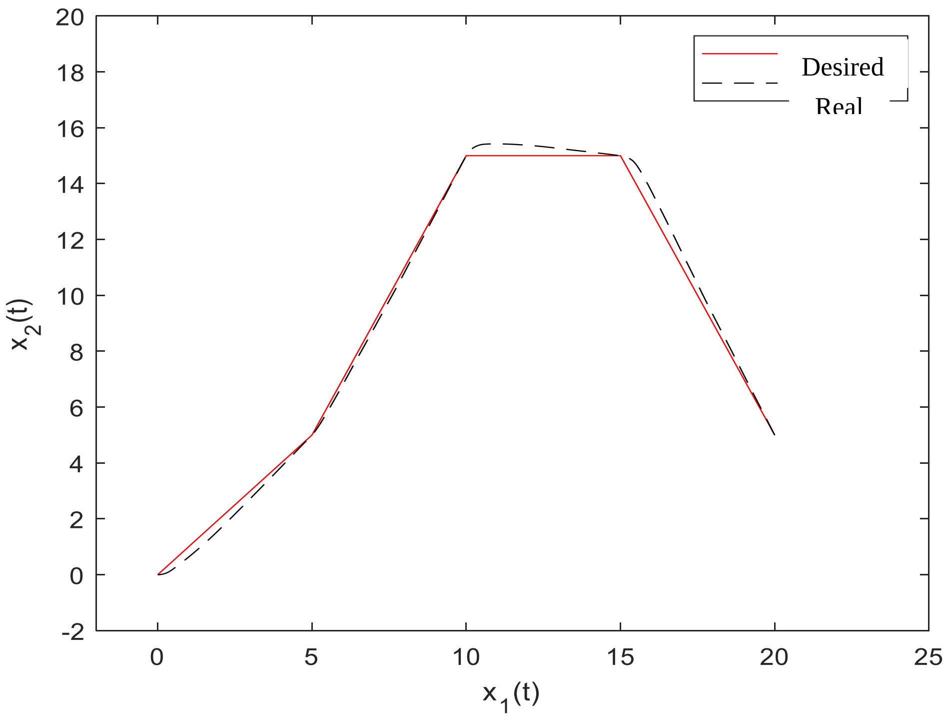


Figure 5.5 Tracking a mobile robot's different shapes and trajectory

This ANFIS controller is widely used for controlling the non-linear system. As this is the best controller as compared to conventional PID controller and other controller.

5.4. Obstacle Avoidance Result

In this case the mobile robot moves from its initial position (0,0) towards the goal without obstacle in its way.

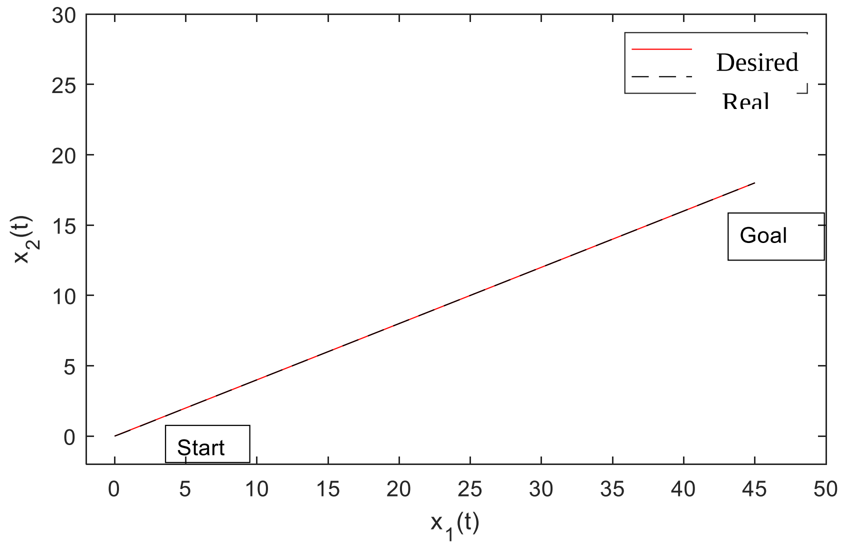


Figure 5.6 Getting to the goal without any difficulties

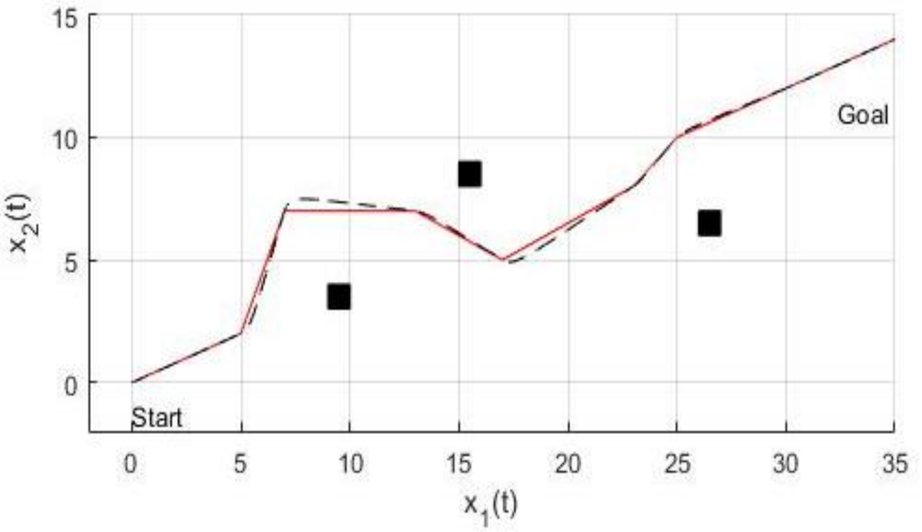


Figure 5.7 Getting to the goal with difficulties

As can be seen, the robot's motion in the workplace is different with and without obstacles. The scene in fig.5.7 has three new obstacles. However, when it approaches the destination, the robot adjusts course to avoid obstructions.

5.5. Simulation Results of DC motor

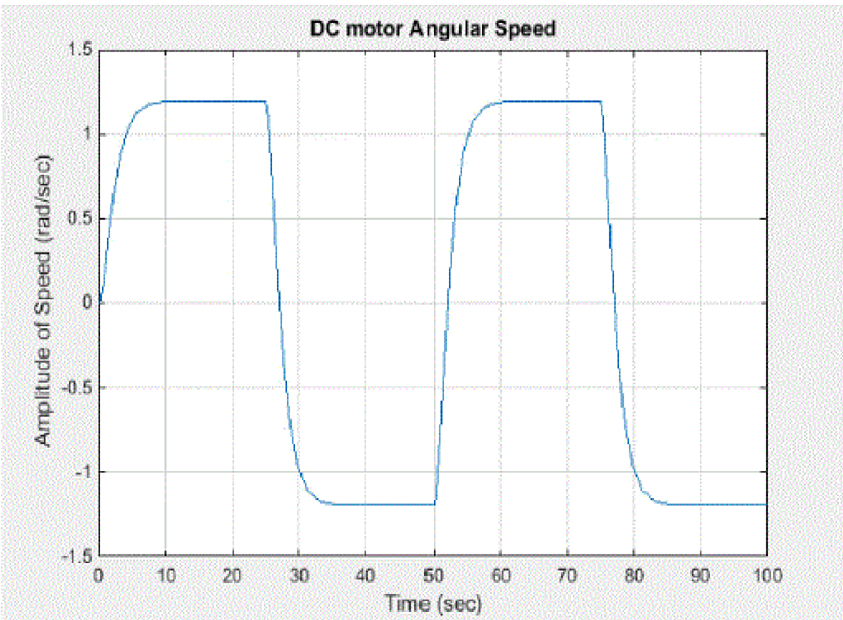


Figure 5.8 Speed of DC motor

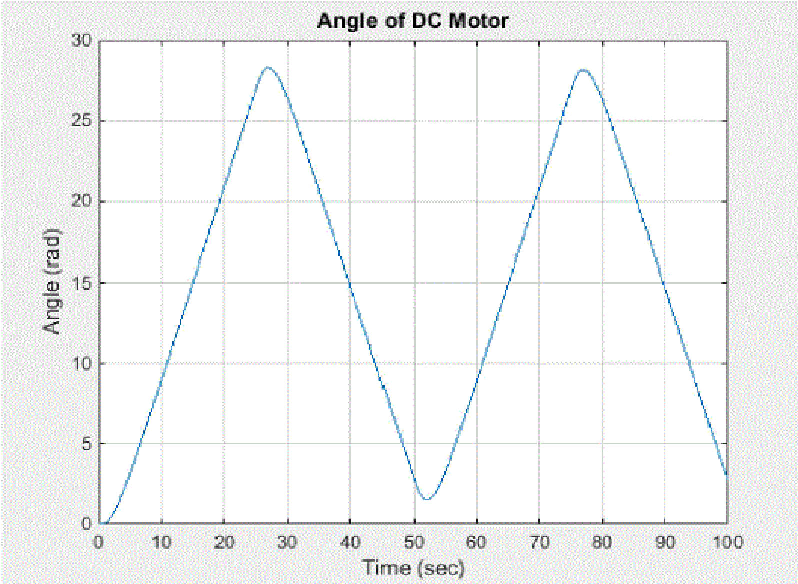


Figure 5.9 Angle of DC motor

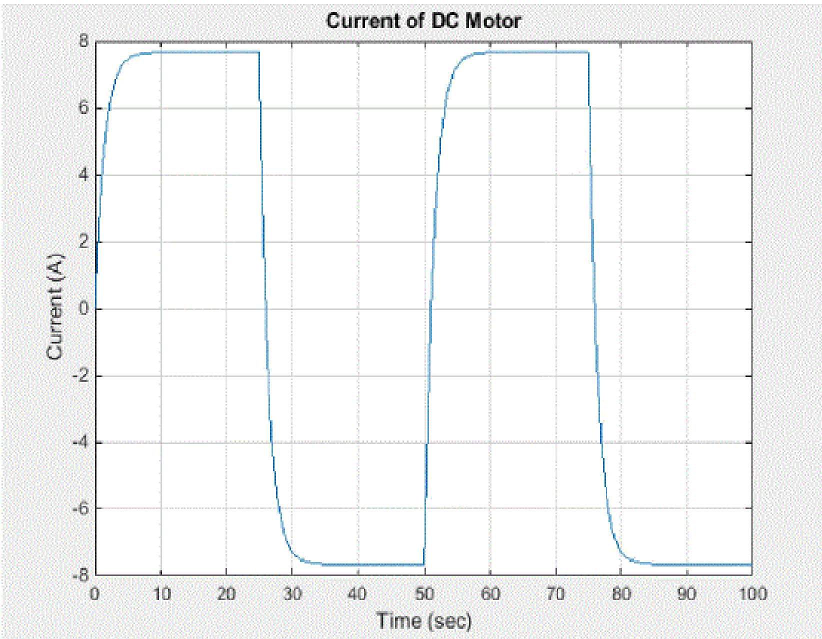


Figure 5.10 Current of DC motor

Why ANFIS is used?

The most significant advantages of ANFIS is it presents a much better learning ability for a similar network complexity, a much smaller convergence error is achieved, and although the convergence is slower the smallness of the error in ANFIS is able to compensate that fact. The ANFIS has also some advantages, including the ability to capture the nonlinear structure of a process, adaptation capability, and rapid learning capacity.

5.6 Comparative analysis of ANFIS controller and FIS controller

The fuzzy controller is a time consuming process to get an accurate membership function. To track the given trajectory efficiently the controller must be ANFIS. The Mean Square Error (MSE) of the position controller and orientation controller for repeated sequence reference is shown in Table 5.2 to compare the deviation of simulated actual path from desired path between the FIS and ANFIS.

Table 5.2 Comparison of MSE value

MSE	FIS	ANFIS
MSE in position x (m)	0.0013	0.0008
MSE in position y (m)	0.0012	0.0010
MSE in orientation theta (rad)	1.8556	0.0064

5.7. Discussion

According to related works for path tracking using fuzzy, unlike the ANFIS base controller fuzzy tracking has some loss precision with simple paths. These types of controllers are not able to follow the extremes curves along the simple path. In conclusion, ANFIS controller shows a better performance in following any kinds of paths by any degrees of complexity also have flexibility to different conditions like sudden turns.

The ANFIS combines the neural networks and fuzzy systems techniques. When you mix the two clever ways, you'll get good quality and quantity reasoning. To put it another way, fuzzy inference and network computing will be available concurrently. The path tracking "Adaptive Neuro Fuzzy Inference System" (ANFIS) controller's job is to lead the robot to follow the route in a smooth, continuous, and precise manner.

Obstacle avoidance robots are created to help robots navigate through unfamiliar environments by avoiding collisions. The robot can avoid obstacles and increase its target-finding skills with the help of an intelligent controller for mobile robots. The robot's angular and linear velocity change as it moves towards the goal (using the obstacle avoidance approach).

CONCLUSION AND RECOMMENDATION

6.1. Conclusion

The related works are reviewed in this study, and data on mobile robots is acquired from the literature. In addition, both the proposed plant and controller have been mathematically modelled.

An adaptive neuro-fuzzy inference system (ANFIS) may be used to learn FIS and identify and refine various parameters in membership functions and fuzzy rules using values from the training data set. The adaptive tracking controller of today learns and develops the information required for the mobile robot's tracking trajectory. Different simulation results should be used to demonstrate the trajectory controller's capabilities. MR can track specific specified reference paths thanks to this neuro-fuzzy controller. The simulation results suggest that the proposed strategy is adequate for solving the problem of target monitoring.

The fuzzy logic control system is utilized to analyze the mobile robot's obstacle avoidance system, and the design process of the mobile robot's obstacle avoidance algorithm based on fuzzy logic is also discussed. The speed of the left and right wheels is its output to the fuzzy control system, and the angle between the obstacle's position and the robot's target position is its input variable.

Proposed ANFIS controller is mainly focused on future Path curvature to produce linear and angular velocities signal for smooth motion of robot discovers that the designed ANFIS base controller has more abilities and better performance in tacking paths with any kinds of complexities, such as sudden turns, successive curves, and slopes. In conclude that designed ANFIS base control system shows great precision and tracking ability. In addition to online network, it has noticeable resistance to noise because of its intelligent algorithm.

6.2. Recommendation

By including trajectory tracking control into the trajectory planning algorithm, future work on tracking the tracking of mobile robots will be possible. When we deploy several mobile robots instead of a single mobile robot, future study on this mobile robot platform will be more intriguing. In addition, developing more powerful hybrid technologies to accomplish better trajectory control will be advocated in the future.

References

[1] Weidong Wang, Wenrui Gao, Siyu Zhao, Wenwu Cao, and Zhijiang DU,"Robot protection in the Hazardous Environments", December 2017.

[2] E. Wijn, “Unstable behavior of a unicycle mobile robot with tracking control,” Report No. DCT 2004.63, Eindhoven, Eindhoven University of Technology, 2004.

[3] R. Barzamini and A. Afshar, “Dynamic adaptive tracking control for wheeled mobile robots,” AmirKabir University of Technology, Tehran, 2006.

[4] https://www.cs.bham.ac.uk/internal/courses/int-robot/halloffame/2013/priya/

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

[6] 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.

[7] S. Liu, D. Sun, and C. Zhu, “A dynamic priority-based path planning for the cooperation of multiple mobile robots information forming,” Robotics and Computer-Integrated Manufacturing, vol. 30, no. 6, pp. 589–596, 2014.

[8] A. Babinec, F. Duchon, M. Dekan, P. P ˇ aszt ´ o, and M. Kele- ´ men, “VFH* TDT (VFH* with Time-Dependent Tree): a new laser rangefinder based obstacle avoidance method designed for an environment with non-static obstacles,” Robotics and Autonomous Systems, vol. 62, no. 8, pp. 1098–1115, 2014.

[9] A. Sgorbissa and R. Zaccaria, “Integrated obstacle avoidance and path following through a feedback control law,” Journal of Intelligent & Robotic Systems: Theory and Applications, vol. 72, no. 3-4, pp. 409–428, 2013.

[10] A. Sgorbissa and R. Zaccaria, “Planning and obstacle avoidance in mobile robotics,” Robotics and Autonomous Systems, vol. 60, no. 4, pp. 628–638, 2012.

[11] Anthony Stentz. Optimal and efficient path planning for unknown and dynamic environments. International Journal of Robotics and Automation, 10:89–100, 1993.

[12] M. Djakulovic and I. Petrovi ´ c. Dynamic window ´ based approach to mobile robot motion control in the presence of moving obstacles. In Proceedings of IEEE International Conference on Robotics and Automation - ICRA 2007, Roma, Italy, 10-14 April 2007, pages 1986–1991, 2007.

[13] G. Oriolo, A. D. Luca, and M. Vendittelli. Wmr control via dynamic feedback linearization: Design, implementation, and experimental validation. IEEE Transactions on Control Systems Technology, 10(6):835–852, 2002.

[14] A. K. GUPTA, S. ARORA and J. WESTCOTT, Industrial Automation And Robotics, 3rd, Ed., Dulles, Virginia: Mercury Learning And Information, 2017.

[15] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics,” Texas: Automation and Robotics Institute, The University of Texas at Arlington, 1996.

[16] J. M. Yang and J. H. Kim, “Sliding mode control for trajectory tracking of non-holonomic wheeled mobile robots,” IEEE Transactions on Robotics and Automation, 578-587, 1999

[17] A. Bloch and S. Drakunov, “Stabilization of a nonholonomic system via sliding modes,” IEEE international conference on Decision Control, pp. 2961-2963, 1994.

[18] J. Ye, “Tracking control for nonholonomic mobile robots: Integrating the analog neural network into backstepping technique,” Neurocomputing, 3373-3378. (2007).

[19] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot using neural networks,” IEEE transactions of neural networks, 589-600, 1998.

[20] J. Velagic, N. Osmic and B. Lacevic, “Neural Network controller for mobile robot motion control,” International Journal of intelligent systems and technologies, 2008.

[21] Mostafa A. Hamood, Waleed F. Faris Marwan A. Badran, "Fuzzy Logic Based Speed Control System for Three Phase Induction Motor," EFTIMIE MURGU, pp. ISSN 1453 - 7397, January 2013.

[22] M. Pascal and S. Claude. “Motion Control of Wheeled Mobile Robots” Handbook of Robotics, S. Bruno, K.S Oussama (Eds), Springer-Verlag Berlin Heidelberg, 2008, pp.799-825.

[23] Reprinted, with permission, from Proceedings of the 1994 IEEE Robotics and Automation Conference, San Diego, California, May 8-13, 1994, pp. 2023-2029.

[24] Ali, Razan Mohammed Abbas Mohammed. *Car crash avoidance system using Ultrasonic Sensor*. Diss. Sudan University of Science and Technology, 2017.

[25] Medina-Santiago, A., et al. "Neural control system in obstacle avoidance in mobile robots using ultrasonic sensors." *Journal of applied research and technology* 12.1 (2014): 104-110.

[26] Kumano, Masako, Akihisa Ohya, and S. Yuta. "Obstacle avoidance of autonomous mobile robot using stereo vision sensor." *Intl. Symp. Robot. Automat*. 2000.

[27] Stachniss, Cyrill, and Wolfram Burgard. "Exploring unknown environments with mobile robots using coverage maps." *IJCAI*. Vol. 2003. 2003.

[28] Vamsi Mohan Peri; Fuzzy Logic Controller for an Autonomous Mobile Robot, Electrical and Electronics Engineering, Jawaharlal Nehru Technological University, India, May, 2002.

[29] Campion Guy, Bastin Georges, and Brigitte D‘ AndrCa-Novel, ―Structural properties and classification of kinematic and dynamic models of wheeled mobile robots, IEEE Transactions on Robotics and Automation 12 (1) (1996) 47–62.

[30] Mahfouz, A. A., Ayman, A. A., and Salem, F. A. Mechatronics design of a mobile robot system. Int. J. Intell. Syst. Appl., 2013, 5, 23–36.

[31] Salem, F. A. and Mahfouz, A. A. Modeling and controller design for PMDC motor, using different control strategies and verification using MATLAB/Simulink (II). Forthcoming.

[32] Dorf, R. C., and Bishop, R. H. Modern Control Systems, 9th ed. Prentice-Hall, New Jersey, 2001.

[33] Richard C. Dorf and Robert H. Bishop. Modern Control Systems. Ninth Edition, Prentice-Hall Inc., New Jersey, 2001.

[34] J. –S. Roger Jang. Anfis: adaptive-network-based fuzzy inference systems. IEEE Trans. On systems, Man and Cybernetics, 1992.

[35] R. Silva-Ortigoza, C. Márquez-Sánchez, M. Marcelino-Aranda, M. Marciano-Melchor,G. Silva-Ortigoza, R. Bautista-Quintero, E. R. Ramos-Silvestre, J. C. Rivera-Díaz, and D.Muñoz-Carrillo, “Construction of a WMR for trajectory tracking control: Experimental
results,” The Scientific World Journal, vol. 2013, Article ID 723645, pp. 1–17, 2013.

APPENDIX

APPENDIX A: MATLAB CODES

MATLAB code for Circular Trajectory

```
%% Tracking trajectory for test signal 2

function test2

clf
global Ra L

load('U1_fis')

load('U2_fis')

Ra= 0.2; L= 0.4;

t= 0; x= [5; 5; 0*pi/180]; h= 0.1; wt= 2*L; ht= 4*Ra;

buf= [t x'];

xx= linspace(5,25,100);

yy= 15+sqrt(100-(xx-15).^2);

trline= [x(1) x(2);[xx' yy']];

xx= linspace(25,5,100);

yy= 15-sqrt(100-(xx-15).^2);

trline= [trline;[xx' yy']];

hold on

plot(trline(:,1),trline(:,2),'r'); % Draw the target line

[xx,yy]= DrawCar([x(1) x(2) wt ht],x(3));

gh2= plot(x(1),x(2),'k--'); % Draw the trajectory

axis equal

axis([-2 30 -2 30])
```

```
xlabel('x_1(t)'), ylabel('x_2(t)')

grid on

hold off

for i= 2:size(trline,1)

    xd= trline(i,:);

    d= 50;

    while d > 0.03

        d= sqrt((xd(1)-x(1))^2+(xd(2)-x(2))^2);

        E1= Saturation(d,[0 10]);

        beta= atan2(xd(2)-x(2),xd(1)-x(1));

        gam1= beta-x(3);

        if gam1 > pi

            gam= gam1-2*pi;

        elseif gam1 < -pi

            gam= gam1+2*pi;

        else

            gam= gam1;

        end

        E2= Saturation(gam,[-3 3]);

        u(1)= evalfis(U1_fis,[E1 E2]);

        u(2)= evalfis(U2_fis,[E1 E2]);

        x= RK4(@ModelCar,t,x,u,h);

        t= t+h;

        buf= [buf;t x'];
```

```
        [xx,yy]= DrawCar([x(1) x(2) wt ht],x(3));
        set(gh2,'xdata',buf(:,2),'ydata',buf(:,3)); % Draw the trajectory
        drawnow
    end
end
```

MATLAB Codes for Twisted Shape

```
%% Tracking trajectory for test signal 3
function test3
clf
global Ra L
load('U1_fis')
load('U2_fis')
Ra= 0.2; L= 0.4;
t= 0; x= [0; 15; 0*pi/180]; h= 0.1; wt= 2*L; ht= 4*Ra;
buf= [t x'];
th= linspace(0,pi,100); r= 15*cos(2*th);
xx= r.*sin(th); yy= r.*cos(th);
trline= [xx' yy'];
hold on
plot(trline(:,1),trline(:,2),'r'); % Draw the target line
[xx,yy]= DrawCar([x(1) x(2) wt ht],x(3));
gh2= plot(x(1),x(2),'k--'); % Draw the trajectory
axis equal
axis([-20 20 -20 20])
xlabel('x_1(t)'), ylabel('x_2(t)')
```

```
grid on
hold off
for i= 1:size(trline,1)
    xd= trline(i,:);
    d= 50;
    while d > 0.03
        d= sqrt((xd(1)-x(1))^2+(xd(2)-x(2))^2);
        E1= Saturation(d,[0 10]);
        beta= atan2(xd(2)-x(2),xd(1)-x(1));
        gam1= beta-x(3);
        if gam1 > pi
            gam= gam1-2*pi;
        elseif gam1 < -pi
            gam= gam1+2*pi;
        else
            gam= gam1;
        end
        E2= Saturation(gam,[-3 3]);
        t= t+h;
        [xx,yy]= DrawCar([x(1) x(2) wt ht],x(3));
        set(gh2,'xdata',buf(:,2),'ydata',buf(:,3)); % Draw the trajectory
        drawnow
    end
end
```