Adaptive Fuzzy Sliding Mode Controller Trajectory Tracking for Three Wheeled Mobile Robot

Eyasu Desta Abasber

A Thesis Submitted to

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

Sept 2024

Adama, Ethiopia

Adaptive Fuzzy Sliding Mode Controller Trajectory Tracking for Three Wheeled Mobile Robot

Eyasu Desta Abasber

Advisor: Dr. Tefera Terefe

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

Sept 2024

Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled "Adaptive Fuzzy Sliding Mode Controller Trajectory Tracking for Three Wheeled Mobile Robot" is my original work. That is, it has not been submitted 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 duly acknowledged through citation.

__Eyasu Desta Abasber__ ________________________ _______________

Name of student Signature Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "Adaptive Fuzzy Sliding Mode Controller Trajectory Tracking for Three Wheeled Mobile Robot" prepared under my guidance by Eyasu Desta Abasber, submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Control Engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Dr. Tefera Terefe	________________	______________.
Advisor	Signature	Date

APPROVAL SHEET

I, the advisors of the thesis entitled "Adaptive Fuzzy Sliding Mode Controller Trajectory Tracking for Three Wheeled Mobile Robot" and developed by Eyasu Desta Abasber, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Tefera Terefe ____________________ ______________.

Advisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by Eyasu Desta Abasber, have read and evaluated the thesis entitled "Adaptive Fuzzy Sliding Mode Controller Trajectory Tracking for Three Wheeled Mobile Robot" and examined the candidate during the 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 Control Engineering.

__________________________ ____________________ ________________

Chairperson Signature Date

__________________________ ____________________ ________________

Internal Reviewer Signature Date

Dr. Riessom Weldegiorgis ____________________ 03/10/2024

External Reviewer Signature Date

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

________________________ ___________________ _______________

Department Head Signature Date

________________________ _________________ _______________

School Dean Signature Date

________________________ _________________ _______________

Office of Postgraduate Studies, Dean Signature Date

ACKNOWLEDGEMENT

I am deeply thankful to God for providing me with the courage and determination to finish my thesis. I am extremely appreciative of my advisor, Dr. Tefera Terefe, for his continuous support, helpful guidance, and constant encouragement. His knowledge and feedback were very important to my research. I also want to show my family my appreciation, friends, and coworkers for having faith in me and supporting me. Their encouragement motivated me a lot. Lastly, I appreciate Dire Dawa University for allowing me to study for my master's degree at ASTU.

Table of Contents

DECLARATION i

RECOMMENDATION ii

APPROVAL SHEET iii

ACKNOWLEDGEMENT iv

LIST OF FIGURES ix

LIST OF TABLES xi

LIST OF ACRONYMS xii

ABSTRACT xiii

CHAPTER ONE 1

1. INTRODUCTION 1

1.1 Background of the Study 1

1.1.1 Types of Mobile Robot 2

1.1.2 Wheeled Mobile Robot 2

1.2 Statement of the Problem 5

1.3 Objective 5

1.3.1 General Objective 5

1.3.2 Specific Objective 5

1.4 Scope of Thesis 6

1.5 Significance of the Study 6

1.6 Limitations of the Study 6

CHAPTER TWO 7

2. LITERATURE REVIEW 7

2.1 Introduction 7

2.2 Related Work of Studies 7

CHAPTER THREE 10

3. TWMR SYSTEM MODELING AND METHODOLOGY 10

3.1. Overview 10

3.2. Methods or Procedures Followed.. 10

3.3 The Block Diagram of TWMR with the Proposed Control Method.................................. 11

3.4 Modeling and Description of the TWMR System 12

3.4.1 Representing Robot Coordinate.. 12

3.4.2 Kinematic Modeling of TMWR 13

3.4.2.1 The Forward Kinematics Model of Differential Drive WMR.................................. 14

3.4.2.2. Nonholonomic Kinematic Constraints 17

3.4.3. Dynamic Modeling of TWMR Using Lagrangian Approach..................................... 20

3.4.3.1. Dynamic Model Using Wheel Constraint 22

3.4.3.2. Dynamic Model Configuration and Simplification 24

3.4.4. Controllability Analysis of TWMR... 30

3.5. Open-Loop System Modeling Simulation Analysis 31

CHAPTER FOUR 4. CONTROLLER DESIGN.. 35

4.1. Reference Trajectory Generation 35

4.2 Adaptive Fuzzy Sliding Mode Control (AFSMC) Design.. 36

4.2.1. Introduction to Sliding Mode Control Approach 36

4.2.2 Kinematic-Based Sliding Mode Controller 37

4.2.2.1. Lyapunov Stability Analysis 39

4.2.3 Adaptive Fuzzy Sliding Mode Control Technique... 41

4.2.3.1 An Overview of Fuzzy Systems 41

4.2.3.2 Lyapunov Stability Analysis 43
4.2.3.3 Extraction of the Rule Base 46
4.3. Nonlinear PID Controller 49
4.3.1. Introduction to PID Controller 49
4.3.2. Nonlinear PID Controller for TWMR Velocity Control 50
4.4. Overall Proposed System Modeling 51
4.5. Controller Gains Optimization by Using a Genetic Algorithm 52
4.5.1. Introduction to Genetic Algorithm (GA) 52
4.5.2. Objective Function 53
4.5.3. Parameter Settings of the Genetic Algorithm 54
CHAPTER FIVE 56
5. RESULTS AND DISCUSSION 56
5.1. Introduction 56
5.2. Simulation Results 56
5.2.1. Trajectory Tracking of TMWR using AFSMC 57
5.2.1.1. Tracking Response of a Linear Reference Trajectory (Straight-line) 58
5.2.1.2. Tracking Response of a Nonlinear Reference Trajectory (Eight-Shape) 62
CHAPTER SIX 69
6. CONCLUSION AND RECOMMENDATION 69
6.1. Conclusion 69
6.2. Recommendation 69
REFERENCES 71
APPENDIXES 75
Appendix A: MATLAB Functions 75

Appendix B: MATLAB/Simulink System Block Diagrams 78

LIST OF FIGURES

Figure 1.1: (A) fixed standard wheel, (B) castor wheel……………………………………….. 3

Figure 1.2: Swedish wheels…………………………………………………………………. 3

Figure 3.1: Proposed workflow or procedures………………………………………………...10

Figure 3.2: Proposed Block diagram of TWMR………………………………………………11

Figure 3.3: Inertial and robot frame coordinate representation…………………………………. 12

Figure 3.4: 2D velocity coordinate configuration of TWMR…………………………………… 15

Figure 3.5: Velocity representation of DWMR………………………………………………… 23

Figure 3.6: Open-loop trajectory tracking, and robot pose for the same magnitude but a different input sign…………………………………………………………………………………32

Figure 3.7: Open-loop trajectory tracking, and robot pose for different magnitudes, but the same +ve sign input……………………………………………………………………………….33

Figure 3.8: Open-loop trajectory tracking, and robot pose for a different magnitude but the same -ve sign input………………………………………………………………………………. 34

Figure 3.9: Open-loop trajectory tracking, and robot pose for the same magnitude and the same +ve sign input………………………………………………………………………………. 34

Figure 4.1: State trajectory of a system under the influence of an SMC………………………...36

Figure 4.2: Block diagram representation of a fuzzy system…………………………………… 42

Figure 4.3: Triangular-shaped input membership functions……………………………………. 48

Figure 4.4: Basic PID controller structure with TWMR dynamic model………………………. 49

Figure 4.5: Nonlinear PID controller with TWMR dynamic model……………………………. 50

Figure 4.6: Overall system model with AFSM and NPID controller…………………………… 51

Figure 4.7: Pseudo-code of the genetic algorithm………………………………………………52

Figure 5.1: Matlab/Simulink system simulation block diagram…………………………………56

Figure 5.2: x, y, and θ variables tracking response and tracking error…………………………..58

Figure 5.3: Linear trajectory tracking and position error…………………………………….. 59

Figure 5.4: x, y, and θ variables tracking response and tracking error with the initial position of [2, 1, pi/2]……………………………………………………………………………….. 60

Figure 5.5: Linear trajectory tracking and tracking error with [2,1, pi/2] initial position……… 61

Figure 5.6: x-position tracking response and tracking error with [0, 0, 0] initial position………62

Figure 5.7: y-position tracking response and tracking error with [0, 0, 0] an initial position…...63

Figure 5.8: Angle tracking response and tracking error with the initial position [0, 0, 0]……… 63

Figure 5.9: Eight-shape trajectory tracking response with initial position [0, 0, 0]…………….. 64

Figure 5.10: Eight-shape trajectory tracking error with the initial position [0,0,0]…………….. 65

Figure 5.11: x position tracking and tracking error with initial position [1, 2, pi/2]…………….66

Figure 5.12: y position tracking and tracking error with initial position [1, 2, pi/2]…………….66

Figure 5.13: Angle tracking and tracking error with the initial position [1, 2, pi/2]……………. 67

Figure 5.14: Eight-shape trajectory tracking response with the initial position [1, 2, pi/2]…….. 67

Figure 5.15: Robot trajectory tracking error with [1, 2, pi/2] initial state………………………. 68

LIST OF TABLES

Table 3. 1: TMWR and motor parameters specifications (Martins & Bertol, 2022).................... 32

Table 4. 1: Parameters of the membership functions of $\sigma*$ 48

Table 4. 2: Optimization parameters for the genetic algorithm 55

Table 5. 1: Parameters Sliding Mode and NPID controller gains optimized by GA.................... 57

Table 5. 2: Adaptive Fuzzy Sliding Mode and NPID controller gains optimized by GA 57

Table 5. 3:RMS values of linear trajectory tracking error with [0, 0, 0] initial state.................... 59

Table 5. 4: RMS values of linear trajectory tracking error with [2, 1, pi/2] as an initial state 61

Table 5. 5:RMS values of the eight-shape trajectory tracking error with [0, 0, 0] initial state 64

LIST OF ACRONYMS

3DOF	Three Degrees of Freedom
AFSMC	Adaptive Fuzzy Sliding Mode Control
BSC	Backstepping Controller
CoM	Center of Mass
DC	Direct Current
DWMR	Differential Wheeled Mobile Robot
FNPN	Fuzzy Neural Petri Net
GA	Genetic Algorithm
IAE	Integral Absolute Error
ITAE	Integral Time-Weighted Absolute Error
MIMO	Multi-Input and Multi-Output
NPID	Nonlinear Proportional Integral Derivative
NWMR	Nonholonomic Wheeled Mobile Robot
PID	Proportional Integral Derivative
SMC	Sliding Mode Controller
TWMR	Three Wheeled Mobile Robot
UAV	Unmanned Aerial Vehicles
UUB	Uniformly Ultimate Bounded
WMR	Wheeled Mobile Robot

ABSTRACT

Over the past three decades, the expanding application of mobile robots in industry, military, home, and public service has piqued the interest of many researchers in wheeled mobile robot trajectory tracking control. Reducing the trajectory reference tracking error of a wheeled mobile robot is a difficult issue because of the nonlinear nature of the system. This study introduces an Adaptive Fuzzy Sliding Mode Controller specifically designed for trajectory tracking of two-wheeled mobile robots. The approach considers both the kinematic and dynamic models, utilizing a method based on Lagrangian mechanics. Stability is ensured through the Lyapunov stability function, which confirms reliable performance of the robot. The Adaptive Fuzzy Sliding Mode Controller effectively manages both position and speed control. Simulations conducted using Matlab/Simulink tested the controller's effectiveness in real-world scenarios, particularly when facing unknown disturbances and variations in initial position. The results revealed impressive tracking performance, with settling time of 0.1203 sec and a peak value of tracking error 0.019 m. Overall, the findings highlight the controller's strong capability to maintain precise movement even under challenging conditions.*

Keyword*: Adaptive Fuzzy Sliding Mode controller, Trajectory Tracking, Three-Wheeled Mobile Robot*

CHAPTER ONE

1. INTRODUCTION

The non-holonomic nature of mobile robots, stemming from non-integrable differentiable constraints, has captivated the robotics and control scientific community. Their kinematic limitations within non-holonomic mechanical systems make them intriguing subjects of study. Direct control law derivation is not feasible due to these constraints. As research into mobile robotics and their diverse configurations expands, so too does control research, driven by their potential applications in military tactics, entertainment, healthcare, industrial settings, hazardous environments, and domestic use. (Morales, Herrera, & Camacho, 2021).

When the robot's interest point is anticipated to follow the intended trajectory, this is known as the tracking trajectory. This means that the robot's reference point and the path's current reference point should be as close to one another as feasible. While functioning in well-known map locations, there are numerous useful applications for controlling a mobile robot to follow a given course. A trajectory planner is a necessary component of the autonomous navigation algorithm. It creates the best possible route to reach a given place while simultaneously patrolling the designated region and avoiding various obstacles. The robot has to basically follow the trajectory after it is formed. As a result, the autonomous navigation algorithm heavily relies on the trajectory tracking control. Non-holonomic constraints are kinematic restrictions that affect a significant portion of mobile robots and cannot be reduced to holonomic ones.

The creation of control principles that compel the robot to approach and adhere to a parameterized reference is connected to the application of trajectory tracking. Generally speaking, the goal is to identify the control actions that, during each sample interval, cause the mobile robot to arrive at the Cartesian position (x, y) with a predetermined orientation ψ. The mobile robot tracks the intended course as a result of these coordinated operations. The present difficulty is designing strong and accurate controllers that are simple to implement and respond correctly to various environmental disturbances. (Guevara, Camacho, Rosales, Guevara, & Scaglia, 2019).

1.1 Background of the Study

One type of robotic vehicle that is not restricted to a certain physical area and may roam around its surroundings is a mobility robot. It can navigate in an uncontrolled environment either fully or partially autonomously, meaning it doesn't require an electromechanical guidance device or a physical operator. A wheeled mobile robot (WMR) is a type of mobile robot that uses the rotation of a wheel to navigate its environment. Because it can replace humans in many fields, it is widely used. In the public and private sectors, the military, industry, manufacturing, and distribution companies, for example (Cen & Singh, 2021).

1.1.1 Types of Mobile Robot

Generally speaking, mobile robots can run, slide, roll, jump, walk, skate, swim, or fly, depending on how the environment interacts with them. Based on how they move, mobile robots can be divided into the following types: Air-based, water-based, and land-based (wheeled mobile robots, walking or legged mobile robots, tracked slip/skid locomotion, hybrids) (Rubio et al., 2019). Wheeled mobile robots are utilized on smoother ground since they are faster and nimbler there, while legged mobile robots are employed in difficult terrain conditions were walking is typically favored. Other types of mobile robots are utilized when we need to operate in three-dimensional spaces, such as Unmanned Aerial Vehicles (UAV) or Unmanned Underwater Vehicles. (KALYONCU & DEMİRBAŞ, 2017).

1.1.2 Wheeled Mobile Robot

Autonomous vehicles or robots are part of a challenging study topic in mobile robotics, with wheels being one of the most substantial components for robot mobility. They will play a critical role in transportation, logistics, and distribution. It travels on a smooth surface, nonrugged terrain, and simply where employing wheels is possible. The construction of WMR is easy to design, build, and program than a robot using tracks or legs. A wheeled mobile robot is less costly and less complicated when compared with a legged mobile robot (typically a four-legged robot). Another benefit is that there aren't many major balancing difficulties because the robot is usually in contact with a surface. Wheels' primary drawback is their inability to move on rough terrain, jagged edges, or areas with little friction around the driven wheel's axle, the contact point, and the rollers. (Rubio, Valero, & Llopis-Albert, 2019).

In applications involving wheeled mobile robots, four kinds of wheels are used: Fixed standard wheel; ball or spherical wheel; Swedish wheel; Castor wheel. Both the castor wheel and the fixed standard wheel are directed since they revolve around the same axis. The castor wheel revolves around an offset axis and applies force to the robot chassis during steering, while the conventional wheel may achieve steering motion with minimal side effects.

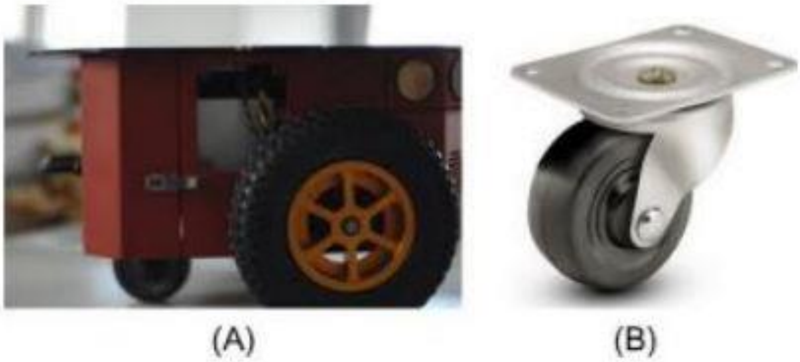

Figure 1.1: (A) fixed standard wheel, (B) castor wheel *(Tzafestas, 2014)*

Compared to the conventional wheel, the Swedish and spherical wheels have less direction limitation. The Swedish wheel offers minimal resistance in a different direction while functioning in the same manner as a conventional wheel. The wheel can move kinematically in this arrangement even if it is only driven along one principal axis (the axle). This is a major advantage. (Rubio, Valero, & Llopis-Albert, 2019).

Figure 1.2: Swedish wheels *(Tzafestas, 2014)*

Thus, as the use of a wheeled mobile robot expanded, the robot's performance became a crucial concern for researchers. When it comes to kinematic control, the robot's performance in trajectory tracking is determined by the position and orientation errors, whereas in dynamic control, the robot's performance is impacted by the velocity error. Three control problems determine how a wheeled mobile robot moves or navigates: line or path following, point stabilization, and reference trajectory tracking. (Fierro R. & Lewis, 1997). The only thing needed to solve the straightforward path following problem is a change in angular velocity to close the gap between the mobile robot and a given path. The robot follows a track based on a given trajectory in reference or trajectory tracking. In the case of point or posture stabilization, the system tries to reach the desired posture goal in any path.

Controlling a robot's location on a predefined trajectory with the least amount of position and orientation error is one of reference tracking's primary goals. Trajectory tracking and motion control are not independent in nonholonomic robots because of nonholonomic constraints. In the robotics area, trajectory tracking of nonholonomic robots entails calculating trajectories between a beginning and a finish point, taking mechanical constraints into account, and directing the robot to follow the suggested trajectories.

The goal of a wheeled mobile robot trajectory or reference tracking control is to maintain a robot's location with the least amount of orientation and positional error possible while following a predefined trajectory. This is particularly true for nonholonomic robots, as nonholonomic constraints prevent tracking trajectories and motion control from being autonomous. Determining trajectories from a starting point to a final point while taking mechanical constraints into account and directing the robot to follow the anticipated trajectories is known as trajectory planning for nonholonomic robots in the robotic domain. (Tiep, Lee, Im, Kwak, & Ryoo, 2018). Numerous studies addressing the problem of wheeled mobile robot trajectory control have been introduced. Since path following is a straightforward and time function independent trajectory, the majority of them concentrate on a nonholonomic restriction that relies on posture stability and trajectory monitoring. (Uddin, 2018). Robots are stabilized at a reference position through posture stabilization, while they follow a reference trajectory through trajectory tracking.

Among the many benefits of the sliding mode control are its insensitivity to shocks and model errors that satisfy the matching requirement, as well as its finite time convergence to a stable manifold. Its reaching phase, chattering phenomenon, and susceptibility to mismatched disruption are some of its drawbacks. It is fascinating to remove the reaching phase and reduce the impact of the mismatched disturbances in order to improve the robustness of the SMC throughout the motion. Applying the integral sliding mode design will enable you to do this. (Zhao, H., D., Han, & Lee, 2015). By imposing sliding mode during the whole system response, the integrated SMC aims to do away with the reaching phase. By allowing the system to begin on the sliding manifold at the beginning condition, this integral term removes the need for the reaching phase. There are two defined controllers. (Chen, Li, & Yeh, 2009): two types of controls: discontinuous (corrective control) and continuous. The discontinuous control is used to reject matched disturbances and reduce unmatched ones, while the continuous controller, which is nonlinear continuous feedback, is intended to stabilize the nominal system.

1.2 Statement of the Problem

Trajectory tracking for wheeled mobile robots (WMRs) is a complex control problem. External factors like noise, terrain interactions, and sensor errors can cause the robot to deviate from its intended trajectory. Moreover, achieving a perfectly error-free trajectory is practically unattainable due to the infinite number of paths between two points and the need for precise knowledge of the robot's dynamics and external disturbances. Therefore, the focus in trajectory tracking is to develop control strategies that minimize the error while considering these limitations and ensuring the robot's safe and efficient navigation.

Given the complexities and challenges associated with trajectory tracking for wheeled mobile robots, it is a compelling area of research. Understanding and addressing these issues can significantly enhance the performance and capabilities of autonomous robotic systems. To that end, this study focuses on investigating the trajectory tracking and velocity control problems for wheeled mobile robots. Numerous control theories and controllers have been proposed to manage the velocity and orientation of these robots. As a potential solution to the aforementioned challenges, this study introduces an adaptive fuzzy sliding mode controller. This controller is specifically designed to address the trajectory control problem in three-wheeled mobile robots.

1.3 Objective

1.3.1 General Objective

The main goal of this thesis work is to control the trajectory tracking of a WMR using an adaptive fuzzy sliding mode controller.

1.3.2 Specific Objective

The specific objectives to achieve this thesis work are: -

- Model the kinematics and dynamics models of a nonholonomic WMR,
- Create a fuzzy sliding mode controller that is adaptable for tracking the trajectory of WMR.
- Simulate the overall system using MATLAB/SIMULINK software to analyze the trajectory tracking performance of the proposed controller,

1.4 Scope of Thesis

This research focuses on controlling the trajectory of a differential-drive three-wheel mobile robot using an adaptive fuzzy sliding mode controller. A mathematical model (kinematics and dynamics) is determined for a two-wheel differential-drive mobile robot with one passive wheel. The system is executed using MATLAB/SIMULINK software.

1.5 Significance of the Study

The wheeled mobile robot is an autonomous mobile robot propelled by a DC motor. This mobile robot uses an electric energy source (specifically a battery), which plays a vital role in decreasing environmental pollution. Many organizations, particularly warehouses or distribution companies, employ carts or forklifts to deliver loaded items from one area to another. This has a side effect. A cart, for example, is extremely difficult for the elderly and disabled to operate. A forklift, or a tiny vehicle employed in a warehouse or distribution company, uses gasoline as its source of energy. However, employing fuel as an energy source increases pollution and makes the work environment less comfortable. As a result, the suggested system uses electricity and automation, providing comfort to all users.

1.6. Limitations of the Study

This research is focused on the trajectory control of a two-wheeled differential-drive mobile robot equipped with a single passive wheel. To simplify the analysis, the mathematical models developed for the robot assume that the two actuated wheels are solely responsible for steering and that factors like rolling, skidding, and the influence of the passive wheel on system dynamics are negligible. Due to the challenges associated with hardware acquisition, timing, and component availability, this study is limited to simulation-based experiments. The system was implemented and evaluated using MATLAB/SIMULINK software.

CHAPTER TWO

2. LITERATURE REVIEW

2.1 Introduction

In recent times, many researchers have tried to control the trajectory of a mobile robot with or without constraints by using different controllers and control theories. These control theories and controllers were based on linear or nonlinear control rules or combining linear and nonlinear controllers. Trajectory Control of a mobile robot depends on a tracking control problem like the reference or trajectory problem and a posture stabilization problem. Different control methods are used to control a wheeled mobile robot’s trajectory tracking and posture stabilization.

This study concentrated on the kinematic-based and dynamic using an adaptive fuzzy sliding mode controller. Several previous works are reviewed and discussed in the next topic, with a gap obtained.

2.2 Related Work of Studies

In work by (KALYONCU & DEMİRBAŞ, 2017), A kinematic controller based on backstepping with a combined PID for differential drive mobile robots was proposed. The kinematic controller is used to calculate the desired velocity within a possible minimum pose error, and the PID controller is used for right and left DC motor control speed adjustment. A Lyapunov function stability analysis checks the stability of a system. This work doesn’t consider dynamic control.

The literature (Saleh L. et al., 2018) introduces the fuzzy neural Petri net (FNPN) controller that is utilized to operate the mobile robot on wheels. In order to achieve the required velocity and azimuth, the path planning controller problem has been resolved utilizing two-FNPN controllers. For FNPN controller settings, the particle swarm optimization (PSO) method is utilized to determine the ideal values. In order to overcome the trajectory problem, the suggested controller has a more effective control method. By adjusting the robot's velocity and azimuthal angle but not its position, the authors attempted to regulate the robot's trajectory tracking. Without taking dynamic control into account, the robot's position is not closed loop or feedback to minimize position error.

The paper (HASSANI, MAALEJ, & REKIK, 2020) investigates the path tracking control problem for a mobile robot using a robust backstepping controller. The controller ensures accurate path following with minimal error. A genetic algorithm is employed to optimize the controller's performance. Simulation results demonstrate the robot's ability to follow various reference paths. However, dynamic control considerations are not included in this study.

In this paper, (Benaziza, Slimane, & Mallem, 2017) a quasi-sliding mode control method is suggested to controls form the basis of the trajectory tracking. Initially, a quasi-sliding mode control of the angular velocity is suggested with the goal of asymptotically stabilizing the angle error and bringing it to zero in a little amount of time. Second, using Lyapunov theory to guarantee asymptotic stability and reduce position error to zero, a global sliding mode control for linear velocity is suggested.

This paper, (Wang & Liu, 2018) addresses trajectory tracking of an omni-directional mobile robot with three mecanum wheels and a fully symmetrical configuration. The omni-directional wheeled robot outperforms the non-holonomic wheeled robot due to its ability to rotate and translate independently and simultaneously. A kinematics model of the omni-directional mobile robot is established and a model predictive control algorithm with control and system constraints is designed to achieve point stabilization and trajectory tracking. Simulation results validate the accuracy of the established kinematics model and the effectiveness of the proposed model predictive control controller.

In work by (Ahmed, Elsayed, & Ibrahim, 2016). investigates the trajectory tracking control of a nonholonomic wheeled mobile robot (NWMR). A backstepping method is employed for kinematic control design, while an adaptive sliding mode controller is proposed for dynamic control, addressing the limitations of conventional tracking laws. A fuzzy logic approach is integrated into the sliding mode controller design. Simulations involving sinusoidal trajectory tracking demonstrate the effectiveness of the proposed adaptive fuzzy sliding mode control in enhancing dynamic performance, reducing tracking error, and mitigating chattering.

Summary of the related works:

The literature review examines various approaches to trajectory tracking control for mobile robots. Key methods include backstepping, fuzzy logic and sliding mode control. While these methods have shown promise in achieving accurate tracking, several limitations and gaps are evident:

- Lack of Dynamic Control: Many studies focus on kinematic control without considering dynamic factors, which can limit performance in real-world scenarios.

- Limited Practical Implementation: Some works are primarily theoretical, with limited or no practical implementation.
- Chattering Issues: Sliding mode controllers can suffer from chattering, which can degrade performance and actuator life.

CHAPTER THREE

3. TWMR SYSTEM MODELING AND METHODOLOGY

3.1. Overview

This chapter will give details of the proposed system models and modeling, the procedures followed, and the materials used for report writing and result analysis. The mathematical model derivation, representation, controllability, and an open-loop systems simulation analysis will be presented

3.2. Methods or Procedures Followed

The following procedure, shown in figure 3.1, will be used throughout this thesis to accomplish a proposed objective.

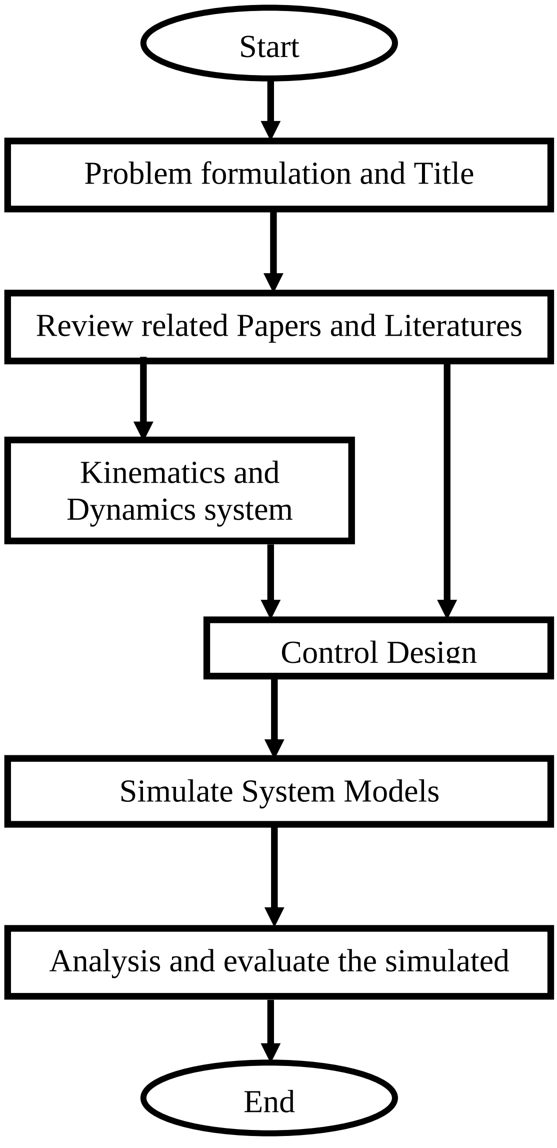

Figure 3.1: Proposed workflow or procedures

Depending on the selected title and a formulation of the problem, the related papers and literature will be selected and reviewed for a wheeled mobile robot and its control mechanism. All the data was gathered from secondary sources via online data access, with the help of a skilled person or researcher, and by experience sharing and observing the working principles of an electric vehicle at Marathon Motor Plc, Addis Ababa. The collected data from selected literature and personnel will be analyzed. Based on data collected for a motor and wheeled mobile robot, the kinematics and dynamics models of WMR will be developed. For developed WMR models, two control methods are employed, i.e., a kinematic-based adaptive fuzzy sliding mode controller will be used to control the position of a robot, and a nonlinear PID controller is used to control the velocity of a WMR robot. The proposed system models and the controller will be simulated on MATLAB/Simulink software to test and analyze the controller's performance.

3.3 The Block Diagram of TWMR with the Proposed Control Method

This work focused on a two-wheeled differential drive mobile robot with a free castor or passive wheel. Two similar DC motors drive the two wheels. I. e., the right wheel is actuated by a right motor, and the other motor also actuates the left wheel. Each motor is controlled separately to achieve a differential drive mechanism. A passive wheel is used to balance a robot's body frame.

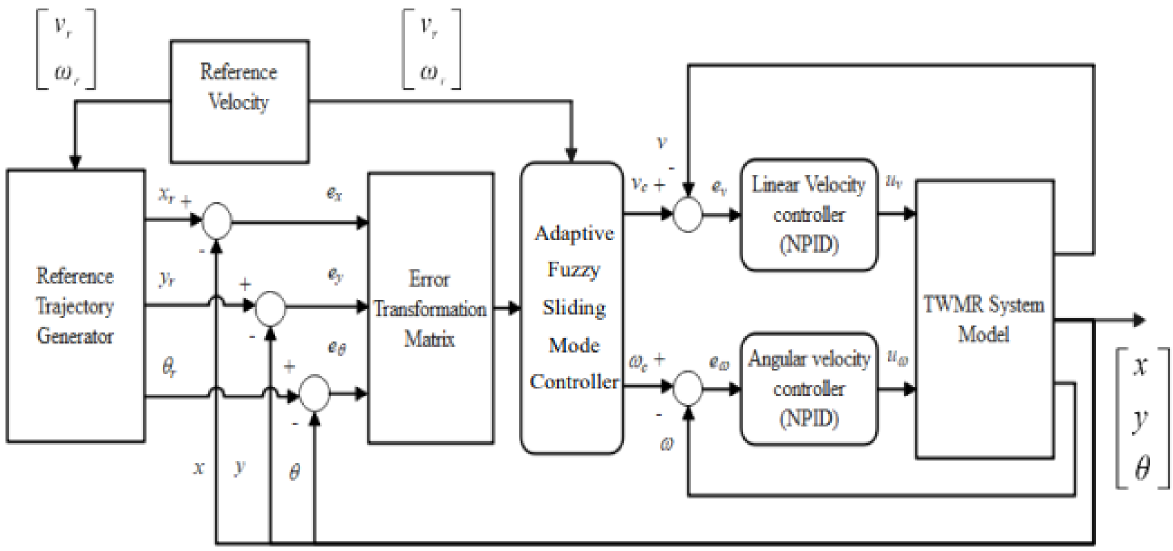

Figure 3.2: Proposed Block diagram of TWMR

3.4 Modeling and Description of the TWMR System

System modeling of a differential-drive mobile robot platform in this work includes kinematic and dynamic modeling as well as modeling of the system actuators. For a system, kinematic modeling is concerned with the geometric relationships and mathematics of motion without considering influencing forces. On the other hand, dynamic modeling is the study of motion in which forces and energy are represented and included, while actuator modeling is required to determine the interaction between the control signal and the mechanical system's input. A derivation of an appropriate system model makes a system more understandable and controllable.

3.4.1 Representing Robot Coordinate

The primary function of the coordinate systems is to represent the robot's position. There are two coordinate systems that are widely used for system model development: Cartesian and Polar coordinates. Hence, because of the ease and simple representation of Cartesian coordinates, system modeling is done in Cartesian coordinates. The following cartesian coordinate systems are used for mobile robot modeling and control purposes:

To specify the robot's position on the coordinate configuration frame, the relationship between the global frame and the robot's local frame is determined, as shown in Figure 3.3.

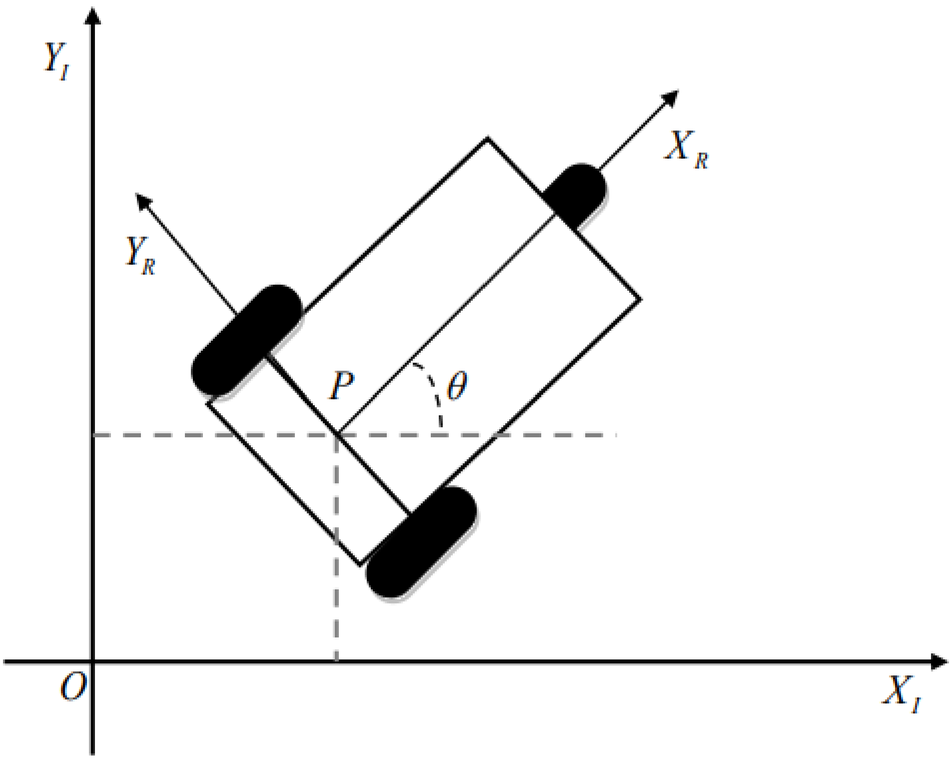

Figure 3.3: Inertial and robot frame coordinate representation

- The inertial reference frame (X_I, Y_I): This is a fixed coordinate system in the plane of the robot. It defines a random inertial base on the plane as the global reference frame from some origin.
- Robot frame (X_R, Y_R): This is the coordinated system attached to the robot's body frame. It defines two axes relative to the robot chassis.

The axes X_I and Y_I are an arbitrary inertial frame on the plane as the global reference frame from the origin O. The basis (X_R, Y_R) is two local robot frame axes relative to P on the robot chassis. The position of P in the global reference frame is specified by the coordinates x and y, and the angular difference between the global and local frames is given by θ. The essential issue in a coordinate system representation is to map the motion along the axes of the global reference frame to the motion along the axes of the local robot frame. The inertial frame and the robot frame can be represented as follows (Ben Jabeur & Seddik, 2021);

$$\begin{cases} q_I = \begin{bmatrix} x_I & y_I & \theta_I \end{bmatrix}^T \\ q_R = \begin{bmatrix} x_R & y_R & \theta_R \end{bmatrix}^T \end{cases} \tag{3.1}$$

A typical orthogonal transformation matrix can be used to map the two frames. The movement of the local robot frame (X_R, Y_R) is mapped to the motion of the global reference frame (X_I, Y_I) using this matrix in the following manner:

$$\dot{q}_R = R(\theta)\dot{q}_I \tag{3.2}$$

The local robot frame (X_R, Y_R) to the global reference frame (X_I, Y_I) was mapped as:

$$\dot{q}_I = R(\theta)^{-1}\dot{q}_R \tag{3.3}$$

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

where the orthogonal transformation matrix is denoted by $R(\theta)$. Equation (3.2) illustrates a relationship that is useful for both robot velocities and inertial frames. This is a crucial duty for the kinematic equation drive of a TWMR in the following article.

3.4.2 Kinematic Modeling of TMWR

Robot kinematics modeling is concerned with how robots are arranged spatially, how their geometric parameters relate to one another, and what limitations are placed on their paths.

The kinematic equations are determined by the geometric structure of the robot. For instance, a mobile robot may have one or more wheels, as well as motion restrictions or not. Understanding stability, control techniques, and system dynamics can be aided by the notion of a kinematic model.

A kinematic equation establishes the robot's speed $\dot{q} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$ as a function of wheels speed $\left(\dot{\varphi}_R, \dot{\varphi}_l\right)$ and robot geometric parameters can be categorized as follows.

- Forward kinematics from:

$$\dot{q} = f(\dot{\varphi}_R, \dot{\varphi}_l, L, r, \theta) \tag{3.5}$$

- Inverse kinematics form:

$$\begin{bmatrix} \dot{\varphi}_r & \dot{\varphi}_l \end{bmatrix}^T = f(x, y, \theta) \tag{3.6}$$

3.4.2.1 The Forward Kinematics Model of Differential Drive WMR

A differential drive of WMR is shown in Figure 3.4, which has two wheels with a radius of r and a distance L from the center P, and free castor wheels to balance the robot setup body. The parameters that are used to derive kinematic and dynamic models are presented as follows (Martins & Bertol, 2022):

- *L*: The distance between the actuated wheel and the axis of symmetry in the y-direction,
- *C*: is the center of mass (CoM) at the center of gravity of the robot platform,
- *d*: The distance between the CoM and the driving wheel axis.
- *P*: The intersection of the axis of symmetry with the driving wheel axis,
- *r*: The radius of each wheel,
- m_c: a mass of WMR devoid of wheels and motors,
- m_w: The combined mass of each wheel and motor,
- m_t: The total mass of the robot,
- Ic : The inertia of WMR without wheels and motor about the y-axis through P,
- Iw : The inertia of each wheel and motor around the wheel axis,
- Im : The inertia of each wheel and motor about the y-axis is parallel to the wheel plane,
- I : total inertia of the robot,

- φ_r, φ_l : angular displacement of the right and left wheels,
- $\dot{\varphi}_r, \dot{\varphi}_l$: Angular velocity of the right and left wheels,
- v, ω: The robot's linear and angular velocity,
- w : velocity vector of linear and angular velocity.

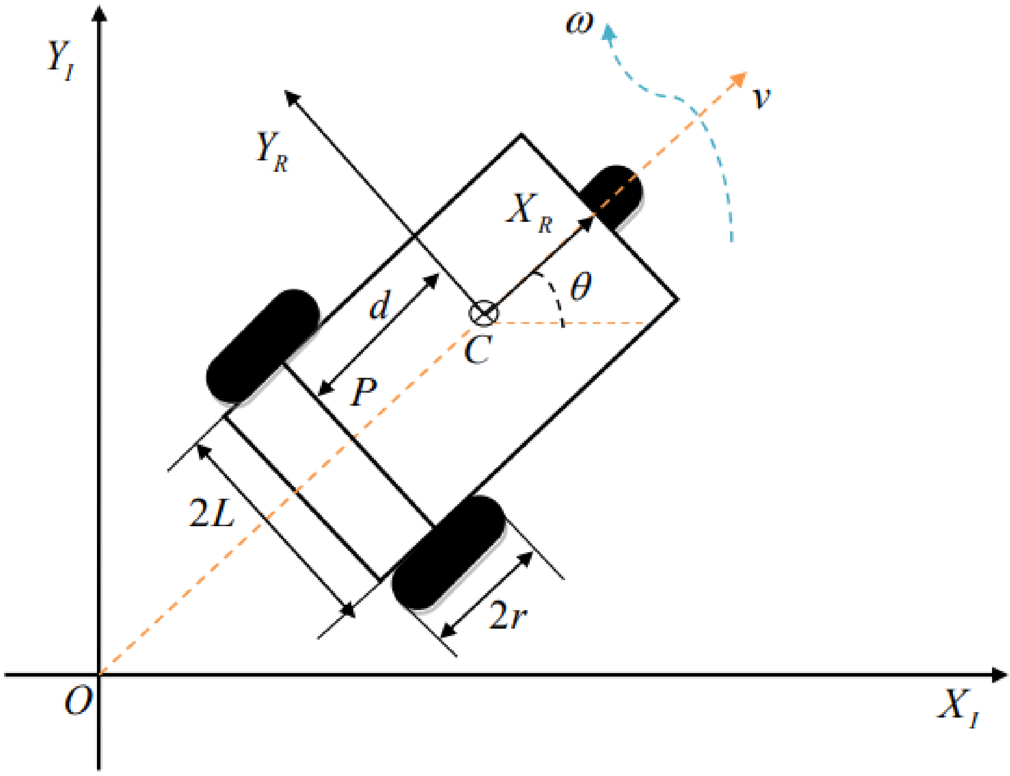

Figure 3.4: 2D velocity coordinate configuration of TWMR

From Equation (3.5) a forward kinematic model of DWMR is a function of the robot's wheel velocity and geometric parameters, representing the robot's velocities in an inertial frame. The velocity coordinate represents the kinematic model problem, as shown below (Benchouche W. M., 2021):

$$\dot{q}_I = \begin{bmatrix} \dot{x}_I & \dot{y}_I & \dot{\theta}_I \end{bmatrix}^T = f\left(\dot{\varphi}_r, \dot{\varphi}_l, L, r, \theta\right) \tag{3.7}$$

From Equations (3.2) and (3.3), we can compute the robot motion in a global reference frame on a local robot frame and vice versa. Therefore, to calculate the contribution of each wheel in a movement of the robot in the local robot frame $\dot{q}_R$ the following methods are employed (Siegwart R, 2011) .

In Figure 3.4, suppose that a robot moves forward along the positive x-plane of the local frame in alignment with a local robot frame. First, let us calculate the contribution of each wheel's rotating speed to the translation speed at point P in the direction of the positive x-plane of the local frame. Suppose one-wheel spins while the other wheel is stationary, then the robot moves instantaneously at half speed since P is the halfway point between two wheels, that is:

$$\begin{cases} \dot{x}_{R1} = \dfrac{1}{2} r\dot{\varphi}_r \\ \dot{x}_{R2} = \dfrac{1}{2} r\dot{\varphi}_l \end{cases} \tag{3.8}$$

In a differential drive mechanism, Equation (3.8) is added together to get the contribution of $\dot{x}_R$ component. The contribution $\dot{x}_R$ component is.

$$\dot{x}_R = \frac{1}{2} r(\dot{\varphi}_r + \dot{\varphi}_l) \tag{3.9}$$

The contribution of $\dot{y}_R$ is zero because there is no sideways movement in the robot reference frame. Therefore, the contribution of $\dot{y}_R$ component is:

$$\dot{y}_R = 0 \tag{3.10}$$

Finally, the contribution of a rotational component to the robot reference frame is computed independently and added together. Let us consider the right wheel, and a forward movement of this wheel results in an anticlockwise rotation at point P. Suppose the right wheel rotates alone, and the robot pivots along with the left wheel and the robot chassis. The angular velocity, $\dot{\theta}_r$ at point P due to wheel movement along the arc of a circle of radius 2L is given as:

$$\dot{\theta}_r = \frac{r}{2L}\dot{\varphi}_r \tag{3.11}$$

The same procedure is applied to the left wheel, except the forward spin of the left wheel results in a clockwise rotation at point P:

$$\dot{\theta}_l = \frac{r}{2L}\dot{\varphi}_l \tag{3.12}$$

Moreover, by adding Equations (3.11) and (3.12) the rotational velocity component is given as:

$$\dot{\theta}_R = \frac{r}{2L}(\dot{\varphi}_r - \dot{\varphi}_l) \tag{3.13}$$

Therefore, by combining Equations (3.9), (3.10) and (3.13) the translational speed and the rotational velocity in the robot reference frame as follows:

$$\dot{q}_I = R^{-1}(\theta)\begin{bmatrix} \frac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l) \\ 0 \\ \frac{r}{L2}(\dot{\varphi}_r - \dot{\varphi}_l) \end{bmatrix} \tag{3.14}$$

Moreover, the inverse orthogonal transformational matrix $R^{-1}(\theta)$ is:

$$R^{-1}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

By substituting Equation (3.15) into Equation (3.14) a DWMR velocity in an inertial frame became:

$$\dot{q}_I = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \frac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l) \\ 0 \\ \frac{r}{L2}(\dot{\varphi}_r - \dot{\varphi}_l) \end{bmatrix} = \frac{r}{2}\begin{bmatrix} (\dot{\varphi}_r + \dot{\varphi}_l)\cos(\theta) \\ (\dot{\varphi}_r + \dot{\varphi}_l)\sin(\theta) \\ \frac{1}{L}(\dot{\varphi}_r - \dot{\varphi}_l) \end{bmatrix} \tag{3.16}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \frac{r}{2}\begin{bmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix}\begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} \tag{3.17}$$

The kinematic equation obtained above shows the DWMR velocity in an inertial frame.

3.4.2.2. Nonholonomic Kinematic Constraints

A mobile robot's holonomy can be defined in two ways: holonomic and nonholonomic. Nonholonomic systems have mathematically nonintegrable constraints, whereas holonomic systems have integrable constraints. On the other hand, a holonomic system can move in any direction in the configuration space, i.e., robot movement is not restricted by direction. In contrast, nonholonomic systems are those in which the velocities and other derivatives of the position are constrained, i.e., a robot's motion is limited. Nonholonomic robots are common because of their basic design and ease of operation. It also has fewer degrees of freedom than holonomic mobile robots by nature.

Several assumptions on wheel motion will cause the robot kinematic constraints to simplify the effort needed: movement on a horizontal axis; point contact between the wheels and ground; pure rolling, no slipping, skidding, or sliding; and no friction for rotation around contact points; steering axes orthogonal to the surface; and wheels connected by a rigid frame (Fufa & Ayenew, 2023).

In Figure 3.3, Three kinematic limitations on DWMR are presented by a wheel. The first restriction is that there is no non-slipping requirement, meaning that DMWR cannot slide sideways. That only moves in the direction of the symmetry axis of the actuated wheels. Algebraically, this constraint is written as (Mohareri, (2012).

- No lateral slip constraints:

$$\dot{y}\cos(\theta) - \dot{x}\sin(\theta) - d\dot{\theta} = 0 \tag{3.18}$$

Where $\dot{x}$ and $\dot{y}$ the robot velocity components in the inertial frames. The other constraints are related to the motion of the wheels, which means the actuated wheels cannot rotate in the wrong direction. i.e., there are only pure rolling constraints.

- Pure rolling constraints:

$$\begin{cases} \dot{x}\cos(\theta) - \dot{y}\sin(\theta) + L\dot{\theta} - r\dot{\varphi}_r = 0 \\ \dot{x}\cos(\theta) - \dot{y}\sin(\theta) - L\dot{\theta} - r\dot{\varphi}_l = 0 \end{cases} \tag{3.19}$$

By rewriting Equation (3.19), we obtain:

$$\begin{cases} v + L\omega = r\dot{\varphi}_r \\ v - L\omega = r\dot{\varphi}_l \end{cases} \tag{3.20}$$

Since,

$$v = \dot{x}\cos(\theta) + \dot{y}\sin(\theta) \tag{3.21}$$

$$\omega = \dot{\theta} \tag{3.22}$$

Equation (3.20) are used to relate the velocities of DWMR, and the angular velocities of wheels within the center of mass (CoM) are written as:

$$\begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} = \phi \begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & L \\ 1 & -L \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.23}$$

where ϕ is the velocity conversion matrix. Similarly, a robot velocity vector is expressed as:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \phi^{-1} \begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} \tag{3.24}$$

The three-constraint in Equations (3.18) and (3.19) can be written as a generalized coordinate vector relation. The relationship is represented in a matrix form like as (Gao X, 2021):

$$A(q)\dot{q} = 0 \tag{3.25}$$

where q is the coordinate configuration space written as a function of the x-axis, y-axis, heading angle theta, right, and left wheel displacements as follows:

$$q = \begin{bmatrix} x & y & \theta & \varphi_r & \varphi_l \end{bmatrix}^T \tag{3.26}$$

Furthermore, its time derivatives are given by:

$$\dot{q} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} & \dot{\varphi}_r & \dot{\varphi}_l \end{bmatrix}^T \tag{3.27}$$

where φ_r and φ_l are right and left angular displacement, and its time derivatives are equivalent to the angular speed of wheels $\dot{\varphi}_r$ and $\dot{\varphi}_l$, respectively. By rewriting Equation (3.25) as:

$$A(q)\dot{q} = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & -d & 0 & 0 \\ \cos(\theta) & \sin(\theta) & L & -r & 0 \\ \cos(\theta) & \sin(\theta) & -L & 0 & -r \end{bmatrix} \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} & \dot{\varphi}_r & \dot{\varphi}_l \end{bmatrix}^T \tag{3.28}$$

The obtained equation is functional when driving dynamic modeling and considering a constraint. If the inertia and mass of the wheels are neglected, DWMR satisfies the pure rolling and non-slipping constraint conditions (Fierro R. & Lewis, 1997). As a result, the nonholonomic matrix A(q) is reduced to the matrix shown below:

$$A(q) = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & -d \end{bmatrix} \tag{3.29}$$

The reduced constraints show that displacements occur only in the direction of the symmetry axis of the actuated wheels. Then the generalized coordinate vector is modified as:

$$q = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \tag{3.30}$$

It is crucial to highlight that the velocity vector is of dimension because the system contains a configuration space (n), a generalized coordinate vector (q), and the number of restrictions (p). $m = n - p$; in this case, $m = 2$ (corresponds to the degrees of freedom). The Jacobian matrix S(q) is formed by a set of linearly independent and smooth vector fields distributed in the null space of A(q) are used to transform velocity term in an inertial frame (Rabbani & Memon, 2021). i.e.

$$A(q)S(q) = 0 \tag{3.31}$$

The Equation (3.31) is used to find auxiliary velocity as a function of time $w(t) \in R^{p\times 1}$ for all t. The relationship between coordinate q and velocity vector $w(t)$ can be written as:

$$\dot{q} = S(q)w(t) \tag{3.32}$$

where $w(t)$ is the velocity vector given as $w(t) = \begin{bmatrix} v & \omega \end{bmatrix}^T$ and $S(q)$ is given by.

$$S(q) = \begin{bmatrix} \cos(\theta) & -d\sin(\theta) \\ \sin(\theta) & d\cos(\theta) \\ 0 & 1 \end{bmatrix} \tag{3.33}$$

The Jacobian matrix of Equation (3.33) has utilized the DWMR velocity terms of $\begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$, i.e.,

$$\dot{q} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T = \begin{bmatrix} \cos(\theta) & -d\sin(\theta) \\ \sin(\theta) & d\cos(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.34}$$

The kinematic model equation obtained in Equation (3.34) is used throughout this thesis for system control and simulation. In the next topic, a dynamic model of a DWMR is derived.

3.4.3. Dynamic Modeling of TWMR Using Lagrangian Approach

A mobile robot system having an n-dimensional configuration space R, with generalized coordinates $q = \begin{bmatrix} q1, q2, \ldots, qn \end{bmatrix}$ and subject to m input constraints, can be described as (Mohamed & Hamza, 2019):

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = E(q)\tau - A^T(q)\rho \tag{3.35}$$

where $H(q)$ is an inertial matrix represented as $n \times n$ positive definite matrix, $C(q,\dot{q})$ is Coriolis and Centripetal torques matrix represented as $n \times n$, $F(\dot{q})$ is surface friction force matrix, $G(q)$ is a gravitational vector, ρ is a vector associated with Lagrange multiplier, $E(q)$ is the control input of $n \times m$ transformation matrix, $A^T(q)$ is matrix associated with kinematic constraints, τ is input torque vector and, τd an unknown disturbance. Dynamic modeling of the mobile robot is determined in two ways:

- Lagrangian approach: which is an energy-based approach,
- Newton-Euler approach: is based on vector mechanics.

Lagrange's equations are differential equations that take into account both the system's immediate work and energy. The independence of the generalized coordinates is necessary to derive the Lagrange equation for holonomic systems. The Lagrangian approach is used to determine a dynamic model because it is simpler to formulate and implement than the Newton-Euler approach. Precisely knowing all the forces applied to the system is challenging in the Newton-Euler approach.

The general form of the Lagrange method for the case of DWMRs with fixed standard wheels is given by (Ahmad Abu Hatab, 2013):

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = E(q)\tau - A^T(q)\rho \qquad (3.36)$$

where q and $\dot{q}$ are given as in Equation (3.26) and (3.27). The total energy of a Lagrangian system is given as L = T – W, where T is kinetic energy, and W is potential energy. A DWMR has zero potential energy because movement is always along a horizontal plane, with no vertical direction movement. In the simple formula, the total potential energy is given as:

$$W = 0 \qquad (3.37)$$

And total kinetic energy as:

$$T = \frac{1}{2} q^T H(q)\dot{q} \qquad (3.38)$$

Equation (3.36) rearranged in standard form as:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} = E(q)\tau - A^T(q)\rho \qquad (3.39)$$

A Coriolis and Centripetal torques matrix are dependent on the posture and velocity with the $n \times n$ dimension given as follows:

$$C(q,\dot{q})\dot{q} = \frac{d}{dt}\left[H(q)\right]\dot{q} - \frac{\partial T}{\partial q} = \frac{d}{dt}\left[H(q)\right]\dot{q} - \frac{1}{2}\frac{\partial}{\partial}\left[\dot{q}^T H(q)\dot{q}\right] \tag{3.40}$$

However, owing of uncertainty and disturbance from mechanical design and robot body interface, such as friction between parts, and wheels, and other body contacts, Equation (3.39) is a disturbance-free system dynamic equation that is not physically relevant. When we take into account a dynamic system with uncertainties and disturbances in Equation (3.39), we obtain:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_d = E(q)\tau - A^T(q)\rho \tag{3.41}$$

where τ_d − $n \times 1$ vector denotes uncertainties and disturbances, $A(q)$ − $p \times n$ matrix associated with nonholonomic constraints, $\rho - p \times 1$ vector of a constraint force (Lagrangian multiplier), τ $- p \times 1$ vector control or input torques.

The dynamics system obtained in Equation (3.41) has the following properties to determine the stability analysis of the control system (Tzafestas, 2014):

- Property 3.1: Limitation - The matrix $H(q)$, the norm of a matrix $C(q,\dot{q})$, and the vector τd are bounded,
- Property 3.2: Skew-symmetry - The matrix $\dot{H}(q) - 2\,C(q,\dot{q})$ is skew-symmetric.

3.4.3.1. Dynamic Model Using Wheel Constraint

Two fixed wheels are standard on a DWMR. To create a differential drive, a motor powers each of these wheels independently. By taking the dynamics of the wheels into consideration, we may obtain kinetic energy from Equation (3.38). The DWMR kinetic energy is derived using the following assumptions, and the individual kinetic energy is ascertained as follows. (Benchouche, Mellah, & Bennouna, 2021):

- Kinetic energy (T1) of WMR without the wheels,
- Kinetic energy (T2) of actuated wheels in the plane only,
- Kinetic energy (T3) of all wheels considering the orthogonal plane.

Total kinetic energy T is given as.

$$T = T_1 + T_2 + T_3 \tag{3.42}$$

A. Kinetic energy (T1) of WMR without the wheels:

By taking into account the vertical velocity vy and the horizontal velocity vx, the kinetic energy is found. The free-body diagram of the mobile robot, as seen in Figure 3.4, is represented by the DWMR velocities in Figure 3.5.

.

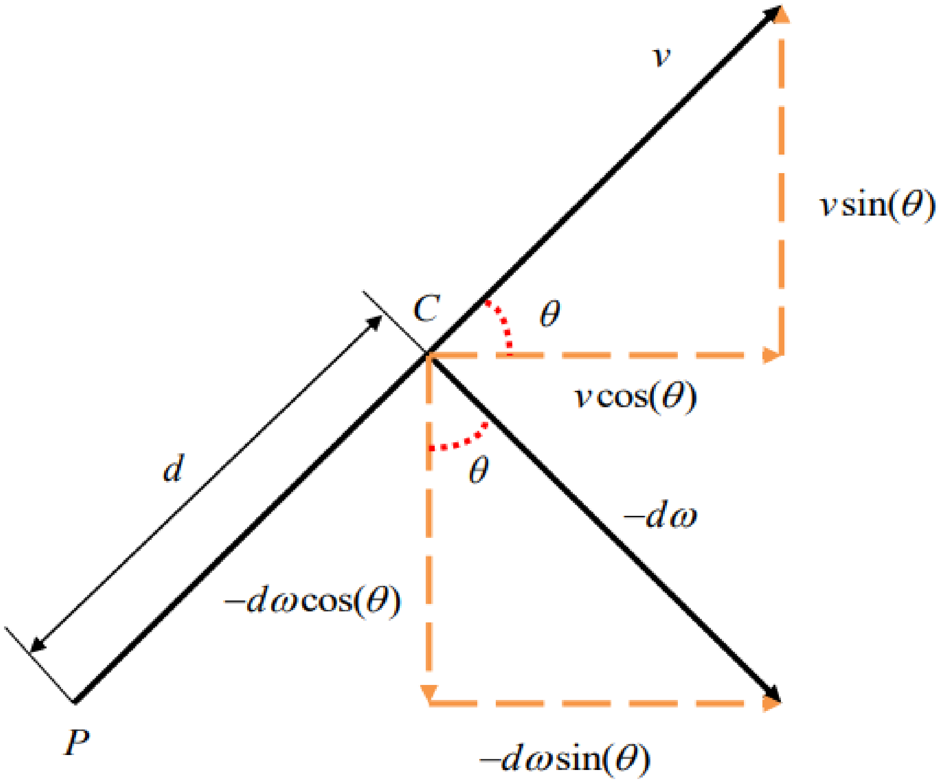

Figure 3.5: Velocity representation of DWMR

From the velocity representation of differential drive WMR in Figure 3.5 and Equation (3.34), I will get v_x and v_y as shown below:

$$\begin{cases} v_x = \dot{x} + d\dot{\theta}\sin(\theta), \\ v_y = \dot{y} - d\dot{\theta}\cos(\theta), \end{cases} \quad and \quad \omega = \dot{\theta} \tag{3.43}$$

We get the kinetic energy of DWMR without a wheel as:

$$T_1 = \frac{1}{2}\left(m_c\left({v_x}^2 + {v_y}^2\right) + I_c\dot{\theta}^2\right) \tag{3.44}$$

By substituting Equation (3.43) into (3.44), we get T1 in the following form:

$$\begin{aligned} T_1 &= \frac{1}{2}\left(m_c\left(\left(\dot{x} + d\dot{\theta}\sin(\theta)\right)^2 + \left(\dot{y} - d\dot{\theta}\cos(\theta)\right)^2\right) + I_c\dot{\theta}^2\right) \\ &= \frac{1}{2}m_c\left(\dot{x}^2 + \dot{y}^2 - 2\dot{y}d\dot{\theta}\cos(\theta) + 2\dot{x}d\dot{\theta}\sin(\theta) + d^2\dot{\theta}^2\right) + \frac{1}{2}I_c\dot{\theta}^2 \end{aligned} \tag{3.45}$$

B. Kinetic energy (T2) of actuated wheels in the plane only:

This kinetic energy is based on the actuated wheels in a wheel's plane. It is calculated based on wheel velocity, i.e., right and left wheels.

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \end{bmatrix} = \begin{bmatrix} v_x + L\dot{\theta}\cos(\theta) \\ v_y + L\dot{\theta}\sin(\theta) \end{bmatrix}, \; and \; \begin{bmatrix} \dot{x}_l \\ \dot{y}_l \end{bmatrix} = \begin{bmatrix} v_x - L\dot{\theta}\cos(\theta) \\ v_y - L\dot{\theta}\sin(\theta) \end{bmatrix} \tag{3.46}$$

Then the kinetic energy T2 is expressed as follows:

$$T_2 = \frac{1}{2} m_w \left(\dot{x}_r{}^2 + \dot{y}_r{}^2 + \dot{x}_l{}^2 + \dot{y}_l{}^2 \right) + \frac{1}{2} I_{mr} \dot{\theta}^2 + \frac{1}{2} I_{ml} \dot{\theta}^2 \tag{3.47}$$

After substituting a velocity Equation (3.46) into (3.47) and rearranging it, we get:

$$T_2 = m_w \left(\dot{x}^2 + \dot{y}^2 - 2\dot{y}d\dot{\theta}\cos(\theta) + 2\dot{x}d\dot{\theta}\sin(\theta) \right) + \dot{\theta}^2 \left(m_w \left(d^2 + L^2 \right) + I_m \right) \tag{3.48}$$

where $I_m = I_{mr} = I_{ml}$

C. Kinetic energy (T3) of all wheels considering the orthogonal plane:

This kinetic energy of a DWMR is obtained by considering the moment of inertia of wheels around the axis that passes through them. The kinetic energy T3 is given as:

$$T_3 = \frac{1}{2} I_w \dot{\varphi}_r^2 + \frac{1}{2} I_w \dot{\varphi}_l^2 = \frac{1}{2} I_w \left(\dot{\varphi}_r^2 + \dot{\varphi}_l^2 \right) \tag{3.49}$$

Finally, the total kinetic energy in Equation (3.42) became:

$$T = \frac{1}{2} m_t \left(\dot{x}^2 + \dot{y}^2 \right) + m_t \left(\dot{x}d\dot{\theta}\sin(\theta) - \dot{y}d\dot{\theta}\cos(\theta) \right) + \frac{1}{2} I\dot{\theta}^2 + \frac{1}{2} I_w \left(\dot{\varphi}_r^2 + \dot{\varphi}_l^2 \right) \tag{3.50}$$

where $m_t = m_c + 2m_w, I = m_c d^2 + I_c + 2m_w(d^2 + L^2) + 2I_m \; and \; \dot{\theta} = \omega$ as in Equation (3.22).

3.4.3.2. Dynamic Model Configuration and Simplification

The total Lagrangian energies of the above kinetic and potential energies were given as:

$$L = T - W, \; where, \; W = 0 \tag{3.51}$$

Therefore, total Lagrangian energy is equivalent to kinetic energy since there is no vertical movement, as shown below:

$$L = \frac{1}{2} \left(m_t \dot{x}^2 + m_t \dot{y}^2 + I\dot{\theta}^2 + I_w \dot{\varphi}_r^2 + I_w \dot{\varphi}_l^2 \right) + m_t \dot{x}d\dot{\theta}\sin(\theta) - m_t \dot{y}d\dot{\theta}\cos(\theta) \tag{3.52}$$

By substituting Equation (3.52) in Equation (3.36), we get the following results. From a generalized coordinate vector, we had q as in Equation (3.26) and q. as in Equation (3.27). So, the partial differential equations of Lagrangian energy with a specified coordinates are given as follows.

$$\frac{\partial L}{\partial \dot{x}} = m_t \dot{x} + m_t d\dot{\theta}\sin(\theta) \tag{3.53}$$

$$\frac{\partial L}{\partial \dot{y}} = m_t \dot{y} - m_t d\dot{\theta}\cos(\theta) \tag{3.54}$$

$$\frac{\partial L}{\partial \dot{\theta}} = I\dot{\theta} + m_t \dot{x}d\sin(\theta) - m_t \dot{y}d\cos(\theta) \tag{3.55}$$

$$\frac{\partial L}{\partial \dot{\varphi}_r} = I_w \dot{\varphi}_r, \;\; and \;\; \frac{\partial L}{\partial \dot{\varphi}_l} = I_w \dot{\varphi}_l \tag{3.56}$$

The time derivative of the Equation (3.53) through Equation (3.56) is as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = m_t \ddot{y} + m_t d\ddot{\theta}\sin(\theta) + m_t d\dot{\theta}^2\cos(\theta) \tag{3.57}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) = m_t \ddot{x} - m_t d\ddot{\theta}\cos(\theta) + m_t d\dot{\theta}^2\sin(\theta) \tag{3.58}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = I\ddot{\theta} + m_t d\left(\ddot{x}\sin(\theta) + \dot{x}\dot{\theta}\cos(\theta) - \ddot{y}\cos(\theta) + \dot{y}\dot{\theta}\sin(\theta)\right) \tag{3.59}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varphi}_r}\right) = I_w \ddot{\varphi}_r; \;\; and \;\; \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varphi}_l}\right) = I_w \ddot{\varphi}_l \tag{3.60}$$

The other derivatives are obtained as follows:

$$\frac{\partial L}{\partial x} = 0, \;\; \frac{\partial L}{\partial y} = 0, \;\; \frac{\partial L}{\partial \theta} = m_t \dot{x}\dot{\theta}d\cos(\theta) + m_t \dot{y}\dot{\theta}d\sin(\theta), \;\; \frac{\partial L}{\partial \varphi_r} = 0, \;\; \frac{\partial L}{\partial \varphi_l} = 0 \tag{3.61}$$

By substituting Equations (3.57) to (3.60) into Equation (3.36); the left side matrices in Equation (3.41) are given as follows:

$$H(q)=\begin{bmatrix} m_t & 0 & m_t d\sin(\theta) & 0 & 0 \\ 0 & m_t & -m_t d\cos(\theta) & 0 & 0 \\ m_t d\sin(\theta) & -m_t d\cos(\theta) & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix} \tag{3.62}$$

$$C(q,\dot{q})=\begin{bmatrix} 0 & 0 & m_t d\dot{\theta}\cos(\theta) & 0 & 0 \\ 0 & 0 & m_t d\dot{\theta}\sin(\theta) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.63}$$

From Equation (3.29), we have a reduced nonholonomic constraints matrix by neglecting wheel inertia. Therefore, the displacements occur only in the direction of the symmetry axis. So, the matrix in Equation (3.62) and (3.63) can be reduced to a 3x3 matrix, and Equation (3.39) is rewritten as:

$$\begin{bmatrix} m_t & 0 & m_t d\sin(\theta) \\ 0 & m_t & -m_t d\cos(\theta) \\ m_t d\sin(\theta) & -m_t d\cos(\theta) & I \end{bmatrix}\ddot{q}+\begin{bmatrix} 0 & 0 & m_t d\dot{\theta}\cos(\theta) \\ 0 & 0 & m_t d\dot{\theta}\sin(\theta) \\ 0 & 0 & 0 \end{bmatrix}\dot{q}=\begin{bmatrix} F_x \\ F_y \\ \tau \end{bmatrix}+\begin{bmatrix} D_x \\ D_y \\ D_\theta \end{bmatrix} \tag{3.64}$$

Let us say the force produced by the right wheel is F_r, by the left wheel F_l and the corresponding force along the x-axis is F_x and y-axis is F_y. The input transformation matrix E(q) depends on the relationship between force and torque and the robot's position (Benchouche, Mellah, & Bennouna, 2021).

$$F_r=\frac{\tau_r}{r},\ and\ F_l=\frac{\tau_l}{r} \tag{3.65}$$

However, force produced by both wheels along the x-axis and y-axis with torque is given as:

$$\begin{bmatrix} F_x \\ F_y \\ \tau \end{bmatrix}=\begin{bmatrix} (F_r+F_l)\cos(\theta) \\ (F_r+F_l)\sin(\theta) \\ (F_r-F_l)L \end{bmatrix}=\frac{1}{r}\begin{bmatrix} (\tau_r+\tau_l)\cos(\theta) \\ (\tau_r+\tau_l)\sin(\theta) \\ (\tau_r-\tau_l)L \end{bmatrix}=\frac{1}{r}\begin{bmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ L & -L \end{bmatrix}\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \tag{3.66}$$

Thus, the input transformation matrix E(q) is obtained as follows.

$$E(q)=\frac{1}{r}\begin{bmatrix}\cos(\theta) & \cos(\theta)\\ \sin(\theta) & \sin(\theta)\\ L & -L\end{bmatrix} \tag{3.67}$$

According to the constraints matrix as in Equation (3.28), force constraint is expressed as:

$$\begin{bmatrix}D_x\\ D_y\\ D_\theta\end{bmatrix}=\begin{bmatrix}m_t(\dot{x}\cos(\theta)+\dot{y}\sin(\theta))\dot{\theta}\sin(\theta)\\ -m_t(\dot{x}\cos(\theta)+\dot{y}\sin(\theta))\dot{\theta}\cos(\theta)\\ m_t(\dot{x}\cos(\theta)+\dot{y}\sin(\theta))d\dot{\theta}\end{bmatrix} \tag{3.68}$$

The following is the matrix representation of the constraint forces:

$$\begin{bmatrix}m_t(\dot{x}\cos(\theta)+\dot{y}\sin(\theta))\dot{\theta}\sin(\theta)\\ -m_t(\dot{x}\cos(\theta)+\dot{y}\sin(\theta))\dot{\theta}\cos(\theta)\\ m_t(\dot{x}\cos(\theta)+\dot{y}\sin(\theta))d\dot{\theta}\end{bmatrix}=A^T(q)\rho \tag{3.69}$$

However, from Equation (3.29), we had $A(q)$ and its transpose $A^T(q)$ and vector of a constraint force (Lagrangian multipliers) ρ in Equation (3.69), is become:

$$A^T(q)=\begin{bmatrix}\sin(\theta)\\ -\cos(\theta)\\ d\end{bmatrix},\quad \rho=m_t\dot{\theta}(\dot{x}\cos(\theta)+\dot{y}\sin(\theta)) \tag{3.70}$$

Thus, the dynamic modeling equation in Equation (3.64) is obtained as:

$$\begin{gathered}\begin{bmatrix}m_t & 0 & m_t d\sin(\theta)\\ 0 & m_t & -m_t d\cos(\theta)\\ m_t d\sin(\theta) & -m_t d\cos(\theta) & I\end{bmatrix}\ddot{q}+\begin{bmatrix}0 & 0 & m_t d\dot{\theta}\cos(\theta)\\ 0 & 0 & m_t d\dot{\theta}\sin(\theta)\\ 0 & 0 & 0\end{bmatrix}\dot{q}\\ =\frac{1}{r}\begin{bmatrix}\cos(\theta) & \cos(\theta)\\ \sin(\theta) & \sin(\theta)\\ L & -L\end{bmatrix}+\begin{bmatrix}\sin(\theta)\\ -\cos(\theta)\\ d\end{bmatrix}\left(m_t\dot{\theta}(\dot{x}\cos(\theta)+\dot{y}\sin(\theta))\right)\end{gathered} \tag{3.71}$$

The dynamic equation can be recast into a suitable state-space representation for control and simulation applications. The transformation can eliminate a constraint term fromthe dynamic equation. Let us define two matrices as having a shift as follows (Benchouche, Mellah, & Bennouna, 2021):

$$w(t)=\begin{bmatrix}v & \omega\end{bmatrix}^T \tag{3.72}$$

$$S(q)=\begin{bmatrix}\cos(\theta) & -d\sin(\theta)\\ \sin(\theta) & d\cos(\theta)\\ 0 & 1\end{bmatrix} \tag{3.73}$$

So, we can rewrite Equation (3.32) as:

$$\dot{q}=S(q)w(t)=\begin{bmatrix}\cos(\theta) & -d\sin(\theta)\\ \sin(\theta) & d\cos(\theta)\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \tag{3.74}$$

By differentiating the Equation (3.74) and using w instead $w(t)$, we have:

$$\ddot{q}=\dot{S}(q)w+S(q)\dot{w} \tag{3.75}$$

Now by substituting Equation (3.75) in (3.35), the following representation will be obtained:

$$H(q)\left[\dot{S}(q)w+S(q)\dot{w}\right]+C(q,\dot{q})\left[S(q)w\right]+F(\dot{q})+G(q)+\tau_d=E(q)\tau-A^T(q)\rho \tag{3.76}$$

By rearranging the above equation, we get:

$$H(q)\dot{S}(q)w+H(q)S(q)\dot{w}+C(q,\dot{q})S(q)w+F(\dot{q})+G(q)+\tau_d=E(q)\tau-A^T(q)\rho \tag{3.77}$$

To remove the constraint matrix $A^T(q)\rho$ multiply Equation (3.77) by $S^T(q)$ as follows:

$$S^T(q)\begin{bmatrix}H(q)\dot{S}(q)w+H(q)S(q)\dot{w}+C(q,\dot{q})S(q)w+F(\dot{q})+G(q)+\tau_d\\ =E(q)\tau-A^T(q)\rho\end{bmatrix} \tag{3.78}$$

From Equation (3.31) we have $S^T(q)A^T(q)=0$. Constraints free dynamic equation is given as:

$$\begin{aligned}&\left[S^T(q)H(q)S(q)\right]\dot{w}+\left[S^T(q)C(q,\dot{q})S(q)+S^T(q)H(q)\dot{S}(q)\right]w\\ &+S^T(q)F(\dot{q})+S^T(q)G(q)+S^T(q)\tau_d=S^T(q)E(q)\tau\end{aligned} \tag{3.79}$$

To rewrite the Equation (3.79), let us define dynamic equations as follows.

$$\bar{H}(q)\dot{w}(t)+\bar{C}(q,\dot{q})w(t)+\bar{F}(\dot{q})+\bar{G}(q)+\bar{\tau}_d=\bar{E}(q)\tau \tag{3.80}$$

where,

$$\bar{H}(q)=S^T(q)H(q)S(q) \tag{3.81}$$

$$\bar{C}(q,\dot{q})=S^T(q)C(q,\dot{q})S(q)+S^T(q)H(q)\dot{S}(q) \tag{3.82}$$

$$\bar{F}(\dot{q}) = S^T(q)F(\dot{q}) = 0 \tag{3.83}$$

$$\bar{G}(q) = S^T(q)G(q) = 0 \tag{3.84}$$

$$\bar{\tau}_d = S^T(q)\tau_d \tag{3.85}$$

$$\bar{E}(q) = S^T(q)E(q) \tag{3.86}$$

$$S^T(q)A^T(q) = 0 \tag{3.87}$$

By ignoring the unknown disturbance vector, Here's the simplified dynamic model

$$\begin{bmatrix} m_t & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \dot{v}(t) \\ \dot{\omega}(t) \end{bmatrix} + \begin{bmatrix} 0 & -m_t d\dot{\theta} \\ m_t d\dot{\theta} & 0 \end{bmatrix}\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = \frac{1}{r}\begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix}\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \tag{3.88}$$

The dynamic model in Equation (3.88) depicts the behavior of a nonholonomic system in new robot or local coordinates; that is, the velocity w in the model of the mobile base's coordinates is transformed to the velocities $\dot{q}$ in cartesian coordinates via the Jacobian matrix $S(q)$. Consequently, for the new set of coordinates, the characteristics of the original dynamics are retained. The dynamics system obtained in Equation (3.80) has the following properties to determine the stability analysis of the control system (Martins & Bertol, 2022):

- ✓ Property 3.3: Limitation - The matrix $\bar{H}(q)$, the norm of a matrix $\bar{C}(q,\dot{q})$, and the vector τ_d is all bounded;
- ✓ Property 3.4: Skew-symmetry - The matrix $\bar{H}(q) - 2\bar{C}(q,\dot{q})$ is skew-symmetric.

Proof: By substituting the time derivative of the $\bar{H}(q)$ and $\bar{C}(q,\dot{q})$ in $\bar{H}(q) - 2\bar{C}(q,\dot{q})$, we get:

$$\dot{\bar{H}}(q) = \dot{S}^T(q)H(q)S(q) + S^T(q)\dot{H}(q)S(q) + S^T(q)H(q)\dot{S}(q), \tag{3.89}$$

$$\begin{aligned} \dot{\bar{H}}(q) - 2\bar{C}(q,\dot{q}) = {} & \dot{S}^T(q)H(q)S(q) + S^T(q)\dot{H}(q)S(q) + S^T(q)H(q)\dot{S}(q) \\ & - 2\left(S^T(q)C(q,\dot{q})S(q) + S^T(q)H(q)\dot{S}(q)\right) \end{aligned} \tag{3.90}$$

$$\begin{aligned} \dot{\bar{H}}(q) - 2\bar{C}(q,\dot{q}) = {} & \dot{S}^T(q)H(q)S(q) - \left[\dot{S}^T(q)H(q)S(q)\right]^T \\ & + S^T(q)\left[\dot{H}(q) - 2C(q.\dot{q})\right]S(q) \end{aligned} \tag{3.91}$$

Since $\dot{H}(q) - 2C(q.\dot{q})$ is skew-symmetry, Equation (3.91) is also skew-symmetry. As a result, the derived system dynamic model is asymptotically stable. The dynamic model in the Equation (3.88) is used in system control and simulation throughout this thesis work.

3.4.4. Controllability Analysis of TWMR

Controllability and stability are the main factors that affect a robotics system's robot selection. We must examine kinematic limitations for a robot to see if they have an impact on the controllability of the system. Nonholonomic constraints restrict a robot's ability to move directly in its principal directions, such as driving straight or rotating. However, by combining simple maneuvers, the robot can achieve any desired configuration.

A system is said to satisfy the rank condition of Lie algebra (e.g., controllability rank condition or rank accessibility requirement) if it has a Lie bracket. The robot is able to reach any point q in the configuration space if the involutive distribution has a rank of n. Chow's theorem states that “the system is controllable if its involutive distribution involving Lie brackets has a rank equal to the configuration space dimension (n).” Some robots can be controlled if their constraints are nonholonomic and their involutive distribution has a rank of n (Klančar et al., 2017). For two vector fields g_1 and g_2 the Lie-bracket is a third vector field g_3 is defined as (Coelho & Nunes, 2003):

$$g_3 = \left[g_1, g_2\right](q) = \frac{\partial g_2}{\partial q} g_1(q) - \frac{\partial g_1}{\partial q} g_2(q) \tag{3.92}$$

where $\left[g_1, g_2\right] = -\left[g_2, g_1\right]$ *and* $\left[g_1, g_2\right] = 0$ for constant vector fields g_1 and g_2.

A nonholonomic constraint represented in Equation (3.25) and matrix $A(q)$ is defined as Equation (3.29); therefore, the system has a three-dimensional configuration space (n=3) and a constraint (p=1) with a velocity vector (m = n–p = 2). The independent field vectors g_1 and g_2 from Jacobian matrix $S(q)$ as in Equation (3.73):

$$g_1 = \left[\cos(\theta) \quad \sin(\theta) \quad 0\right]^T, \ and \ \ g_2 = \left[-d\sin(\theta) \quad d\cos(\theta) \quad 1\right]^T \tag{3.93}$$

By using the Lie-bracket provided in Equation (3.92), we will get the third field vector g_3 ;

$$g_3 = [g_1, g_2] = \frac{\partial g_2}{\partial q} g_1(q) - \frac{\partial g_1}{\partial q} g_2(q) = \begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \\ 0 \end{bmatrix} \tag{3.94}$$

If the system in Equation (3.32) is completely integrable (holonomic), then it is involutive, i.e., $rank(g_1\, g_2) = rank(g_1\, g_2\, [g_1, g_2])$.

$$rank\left\{\begin{bmatrix} \cos(\theta) & -d\sin(\theta) & \sin(\theta) \\ \sin(\theta) & d\cos(\theta) & -\cos(\theta) \\ 0 & 1 & 0 \end{bmatrix}\right\} = 3 \tag{3.95}$$

However, the system is not completely integrable (holonomic) because the rank of the two vector fields, $\left(rank(g_1\, g_2) = 2\right)$, is not equal to the Lie brackets' $\left(rank(g_1\, g_2\, [g_1, g_2]) = 3\right)$. To conclude, the dimension of the generalized configurable vector is equal to the rank of a Lie bracket. i.e., n = 3, which makes the system controllable and nonholonomic according to Chow's theorem. In DWMR, a system's stability is not challenging because the platform has two wheels and one free castor wheel that is used to keep the balance of a system.

3.5. Open-Loop System Modeling Simulation Analysis

To verify whether a system's generated mathematical model fulfills the differential drive mechanism of the mobile robot, an open-loop system simulation is employed. Software such as Matlab and Simulink toolbox are used to simulate the suggested approach. The TWMR and DC motor parameter specification is displayed in Table 3.1. Equations (3.23), (3.34), and (3.88) recreate the kinematics and dynamics model of a DWMR using the motor dynamic model equation. The Appendices contain the Matlab code, functions, and Simulink block diagram of an open-loop system model that were utilized to get the following results. The robot revolves around the arc of the wheel in a differential drive system if the two motors have identical speed magnitudes but distinct signs. If the two motors have identical magnitude speeds with the same +ve sign (forward movement indicated by the +ve sign), then the robot is moving forward. When the motor was given an equal magnitude to the -ve speed symbol, the robot retreated. The robot is moving at a turning point or curvature if the two motor speeds have the same sign but are not equal in magnitude.

Table 3. 1: TMWR and motor parameters specifications (Martins & Bertol, 2022)

Dynamics and Kinematic models Parameters		Actuator Parameters	
Parameters	Values with unit	Parameters	Values with unit
m_t	120 kg	k_m	0.2247Nm/A
L	0.33 m	k_b	0.02 Vs/rad
d	0.10 m	La	0.01 H
r	0.135 m	Ra	6 Ω
I	15.0656 kg-m2	$\tau_{m_{max}}$	20.45 Nm
I_c	11.4180 kg-m2	$V_{s_{max}}$	24 Vdc (maximum motor voltage)
I_w	0.0456 kg-m2	$I_{a_{max}}$	4.0 A (Maximum motor current)
I_m	0.7293 kg-m2		

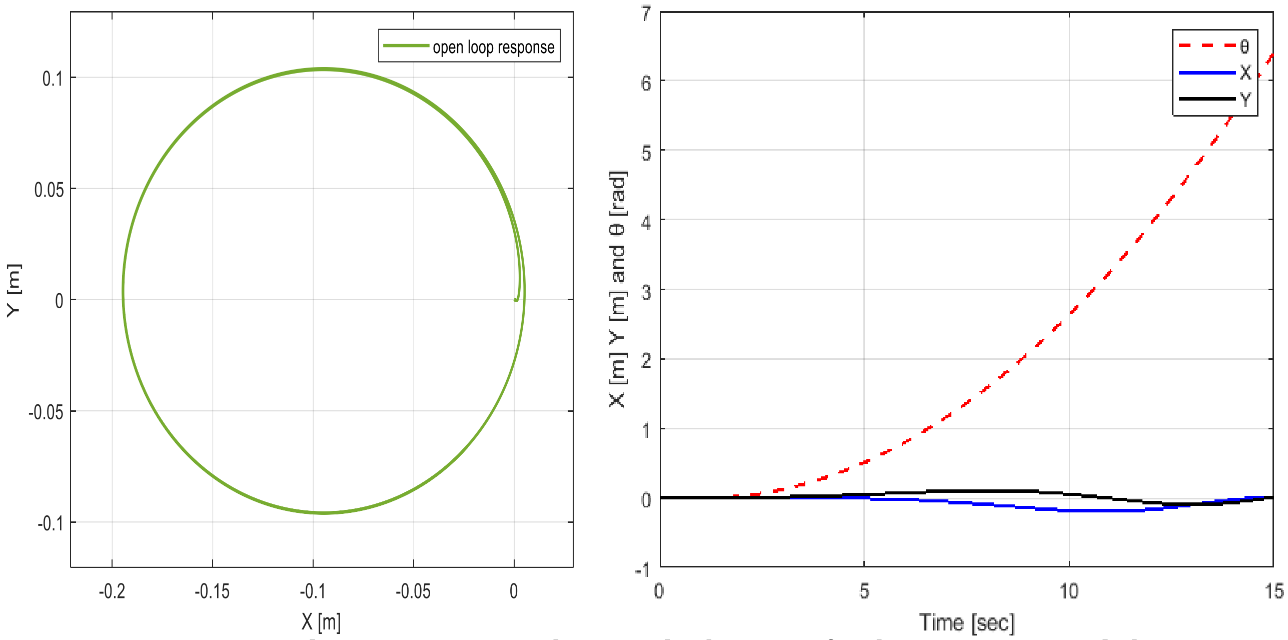

Figure 3.6: Open-loop trajectory tracking, and robot pose for the same magnitude but a different input sign

As seen in Figure 3.6, When the robot's motors operate at equal speeds but with opposite signs, it will rotate around a central point. This rotational motion is caused by the misalignment between the robot's center of mass and its center of rotation. As a result, the robot traces a circular path while revolving around itself. The angles between the robot's reference frame and the local frame will increase during this rotation. which makes the system unstable because the robot revolves around the midpoint of the body frame.

The other test that can be verified the model is by giving the same sign but not equal input to the robot. Figure 3.7 and Figure 3.8 also show an open-loop robot response with different magnitudes and the same sign of the input speed of the motor. As a result, the robot tries to verify the turning point condition during its movement. A robot's posture response is unbounded and unstable for a given input.

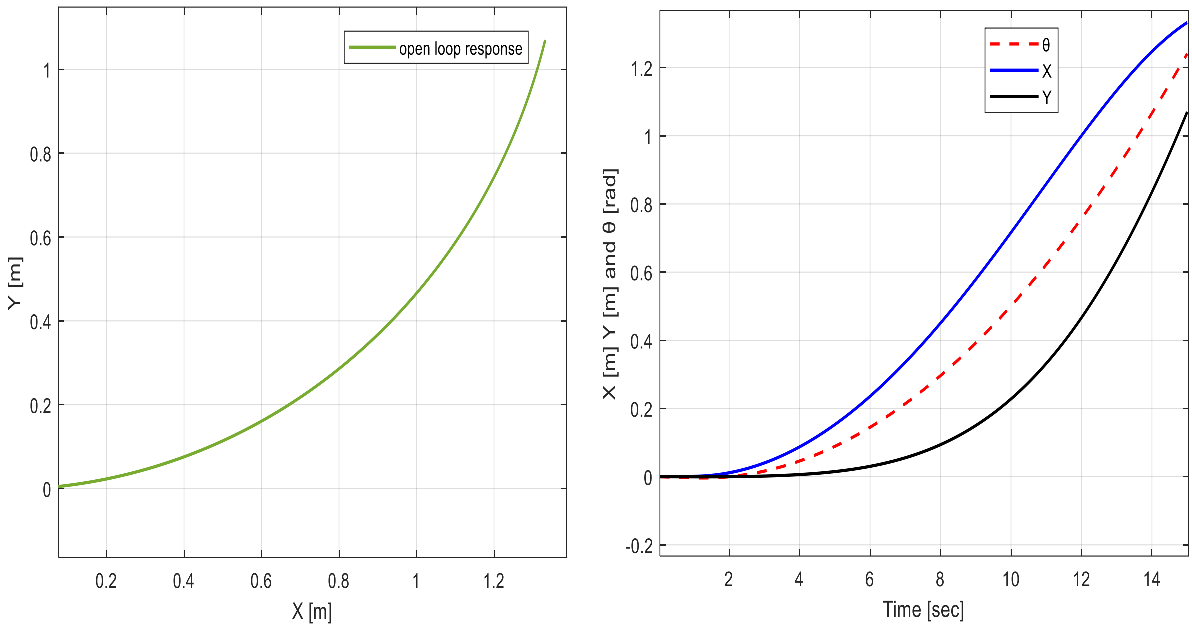

Figure 3.7: Open-loop trajectory tracking, and robot pose for different magnitudes, but the same +ve sign input

The other test employed for the derived system modeling is that a robot moves forward and backward. As seen in Figure 3.9, the robot moves from its initial point to the left and right, respectively. The result shows that a robot's position tracking response is unbounded within a given input.

An open-loop simulation result shows that the derived TWMR system models (kinematic and dynamic) satisfy the differential drive method. An open-loop robot position does not satisfy the desired position tracking because the system response is not bounded or converged in all conditions, which makes it unstable and challenging to control in an interesting region.

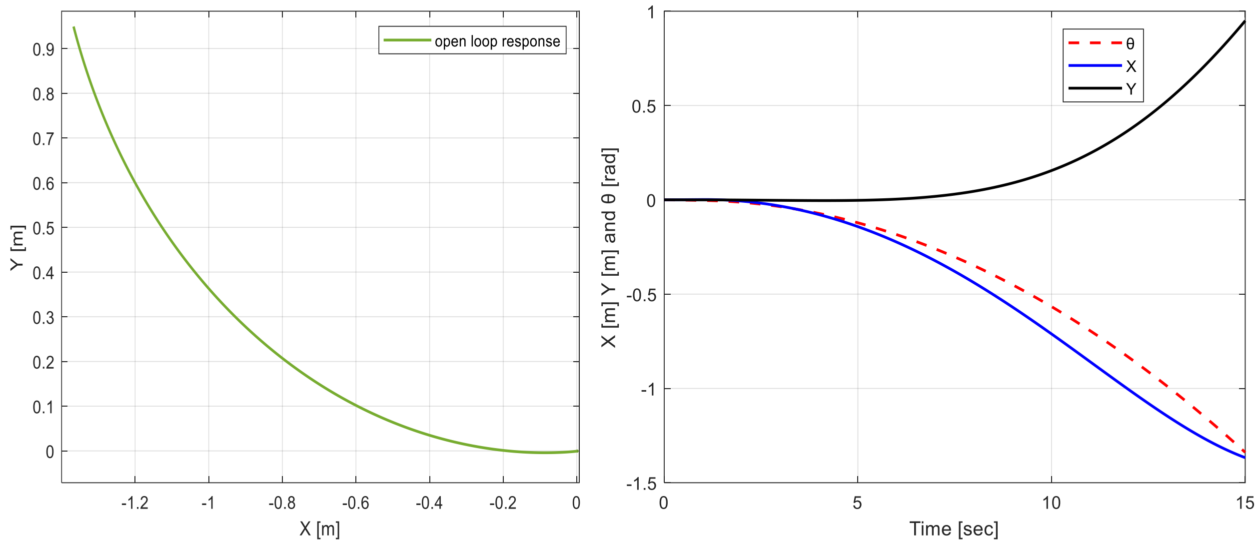

Figure 3.8: Open-loop trajectory tracking, and robot pose for a different magnitude but the same -ve sign input

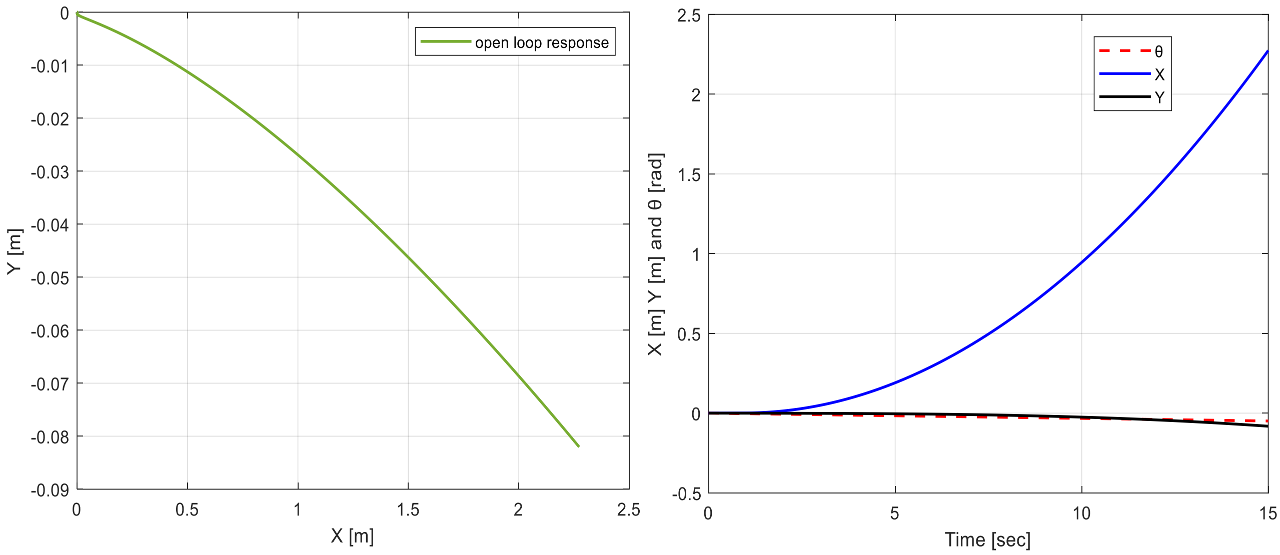

Figure 3.9: Open-loop trajectory tracking, and robot pose for the same magnitude and the same +ve sign input

Generally, the system is highly nonlinear and sensitive to input changes and parameter uncertainty. The control law that solves the aforementioned problem is designed and explained in the following chapter.

CHAPTER FOUR

4. CONTROLLER DESIGN

4.1. Reference Trajectory Generation

The main goal of trajectory tracking of nonholonomic TWMR is to eliminate the tracking error between the desired and actual trajectory. In a mobile robot, reference tracking problems are widely used and researched by numerous researchers because they use a control law to predict the future robot state. Hence, the reference trajectory generation of a TWMR is an essential task for the smooth movement of a robot. A nonholonomic mobile robot trajectory track can be described and formulated as follows (Hwang, Kang, Hyun, & Park, 2013):

$$\begin{cases} \dot{x}_r = v_r \cos(\theta_r) \\ \dot{y}_r = v_r \sin(\theta_r) \\ \dot{\theta}_r = \omega_r \end{cases} \tag{4.1}$$

where x_r and y_r are reference robot positions in the x and y axes, θ_r is the reference angle between the local and the inertial reference frames, v_r and ω_r are reference velocity inputs. From Equation (4.1) the velocity profile has two components: the x-axis and y-axis velocity components. The total tangential (linear) velocity of a robot is:

$$v_r = (v_x(t)^2 + v_y(t)^2)^{\frac{1}{2}} = (\dot{x}_r(t)^2 + \dot{y}_r(t)^2)^{\frac{1}{2}} \tag{4.2}$$

The robot orientation angle θ_r that satisfies a nonholonomic constraint is given as follows:

$$\theta_r = \tan^{-1}(\dot{y}_r(t) / \dot{x}_r(t)) \tag{4.3}$$

By differentiating the Equation (4.3), the reference angular velocity ω_r of a robot is obtained as:

$$\omega_r = \frac{d\theta_r}{dt} = \frac{\ddot{y}_r(t)\dot{x}_r(t) - \ddot{x}_r(t)\dot{y}_r(t)}{(\dot{x}_r(t))^2 + (\dot{y}_r(t))^2} \tag{4.4}$$

Moreover, in a generalized vector, the reference trajectory and velocity are represented by:

$$q_r = [x_r \quad y_r \quad \theta_r]^T \tag{4.5}$$

$$w_r = \begin{bmatrix} v_r & \omega_r \end{bmatrix}^T,\ w_r > 0, \forall t \tag{4.6}$$

For smooth and perfect velocity control, the tracking error between the reference and actual trajectory is near zero (ideally zero). Mathematically, $\lim_{t\to\infty}(q_r - q) = 0$. The following topic introduces a velocity control law as a function of position error, and reference velocity is proposed to reduce trajectory tracking errors.

4.2 Adaptive Fuzzy Sliding Mode Control (AFSMC) Design

In a control system, dealing with a nonlinear system is challenging because in a nonlinear system, the system's state does not change linearly. Many nonlinear controllers have been developed for nonlinear systems control. Two popular nonlinear controllers are the sliding mode controller (SMC) and backstepping controller (BSC). This study uses a sliding mode controller (SMC) controller, which is described in more detail in the following item.

4.2.1. Introduction to Sliding Mode Control Approach

The control structure is purposefully changed to satisfy a criterion or condition, which gives rise to the term sliding mode control (SMC). One straightforward illustration is when the feedback loop switches between various gain values in accordance with a rule that meets the control objectives. The SMC is a high-velocity switching feedback control whose job it is to push the system's trajectory of states—linear or nonlinear—to a predetermined area of the state-space and hold it there for all following times. This area is generally described as a sliding surface, or hypersurface of the system's states. The trajectory of the controlled system's states when restricted to this surface will be determined by the surface's dynamic features and will become less susceptible to unknowns and disruptions. (Martins & Bertol, 2022).

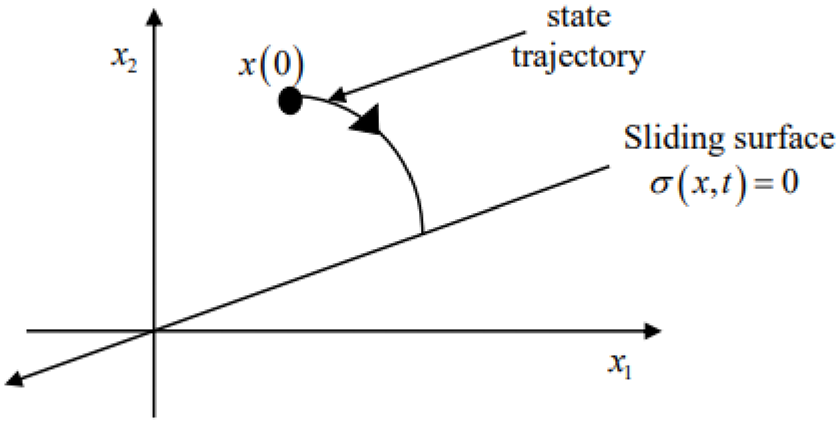

Figure 4.1: State trajectory of a system under the influence of an SMC

The relative position of the sliding surface-related states' trajectories determines how the control structure is altered. A suitable gain value is assigned if the trajectory of the states is above this surface; if it is below it, a different gain is selected so that the trajectory of the states is always oriented towards the sliding surface.

When the system errors, $\tilde{z}$, approach zero, the sliding surface, σ, should be aimed at:

$$\sigma\left(\tilde{z},t\right)=\Lambda^{T}\tilde{z}=0, \tag{4.7}$$

4.2.2 Kinematic-Based Sliding Mode Controller

Finding the linear and angular velocities required for the real DWMR to follow a specific trajectory, as defined by posture coordinates, is the aim of kinematic control. The posture error dynamics of the closed-loop system must be satisfied in order to construct the control synthesis and satisfy the SMC theory.

$$\begin{aligned}\dot{q}_c &= A_0(q_c,t)+B_0(q_c,t)v_c(q_c,t)+d_b(t),\\ \dot{q}_c &= A_0(q_c,t)+B_0(q_c,t)v_c(q_e,t)+B_0(q_c,t)\bar{d}_0(t),\\ \dot{q}_c &= A_0(q_e,t)+B_0(q_c,t)v_c(q_e,t)+B_0(q_c,t)v_c(t).\end{aligned} \tag{4.8}$$

Equation (4.8) is represented in matrix form as:

$$\begin{bmatrix}\dot{x}_c\\ \dot{y}_c\\ \dot{\theta}_c\end{bmatrix}=\begin{bmatrix}u_\varepsilon\cos(\theta_c)\\ u_\varepsilon\sin(\theta_c)\\ \omega_r\end{bmatrix}+\begin{bmatrix}-1 & y_c\\ 0 & -x_c\\ 0 & -1\end{bmatrix}\begin{bmatrix}u_e\\ \omega_e\end{bmatrix}+\begin{bmatrix}-1 & y_c\\ 0 & -x_e\\ 0 & -1\end{bmatrix}\begin{bmatrix}u_e\\ \omega_e\end{bmatrix}. \tag{4.9}$$

To choose σ, it is considered that the DWMR is a sub-actuated system, in which the error dynamics consists of a state vector $q_e\in\mathbb{R}^3$, and must be controlled by the control inputs $v_c\in\mathbb{R}^2$. As the sliding surface synthesis has the aim of converging posture error q_e to zero, the sliding surface σ must have the same dimension as the control velocities v_c, so that their elements σ_1 and σ_2 tends to zero and are associated with v_c e ω_c respectively.

these facts, the choice of σ is defined as:

$$\begin{aligned}\sigma(q_c,I) &= \Lambda^{\mathrm{r}}q_c\\ \begin{bmatrix}\sigma_1\\ \sigma_2\end{bmatrix} &= \begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & \lambda_3\end{bmatrix}\begin{bmatrix}x_c\\ y_c\\ \theta_c\end{bmatrix}\\ &= \begin{bmatrix}\lambda_1 x_c\\ \lambda_2 y_c+\lambda_3\theta_c\end{bmatrix},\end{aligned} \tag{4.10}$$

where λ_1, λ_2, and λ_3 are positive constants chosen by the designer to achieve the desired dynamic behavior in the sliding mode. Consequently, σ^* is given as:

$$\begin{aligned}
\sigma^*(q_e, I) &= \left(\frac{\partial \sigma}{\partial q_c} B_0\right)^2 \sigma(q_c, I) \\
\begin{bmatrix} \sigma_1^* \\ \sigma_2^* \end{bmatrix} &= \left(\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & \lambda_3 \end{bmatrix} \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -1 \end{bmatrix}\right)^x \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} \\
&= \begin{bmatrix} -\lambda_1^2 x_c \\ \lambda_1^2 x_c y_e - (\lambda_2 x_e + \lambda_3)(\lambda_2 y_e + \lambda_3 \theta_e) \end{bmatrix}.
\end{aligned} \tag{4.11}$$

Obtaining the desired sliding motion is determined by the equivalent control method

$$\begin{aligned}
\dot{\sigma}(q_c, I) &= \frac{\partial \sigma}{\partial q_e} \dot{q}_c + \frac{\partial \sigma}{\partial t} = 0 \\
&= \frac{\partial \sigma}{\partial q_e} \left(A_0 + B_0 v_{c_{cec}} + B_0 v_c\right) + \frac{\partial \sigma}{\partial t} = 0.
\end{aligned} \tag{4.12}$$

isolating $\boldsymbol{v}_{\text{c-s}}$, becomes:

$$v_{c_{e4}} = -\left(\frac{\partial \sigma}{\partial q_e} B_0\right)^{-t} \left(\frac{\partial \sigma}{\partial q_c} (A_0 + B_0 v_c)\right) = v_{c_{ces}}^* \tag{4.13}$$

where

$$\frac{\partial \sigma}{\partial I} = 0, \ \frac{\partial \sigma}{\partial q_e} = \Lambda^T. \tag{4.14}$$

After designing the sliding surfaces, Eqs. (4.10) and (4.11), for $\sigma(q_e, t) = \mathbf{0}$ and $\sigma^*(q_e, t) = \mathbf{0}$, one has that:

$$x_c = 0, \ \theta_c = -\frac{\lambda_2}{\lambda_3} y_c, \tag{4.15}$$

and for $\dot{\sigma} = 0$ and $\dot{\sigma}^* = 0$, results in:

$$\dot{x}_c = 0 \ \dot{\theta}_c = -\frac{\lambda_2}{\lambda_3} \dot{y}_e. \tag{4.16}$$

Substituting Eq. (4.13) into Eq. (4.8) and considering Eqs. (4,14) and (4.15), one obtains the desired sliding motion as:

$$\dot{\theta}_c = -\frac{\lambda_2}{\lambda_3} v_r \sin(\theta_c)\,, \tag{4.17}$$

which is identical to Eq. (4.15) with

$$\dot{y}_e = v_x \sin(\theta_e) \tag{4.18}$$

4.2.2.1. Lyapunov Stability Analysis

System stability can be analyzed in different ways in a control system. The Lyapunov function stability analysis is widely used to stabilize a system at an equilibrium point. Any Lyapunov function candidate $V(q)$ must satisfy the following properties (Hadi & Younus, 2020):

1. $V(q)$ this function and its derivatives are continuous.
2. $V(q) = 0,$ for all $q = 0;$
3. $V(q) > 0,$ for all $q \neq 0;$
4. $V(q) < 0,$ for all $q \neq 0;$

Choosing the Lyapunov function candidate in the form:

$$V = \frac{1}{2}\sigma^T \sigma, \tag{4.19}$$

As long as $\lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0, \theta_c \in (-\pi, \pi)$ and $v_r > 0$ In the sliding mode, the surfaces are asymptotically stable. In order to conduct a stability study of the intended sliding motion, the candidate Lyapunov function is defined as follows:

$$V = \frac{1}{2}\theta_e^{\mathrm{T}} \theta_c \tag{4.20}$$

Differentiating Eq. (4.19) and substituting Eq. (4.17), $\dot{V}$ yields to

$$\dot{V} = \theta_c \dot{\theta}_c = -\frac{\lambda_2}{\lambda_3} v_r \theta_c \sin(\theta_c) \leq 0, \tag{4.21}$$

one has $\dot{V} \leq 0$ for $\theta_c \in (-\pi,\, \pi)$ and $v_r > 0$, and $\dot{V} = 0$ if $\theta_c = 0$ and $\theta_c = -\frac{\lambda_1}{\lambda_3} y_c$. When x_c converge to $-\frac{\lambda_2}{\lambda_3} y_c$, the system state y_c will also converge to zero. Therefore, the developed sliding surfaces can have the states $x_c = 0$ and $\theta_c = -\frac{\lambda_2}{x_1} y_c$ in the sliding mode, which are asymptotically stable.

Finding an SMC law is the second stage in the SMC design process after the choice of sliding surfaces and the stability of the intended sliding motion have been determined, which allows $\sigma(q_e, t) \to 0$ and $\sigma^*(q_e, t) \to 0$ when $x_e \to 0$ and $\theta_c \to -\frac{\lambda_2}{\lambda_3} y_c$. Finally, the aim of DWMR posture can be performed by $y_c \to 0$ and $\theta_c \to 0$.

This control rule is determined using stability analysis with the same Lyapunov function candidate provided by Eq. (4.19). This allows one to observe that by changing the sliding surfaces, Eq. (4.10), $V = 0$ for $q_c = 0$ and $V > 0$ for $q_e \neq 0$, i.e., V is positive definite. Differentiating Eq. (4.19) and substituting Eq. (4.13), $\dot{V}$ gives:

$$\dot{V} = \sigma^2 \dot{\sigma} = \sigma^2 \frac{\partial \sigma}{\partial q_c} A_0 + \sigma^2 \frac{\partial \sigma}{\partial q_c} B_0 v_c + \sigma^2 \frac{\partial \sigma}{\partial q_c} B_0 v_c \tag{4.22}$$

Equation (4.11), which represents the new sliding surfaces, along with the relevant non-singular transformation and algebraic operations yield the following results:

$$\dot{V} = {\sigma^*}^2 B_{0_e}^{-1} A_{0_e} + {\sigma^{2}}^2 (v_c + v_c), \tag{4.23}$$

Here is how the control law is defined:

$$\boldsymbol{v}_c = -\boldsymbol{B}_{0_\alpha}^{-1} A_{0_e} - \left(\mathbf{G}\,\mathrm{sign}(\sigma^*) + \mathbf{K} h(\sigma^*)\right), \tag{4.24}$$

with

$$\begin{aligned} A_{0_a} &= \frac{\partial \sigma}{\partial \bar{z}} A_0 = \begin{bmatrix} \lambda_1 v_\varepsilon \cos(\theta_c) \\ \lambda_2 v_r \sin(\theta_c) + \lambda_3 \omega_r \end{bmatrix}. \\ B_{0_a} &= \frac{\partial \sigma}{\partial \bar{z}} B_0 = \begin{bmatrix} -\lambda_1 & \lambda_1 y_c \\ 0 & -\lambda_2 x_c - \lambda_3 \end{bmatrix}. \\ B_{0_\alpha}^{-1} &= \begin{bmatrix} -\dfrac{1}{\lambda_1} & -\dfrac{y_4}{\lambda_2 x_1 + \lambda_1} \\ 0 & -\dfrac{1}{\lambda_2 x_2 + \lambda_2} \end{bmatrix}, \end{aligned} \tag{4.25}$$

and so that $\boldsymbol{B}_{0_\alpha}$ is always nonsingular, the following prerequisite needs to be met:

$$\lambda_2 = \lambda_3 \kappa, \text{ with } 0 \le k \le \frac{1}{|x_e| + 1}. \tag{4.26}$$

Substituting Eq. (4.24) into $\dot{V}$, Eq. (4.23), results in:

$$\dot{V} = -\sigma^{*^2}\left(\mathbf{G}\,\mathrm{sign}(\sigma^*) + \mathbf{K}h(\sigma^*)\right) + \sigma^{*^x}\boldsymbol{v}_{\mathrm{e}-} \tag{4.27}$$

The final control for sliding mode controller

$$\begin{gathered}\boldsymbol{v}_c = -B_{0_e}^{-1}A_{0_e} + v_{c_1}^*,\ v_{\alpha_0}^* = -\mathbf{Q}\,\mathrm{sign}(\sigma^*) - \mathbf{K}h(\sigma^*), \\ \mathbf{G} = \mathbf{Q},\ h(\sigma^*) = \sigma^*.\end{gathered} \tag{4.28}$$

4.2.3 Adaptive Fuzzy Sliding Mode Control Technique

Adaptive Fuzzy Sliding Mode Control (AFSMC) is a hybrid control strategy that combines the robustness of sliding mode control (SMC) with the adaptability of fuzzy logic systems. This technique is particularly effective for handling nonlinear systems with uncertainties and external disturbances.

By adopting an adaptive fuzzy sliding mode control (AFSMC) that takes into account two variations of the parameter adaptation rule, a control solution utilizing fuzzy systems can address the chattering phenomena while improving outcomes without requiring knowledge of the boundaries of uncertainties and disturbances in the system.

4.2.3.1 An Overview of Fuzzy Systems

Four fundamental components make up a fuzzy system, as the block diagram in Figure 4.2 illustrates. The fuzzification process maps a set of non-fuzzy entries from an external system to input fuzzy sets and assesses each entry's degree of membership for each fuzzy set. The rules specify a relationship between the input space and the output space and are stated in the form of "if... then..." Next, in accordance with the relationship specified by the rules, the inference machine maps an input set to an output fuzzy set for each rule. The output fuzzy set is then created by combining the fuzzy sets from each of the rules in the rule base. Ultimately, fuzzy output is converted to a real number for the system through defuzzification.

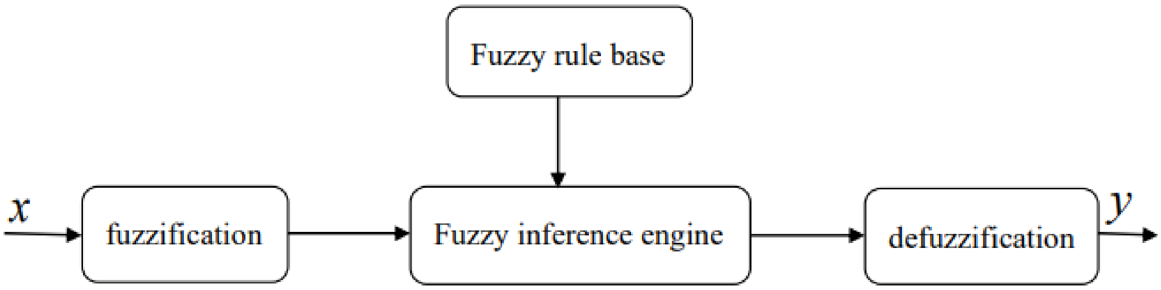

Figure 4.2: An illustration of a fuzzy system using block diagrams

One can express all four components analytically. The ith output of a fuzzy system with n outputs can be expressed as follows in this work by selecting singleton fuzzification, center average defuzzification, the Sugeno implication in the rule base, and the product inference engine:

$$y_i = \frac{\sum_{j=1}^{m} \beta_i^j \mu_i^j(x)}{\sum_{j=1}^{m} \mu_i^j(x)} = \beta_i^T \phi_i(x), \tag{4.29}$$

where x is the input, $\beta_i = \left[\beta_i^1 \dots \beta_i^m\right]^T$ is the vector of consequences, $\phi_i(x) = \left[\phi_i^1(x) \dots \phi_i^m(x)\right]^T$ is the vector of rule weights in which:

$$\phi_i^j(x) = \frac{\mu_i^j(x)}{\sum_{k=1}^{m} \mu_i^k(x)}, \tag{4.30}$$

with $\mu_i^j(x)$ being the membership function of the j^{th} rule of the i^{th} output, and m the number of rules.

The chattering phenomenon in the control presented in is caused by the constant value of G and the discontinuous function sign(σ∗). Let the control gain G sign(σ∗) be replaced by a fuzzy gain $\hat{\boldsymbol{F}}(\sigma^*)$, and therefore, the new control input (AFSMC) is then written as:

$$\begin{aligned} \boldsymbol{v}_{\mathrm{c}} &= -\boldsymbol{B}_{0_0}^{-1}\boldsymbol{A}_{0_0} - \boldsymbol{v}_{\mathrm{c}}^* - \mathbf{K}\boldsymbol{\sigma}^*, \\ \boldsymbol{v}_{\mathrm{c}} &= -\boldsymbol{B}_{0_0}^{-1}\boldsymbol{A}_{0_0} - \hat{\boldsymbol{F}}(\boldsymbol{\sigma}^*) - \mathbf{K}\boldsymbol{\sigma}^*, \end{aligned} \tag{4.31}$$

where v_{c}^* is the control portions of the compensations given by a fuzzy gain defined as $\hat{\boldsymbol{F}}(\sigma^*) = \left[\hat{f}_1(\sigma^*) \dots \hat{f}_{\mathrm{n}}(\sigma^*)\right]^{\mathrm{T}}$, and each $\hat{f}_i(\sigma^*)$ is estimated by an individual fuzzy system that can be written as:

$$\hat{f}_i(\boldsymbol{\sigma}^*) = \frac{\sum_{j=1}^{\mathrm{m}} \hat{\beta}_i^j \mu_i^j(\boldsymbol{\sigma}^*)}{\sum_{j=1}^{\mathrm{m}} \mu_i^j(\boldsymbol{\sigma}^*)} = \hat{\boldsymbol{\beta}}_i^T \phi_i(\boldsymbol{\sigma}^*), \tag{4.32}$$

where $\hat{\boldsymbol{\beta}}_i = [\hat{\beta}_i^1 \quad \dots \quad \hat{\beta}_i^{\mathrm{m}}]^{\mathrm{T}}$ is the vector of online updated consequences, $\phi_i(\boldsymbol{\sigma}^*) = [\phi_i^1(\sigma^*) \dots \phi_i^{\mathrm{m}}(\sigma^*)]^{\mathrm{T}}$ is the vector of rule weights. For an online update of the parameters of the output functions $\hat{f}_i(\sigma^*), \hat{\beta}_i$ was chosen as the parameter to be updated.

Defining $\boldsymbol{\beta}_i$, so that $f_i(\boldsymbol{\sigma}^*) = \boldsymbol{\beta}_i^{\mathrm{T}} \boldsymbol{\phi}_i(\boldsymbol{\sigma}^*)$ is the optimal compensation for $\tilde{\boldsymbol{d}}_{0,}$. According to Wang's theorem [7,13,20] there exists $\psi_i > 0$, satisfying

$$\left|\tilde{d}_{0i} - \beta_i^{\mathrm{T}} \phi_i(\sigma^*)\right| \leq \Psi_i,\ \tilde{d}_0 = v_{\mathrm{e}}(t), \tag{4.33}$$

where ψ_i can be as small as possible. Now, the estimation error and its derivative are defined in the following form:

$$\begin{aligned} \tilde{\boldsymbol{\beta}}_i &= \boldsymbol{\beta}_i - \hat{\boldsymbol{\beta}}_i, \\ \dot{\tilde{\boldsymbol{\beta}}}_i &= -\dot{\hat{\boldsymbol{\beta}}}_i, \end{aligned} \tag{4.34}$$

the Eq. (4.32) can be rewritten as:

$$\begin{aligned} \hat{f}_i(\sigma^*) &= \boldsymbol{\beta}_i^{\mathrm{T}} \phi_i(\boldsymbol{\sigma}^*) - \overline{\boldsymbol{\beta}}_i^{\mathrm{T}} \phi_i(\boldsymbol{\sigma}^*), \\ \hat{\boldsymbol{F}}(\sigma^*) &= \boldsymbol{F}(\sigma^*) - \tilde{\boldsymbol{F}}(\boldsymbol{\sigma}^*), \end{aligned} \tag{4.35}$$

and the adaptation law of the AFSMC can be chosen as:

$$\dot{\hat{\boldsymbol{\beta}}}_i = \sigma_i^* \phi_i(\sigma^*). \tag{4.36}$$

4.2.3.2 Lyapunov Stability Analysis

The Lyapunov function candidate can be used for the following stability study of the AFSMC:

$$V = \frac{1}{2}\left(\sigma^2 \sigma + \sum_{i=1}^{n} \bar{\beta}_i^T \bar{\beta}_i\right), \tag{4.37}$$

where $\bar{\beta}_i^T \bar{\beta}_i > 0$, therefore V is a positive definite.
Differentiating Eq. (4.37), using the definition of Eq. (4.34) and considering the Eqs. (4.13) one obtains:

$$
\begin{aligned}
\dot{V} &= \sigma^{\top}\dot{\sigma} + \sum_{i=1}^{\mathrm{n}} \bar{\beta}_i^{\top}\bar{\beta}_i, \\
\dot{V} &= \sigma^{\mathrm{x}}\frac{\partial\sigma}{\partial q_c}A_0 + \sigma^2\frac{\partial\sigma}{\partial q_c}B_0 v_c + \sigma^x\frac{\partial\sigma}{\partial q_c}d_b - \sum_{i=1}^{\mathrm{n}} \bar{\beta}_i^x\hat{\beta}_i, \\
\dot{V} &= \sigma^{\top}\frac{\partial\sigma}{\partial q_c}A_0 + \sigma^2\frac{\partial\sigma}{\partial q_c}B_0 v_c + \sigma^r\frac{\partial\sigma}{\partial q_c}B_0\bar{d}_0 - \sum_{i=1}^{n} \bar{\beta}_i^i\hat{\beta}_i,
\end{aligned}
\tag{4.38}
$$

considering $\mathbf{I} = \boldsymbol{B}_0\boldsymbol{B}_0^{-1}$,

$$
\begin{aligned}
\dot{V} &= \sigma^{\top}B_0\boldsymbol{B}_{0,-1}^{-1}\boldsymbol{A}_0 + \boldsymbol{\sigma}^{\top}\boldsymbol{B}_{0,}v_c + \boldsymbol{\sigma}^{\top}\boldsymbol{B}_{0,}\bar{\boldsymbol{d}}_0 - \sum_{i=1}^{n} \bar{\boldsymbol{\beta}}_i^{\top}\dot{\boldsymbol{\beta}}_i \\
\dot{V} &= \left(B_{Q_0}^{\tau}\sigma\right)^{\tau} B_{0_c}^{-1}A_{Q_0} + \left(\boldsymbol{B}_{0_c}^{2}\sigma\right)^{2}\left(\boldsymbol{v}_c + \bar{d}_0\right) - \sum_{i=1}^{n} \bar{\beta}_i^{2}\hat{\beta}_i^{2} \\
\dot{V} &= \sigma^{\alpha^x}B_{0_i}^{-1}A_{0_0} + \sigma^{\alpha^x}\left(\boldsymbol{V}_{\mathrm{c}} + \bar{d}_0\right) - \sum_{i=1}^{\mathrm{n}} \bar{\boldsymbol{\beta}}_i^{\mathrm{T}}\dot{\beta}_i -
\end{aligned}
\tag{4.39}
$$

Using the AFSMC, Eq. (4.31), and replacing Eqs. (4.33) and (4.35), $\dot{V}$ results in:

$$
\begin{aligned}
\dot{V} &= \sigma^{*^T}B_0^{-1}A_{0_0} - \sigma^{*^\tau}B_{0_0}^{-1}A_{0_0} - \sigma^{*^T}\hat{\boldsymbol{F}}(\sigma^*) - \sigma^{*^\tau}\mathbf{K}\sigma^* + \sigma^*\bar{d}_0 - \sum_{i=1}^{n} \bar{\beta}_i^{\tau}\dot{\beta}_i \\
\dot{V} &= -\sigma^{*^2}F(\sigma^*) + \sigma^{*^2}\tilde{F}(\sigma^*) - \sigma^{\alpha^2}\mathbf{K}\sigma^* + \sigma^{*^2}v_c - \sum_{i=1}^{\mathrm{n}} \bar{\beta}_i^{\mathrm{T}}\hat{\beta}_i^{2}, \\
\dot{V} &\le -\sigma^{\alpha^2}\mathbf{K}\sigma^* + \sum_{i=1}^{\mathrm{n}} \sigma_i^*\left|v_{c_j} - \boldsymbol{\beta}_i^{\tau}\phi_i(\sigma^*) + \bar{\beta}_i^{\top}\phi_i(\sigma^*)\right| - \sum_{i=1}^{\mathrm{n}} \bar{\beta}_i^{2}\hat{\beta}_i, \\
\dot{V} &\le -\sigma^*\mathbf{K}\sigma^* + \sum_{i=1}^{n} \sigma_i^*\left|v_{c_i} - \beta_i^2\phi_i(\sigma^*)\right| + \sum_{i=1}^{n} \bar{\beta}_i^{2}\left(\sigma_i^*\phi_i(\sigma^*) - \bar{\beta}_i\right).
\end{aligned}
\tag{4.40}
$$

Replacing the adaptation law, Eq. (4.36), $\dot{V}$ yields in:

$$
\begin{aligned}
\dot{V} &\le -\sigma^{*^T}\mathbf{K}\sigma^* + \sum_{i=1} \sigma_i^*\left|v_{c_i} - \beta_i^{\top}\phi_i(\sigma^*)\right| + \sum_{i=1} \bar{\beta}_i^{T}\left(\sigma_i^*\phi_i(\sigma^*) - \sigma_i^*\phi_i(\sigma^*)\right) \\
\dot{V} &\le -\sigma^*\mathbf{K}\sigma^* + \sum_{i=1}^{n} \sigma_i^*\left|v_{c_i} - \beta_i^{\top}\phi_i(\sigma^*)\right|.
\end{aligned}
\tag{4.41}
$$

Thus, the second term on the right side of Eq. (4.41) satisfies the condition of Eq. (4.33), i.e.,

$$\dot{V} \leq -\sigma^{*T}\mathbf{K}\sigma^{*} + \sum_{i=1}^{n} \sigma_i^{*}\mathcal{K}_i. \tag{4.42}$$

The right side of Eq. (4.42.) can be written as

$$\dot{V} \leq -\sigma^{*^3}(\mathbf{K} - \Psi)\sigma^{*} \leq -\lambda_{\text{mix}}\{\mathbf{K}\}|\sigma^{*}|^2 + \bar{\psi}_{\max}|\sigma^{*}| \leq 0, \tag{4.43}$$

ces, $\lambda_{\min}\{\mathbf{K}\}$ is the minimum singular value of $\mathbf{K}$ and $\bar{\psi}_{\max} > |\Psi|$ is a known scalar that represents the maximum effect of the uncertainties and disturbances. Therefore, $\dot{V} \leq 0$ as long as the term in parenthesis in Eq. (7.12) is positive,

$$\dot{V} \leq -|\sigma^{*}|(\lambda_{\min}\{\mathbf{K}\}|\sigma^{*}| - \bar{\psi}_{\max}) \leq 0. \tag{4.44}$$

either

$$|\sigma^{*}| > \frac{\bar{\Psi}_{\max}}{\lambda_{\min}[\mathbf{K}]}. \tag{4.45}$$

This demonstrates that σ^{*} and σ are bounded and the continuity of all functions shows the boundedness as well of $\dot{\sigma}^{*}$ and $\dot{\sigma}$. The demonstration of boundedness $\dot{\beta}$, or equivalently $\bar{\beta}$, is like that of. Moreover, supposing that the vector of rule weights $\phi_i(x)$ is persistently exciting or meets the condition of the persistence of excitation (PE). Then σ^{*} is uniformly ultimate bounded (UUB), with a practical bound given by the right-hand side of the Eq. (4.45), and the estimates of the consequences $\hat{\beta}$ are bounded. Moreover, σ^{*} may be kept as small as desired by increasing the gain $\mathbf{K}$.

In a similar manner, another stability study can also be carried out. In light of this, assuming that Eq. (4.33)

$$\left|v_{e_i} - \beta_i^2\phi_i(\sigma^{*})\right| \leq \psi_i \leq \eta_i|\sigma_i^{*}|, \tag{4.46}$$

where $0 < n_i < 1$. Thereby, the second term on the right side of Eq. (4.41) satisfies the requirement of Eq. (4.46), resulting in:

$$\sigma_i^{*}\left|v_{\sigma_0} - \beta_i^2\phi_i(\sigma^{*})\right| \leq \eta_i|\sigma_i^{*}|^2 = \eta_i\sigma_i^{*^2} \tag{4.47}$$

consequently, using Eq. (4.47), $\dot{V}$ becomes:

$$\dot{V} \leq -\sigma^{\alpha^2}\mathbf{K}\sigma^* + \sum_{i=1}^{n} \eta_i \sigma_i^{\alpha^2}. \tag{4.48}$$

Defining $\Psi = \text{diag}\,[\eta_1 \ldots \eta_{\text{n}}]$ as positive definite matrix and $\bar{\Psi}_{\max} > |\Psi|$, the right side of Eq. (4.48) can be written as

$$\dot{V} \leq -\sigma^*(\mathbf{K} - \Psi)\sigma^* \leq |\sigma^*|^2(-\lambda_{\min}\{\mathbf{K}\} + \bar{\Psi}_{\max}) \leq 0, \tag{4.49}$$

where $\mathbf{K} - \Psi > 0$ or $\lambda_{\min}\{\mathbf{K}] > \bar{\Psi}_{\max}$. Since these conditions must be satisfied, then $\dot{V}$ is negative definite, i.e., $\dot{V} \leq 0$. Moreover, $\dot{V} < 0$ for $\sigma \neq 0, \sigma^* \neq 0$ and, therefore, $q_c \neq 0$ while that $\dot{V} = 0$ only when $\sigma = 0, \sigma^* = 0$ and, consequently, $\boldsymbol{q}_{\text{c}} = 0$, which is an asymptotically stable equilibrium point. Regarding the estimates of the consequences, $\bar{\boldsymbol{\beta}}$, or equivalently $\bar{\boldsymbol{\beta}}$, these remain bounded.

4.2.3.3 Extraction of the Rule Base

In order to determine the fuzzy system rules, V is taken into consideration as stated in Equation (4.19). It is believed that V serves as a gauge for σ energy. the stability of the system is guaranteed by choosing a control law such that $\dot{V} \leq 0$. In the AFSMC, the fuzzy gain $\hat{\boldsymbol{F}}(\sigma^*)$ is applied to compensate uncertainties and disturbances in the system and to reduce the energy of σ^*.

Because of the function sign (σ^*), the control gain has the same signal as σ^*. Therefore, $\hat{f}_i(\sigma^*)$ should have the same signal as σ_i^*. Now consider $\sigma_i^*\left(\tilde{d}_{0_i} - \hat{f}_i(\sigma^*)\right)$. When $|\sigma_i^*|$ is large, it is expected that $\left|\hat{f}_i(\sigma^*)\right|$ is larger so that $\dot{V}$ has a large negative value. This causes the energy of σ^* to decay fast. When $|\sigma_i^*|$ is small, $\sigma_i^*\left(\tilde{d}_{0_i} - \hat{f}_i(\sigma^*)\right)$ is also small and has little effect on the value of $\dot{V}$. Therefore, $\left|\hat{f}_i(\sigma^*)\right|$ can be small

to avoid the chattering phenomenon. When $|\sigma_i^*|$ is zero, $\left|\hat{f}_i(\sigma^*)\right|$ is also zero. Based on this analysis, the rule base is chosen as:

- IF σ_i^* is NB, THEN $\hat{f}_i(\sigma^*)$ is $\hat{\beta}_i{}^2\phi_i{}^1(\sigma^*)$,
- IF σ_i^* is NM, THEN $\hat{f}_i(\sigma^*)$ is $\hat{\beta}_i^2\phi_i^2(\sigma^*)$,
- IF σ_i^* is NS, THEN $\hat{f}_i(\sigma^*)$ is $\hat{\beta}_i^{3^2}\phi_i^3(\sigma^*)$,

- IF σ_i^* is ZE, THEN $\vec{f}_1(\sigma^*)$ is $\vec{\beta}_1^4 4^T \phi_1^4(\sigma^*)$,
- IF σ_i^* is PS, THEN $\hat{f}_1(\sigma^*)$ is $\hat{\beta}_i^{5T} \phi_i^5(\sigma^*)$.
- IF σ_i^* is PM, THEN $\hat{f}_i(\sigma^*)$ is $\vec{\beta}_i^{6^2} \phi_i^6(\sigma^*)$,
- IF σ_i^* is PB, THEN $\hat{f}_i(\sigma^*)$ is $\dot{\beta}_1^{7^T} \phi_i^7(\sigma^*)$.

where ZE is zero, N is negative, P is positive, S is tiny, M is medium, and B is large. The triangular-shaped membership functions were selected to be triangular-shaped functions as:

$$\phi_1^j(\sigma^*) = \begin{cases} 0, & \alpha_i^* < \alpha_1^{i,j} \\ \dfrac{\sigma_i^* - \alpha_1^{i,j}}{\alpha_2^{ij} - \alpha_1^{i,j}}, & \alpha_1^{ij} \leq \sigma_i^* \leq \alpha_2^{i,j} \\ \dfrac{\alpha_3^{2j} - \alpha_i^*}{\alpha_3^{ij} - \alpha_2^{i,j}}, & \alpha_2^{ij} \leq \sigma_i^* \leq \alpha_3^{i,j} \\ 0, & \alpha_3^{i,j} < \sigma_i^* \end{cases} \tag{4.50}$$

where $\alpha_1^{i,j}$, and α_3^{ij} are the "feet" of the triangle, and the parameter $\alpha_2^{\overset{j,j}{2}}$ locates the peak, all for the jrule of the ith output. The parameters of the input membership functions are predefined and given in Table 4.1 as well as can be seen in Figure 4.3, while those of the output, $\hat{\beta}_i^j$, are updated online. Therefore, determining the adaptive characteristic of the controller.

It is important to note that the number of rules that need to be assessed when employing fuzzy logic control determines how much processing time is needed. Large systems with lots of rules would also need very strong and quick processors in order to compute.

Table 4.1: The membership functions' parameters of σ^*

	σ_1^*				σ_2^*		
	j th rule	$\alpha_1^{1,j}$	$\alpha_2^{1,j}$	$\alpha_3^{1,j}$	$\alpha_1^{2,j}$	$\alpha_2^{2,j}$	$\alpha_3^{2,j}$
NB	1	$-\infty$	-0.3	-0.2	$-\infty$	-0.5	-0.33
NM	2	-0.3	-0.2	-0.1	-0.5	-0.33	-0.16
NS	3	-0.2	-0.1	0	-0.33	-0.16	0
ZE	4	-0.1	0	0.1	-40.16	0	0.16
PS	5	0	0.1	0.2	0	0.16	0.33
PM	6	0.1	0.2	0.3	0.16	0.33	0.5
PB	7	0.2	0.3	∞	0.33	0.5	∞

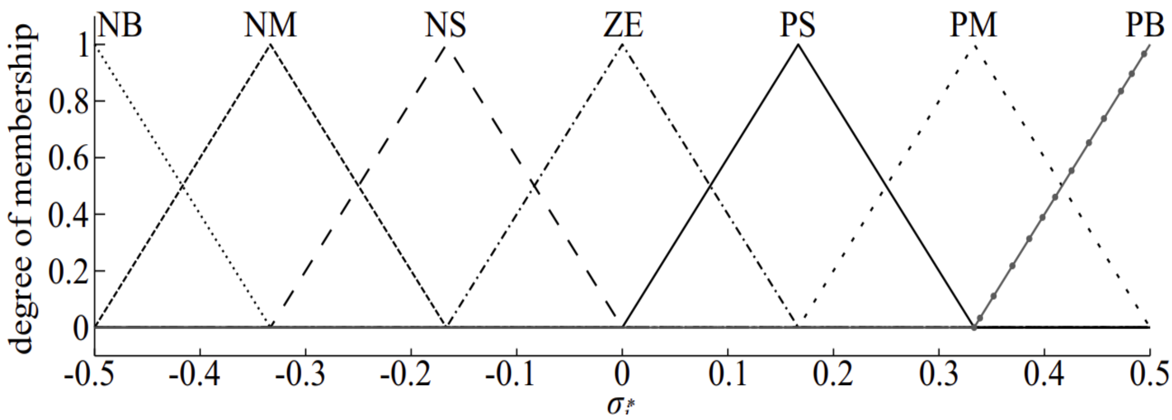

Figure 4.3: Functions for input membership with a triangle shape

in actual time. Less computing power will be required the smaller the rule base. Therefore, in contrast to a pure fuzzy logic controller, which is encountered in the rule expanding problem, the AFSMC employs triangular membership functions, further simplifying their structure, and just 7 if-then rules in the rule base with regard to the sliding surfaces.

4.3. Nonlinear PID Controller

4.3.1. Introduction to PID Controller

The PID controller has been widely used among numerous classical industrial process control controllers since 1930. Because of its simplicity in structure and ease of implementation, PID is widely applicable in closed-loop system control (Zaidner, et al., 2010). As its name describes, a PID controller is a combination of proportional, integral, and derivative functions. It has three control behaviors (Song, 2018).

The proportional controller (P): gives an output proportional to the error signal. The resulting error is multiplied by the proportional gain constant to get the output. However, offset has always existed in this controller.

The Integral controller (I): provides a control action to eliminate the steady-state error over a period of time until the error value reaches zero. This controller improves the limitations of a P-controller. However, overshoot will be the limitation of this I-type controller, and it cannot predict the future state of the error signal. It is used with a combination of P-controllers (PI).

The derivative controller (D): will overcome the I-controller's limitation by anticipating an error's future state. Its output depends on the rate of error change with respect to time, multiplied by a derivation constant.

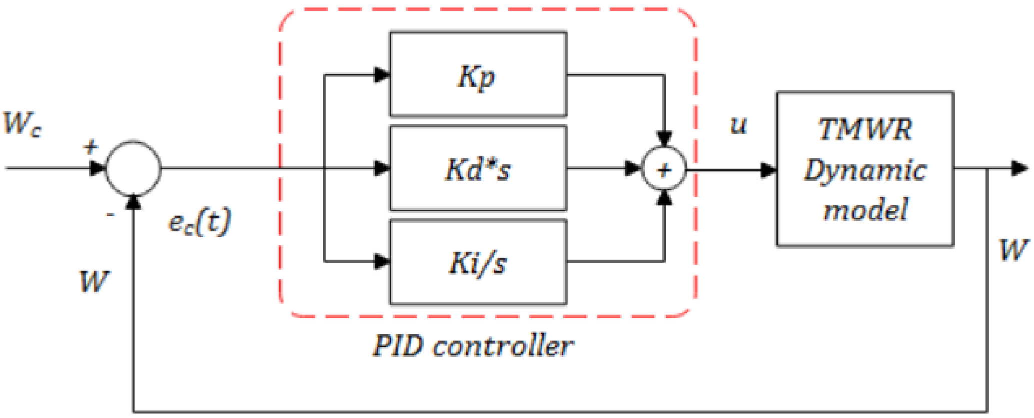

Figure 4.4: Basic PID controller structure with TWMR dynamic model

The time-domain representation of the standard PID controller is given as follows.

$$u(t) = K_p\left[e_c(t) + \frac{1}{T_i}\int_0^t e_c(t)dt + T_d\frac{de_c(t)}{dt}\right] = K_p e_c(t) + K_i\int_0^t e_c(t)dt + K_d\frac{de_c(t)}{dt} \tag{4.51}$$

$$u(s) = K_p + K_i\frac{1}{s} + K_d s \tag{4.52}$$

where u is PID controller output, K_p is the proportional gain, T_i is the integral time, T_d is the derivative time, $K_{i=}K_p / T_i$ is the integral gain, $K_{d=}K_pT_d$ is the derivative gain in the case of a parallel PID controller. An error signal $e_c(t)$ is given as the difference between the control and actual velocity values $(e_c(t) = w_c - w)$. The integral of the error function is accumulated to achieve better tracking performance. A derivative predicts the rate of change of an error to prevent an overshoot. For TWMR velocity control, a PID controller is designed as follows:

$$u_k = K_{pk} + K_{ik}\frac{1}{s} + K_{dk}s,\ \ where\, k = 1,2 \tag{4.52}$$

where $u_k = [u_1, u_2]^T = [u_v, u_\omega]^T$ controller output, and k is the number of controllers.

4.3.2. Nonlinear PID Controller for TWMR Velocity Control

It is suggested to use an NPID controller with an architecture resembling that of a traditional PID controller. While the integral action has a nonlinear function, the proportional and derivative actions are linear. Consequently, the error input to the integral action is scaled by a nonlinear gain function in the product of the error and the nonlinear gain. A work-related NPID controller is comparable to a work-related NPID controller. (JIN & SON, 2019). A proposed nonlinear PID controller block diagram is shown in Figure 4.5.

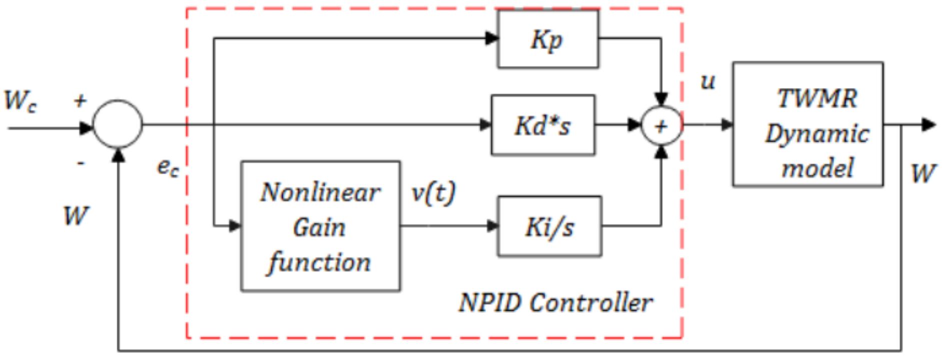

Figure 4.5: Nonlinear PID controller with TWMR dynamic model

The NPID controller's time-domain equation for TMWR velocity control is as follows:

$$
\begin{aligned}
u_k(t) &= K_{pk}\left[e_c(t)+\frac{1}{T_{ik}}\int_0^t v(t)dt + T_{dk}\frac{de_c(t)}{dt}\right] \\
&= K_{pk}e_c(t)+K_{ik}\int_0^t v(t)dt + K_{dk}\frac{de_c(t)}{dt}
\end{aligned}
\tag{4.53}
$$

The nonlinear scaled error function $v(t)$ in the integral and derivative control action is given as (Garbaabaa, et al., 2020):

$$
v(t)=k(e)e_c(t),\ where, k(e)=\exp\left(-\frac{e_c(t)^2}{2\Delta w_c^2}\right) \tag{4.54}
$$

where $u_k=[u_1,u_2]^T=[u_v,u_\omega]^T$ is controller output, $k(e)$ is a nonlinear gain function, $\Delta w_c \neq 0$ is controlled velocity change, i.e., $\Delta w_c = w_c(i)-w_c(i-1)$ and i denote the discrete instant of time. According to (JIN & SON, 2019), a typical value of Δw_c is 0.5, 1, and 1.5. A nonlinear PID controller output in Equation (4.53) with a nonlinear gain function as in Equation (4.54) is used throughout this thesis. A controller's gain parameters are adjusted using genetic algorithm optimization techniques to get an optimal value for better performance tracking.

4.4. Overall Proposed System Modeling

The WMR system model derived in Chapter 3 is a nonlinear system. A nonlinear system needs an appropriate control mechanism to control and stabilize at an equilibrium point. As an adaptive fuzzy sliding mode control rule that stabilizes a nonholonomic system from inner to outer subsystem was designed. The auxiliary velocity control input is used to stabilize the internal subsystem at the equilibrium point $(e_c = 0)$ as $t \to \infty$ and step back out to the external subsystem at the equilibrium point $(e_p = 0)$ as $t \to \infty$. This could be achieved by using a Lyapunov stability function.

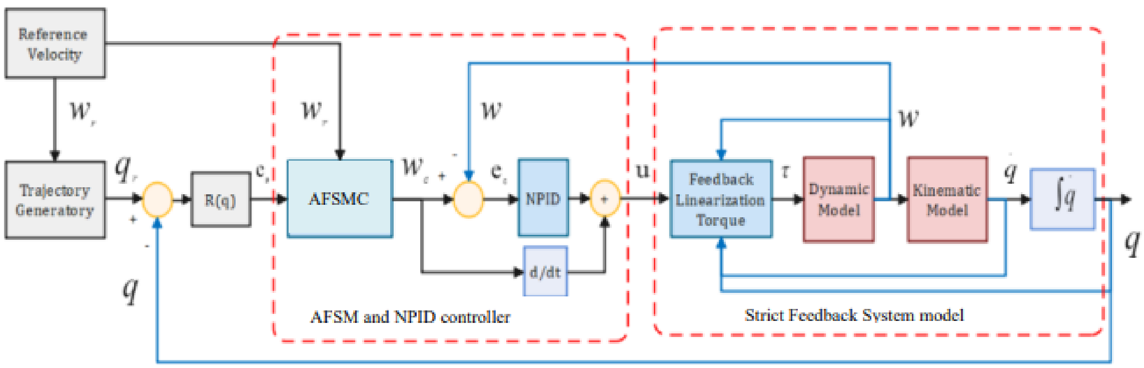

Figure 4.6: Overall system model with AFSM and NPID controller

The overall system modeling using system dynamics as in Equations (3.34) and (3.88) with a proposed control rule in Equation (4.46) combined with an NPID controller as in Equation (4.53) will be designed as shown in Figure 4.6.

The trajectory tracking performance of a sliding mode controller is compared with that of an adaptive fuzzy sliding mode controller. All controller gains are tuned using a genetic algorithm to get optimal values for better trajectory tracking with the possible minimum posture and velocity errors.

4.5. Controller Gains Optimization by Using a Genetic Algorithm

4.5.1. Introduction to Genetic Algorithm (GA)

In this paper, GA is considered to have an optimal linear and nonlinear gain tuner because it is simple to use and understand, and it has almost identical characteristics to that of PSO. As GA optimization is used to tune the controller parameters, one of the fundamental things to define first is the objective function to be minimized. Before thinking about how to tune a controller, we must first determine what constitutes a good response. There are a variety of measures that can be used to compare the quality of controlled responses. (Messom, 2002).

Start

Set t = 0;

Generate initial population $P(t)$;

Compute the fitness of an individual in $P(t)$;

do while < Stop condition for not satisfied >

Set t = t+1;

Select from individual $P(t-1)$ to set a tentative population $P(t)$;

Perform Cross-over individual in $P(t)$;

Perform Mutation of an individual in $P(t)$;

Compute the fitness of an individual in a new population $P(t)$;

end while

Output the best individual $P(t)$ as the best solution;

Stop

Figure 4.7: Pseudo-code of the genetic algorithm

The starting population, fitness function, selection, crossover, and mutation phases make up the genetic algorithm. The GA method process' pseudo-code is displayed in Figure 4.7. The process starts with a population, which is a group of people.

The ability to compete with others is determined by an individual's level of fitness, which is determined by the fitness function. The most physically fit people are selected, and selection passes on their genes to the following generation. Every time a pair of parent's mates, a gene's crossover locations are chosen at random, and mutation takes place to preserve population variety and avoid premature convergence. If the population has converged, meaning it is not producing significantly different offspring from the previous generation, the process stops. After then, it is claimed that the GA has offered a fix for our issue. The fittest people will eventually emerge from this iterative process, which continues generation after generation. The most physically fit person is thought to be the answer to an issue. (Alam, Qamar, Dixit, & Benaida, 2021).

There is no analytical proof that the solution generated in a nature-inspired optimization tool solver is globally optimal or not. The probabilistic random functions provide the foundation of the searching mechanism. However, we took into account the unique behavior of the solvers in order to choose an optimization tool for a suitable system. A few behaviors we should take into account are the data handling capability, the type of cost function, the program's complexity, and the convergence rate. As a result, code is simpler, GA optimization is simply comprehended, and it can handle a multi-objective function. A popular optimization tool solver for mixed or continuous/discrete systems is this one.

4.5.2. Objective Function

By using a possible minimum objective function, the GA optimization optimizes a controller to produce optimal settings. By applying integral error criteria, such as integral square error (ISE), integral absolute error (IAE), or integral time-weighted absolute error (ITAE), the goal function can be reached. The time function of the objective/cost function using integral error criteria is as follows:

$$ISE = \int_0^{t_f} e^2(t)dt,\ IAE = \int_0^{t_f} |e(t)|\,dt,\ and\ \ ITAE = \int_0^{t_f} t\,|e(t)|\,dt \tag{4.55}$$

where $e(t)$ represents the difference between the desired and actual values, and t_f is the time duration. Because it is easily analyzed, ISE is frequently utilized for optimum control. IAE has more sensitivity than ISE because it uses the absolute magnitude of the error. However, ITAE penalizes an error that persists long, resulting in more significant discrimination than IAE or ISE (JIN & SON, 2019).

A TMWR is a system with a multi-input and multi-output (MIMO). The cost function used to minimize is a sum of individual cost functions. That is, the cost function is rewritten as (Al-Mayyahi, Wang, & Birch, 2016);

$$ITAE = \int_0^{t_f} \left(t\left|e_p(t)\right| + t\left|e_c(t)\right| \right) dt \tag{4.56}$$

where $e_p = R(\theta)[q_r - q]$ is pose error and $e_c = w_c - w$ is a velocity error. This cost function should be reduced to minimum values for better tracking performance response.

4.5.3. Parameter Settings of the Genetic Algorithm

The following parameters are used in Matlab/Simulink software simulation to obtain adaptive fuzzy sliding mode, SMC, and NPID controller gains parameters. A kinematic-based SMC has five gains $\left(\lambda_1,\, \lambda_2,\, \lambda_3,\, Q_1,\, and\ Q_2\right)$ optimized to reduce a tracking error to the minimum possible value. Similarly, NPID controllers have three parameters $\left(K_p,\, K_i,\, and\ K_d\right)$. However, the dynamic model has two controlled variables: angular velocity and linear velocity.

In GA optimization techniques, the optimal solution to a problem is obtained depending on a parameter we set during simulation. The best solution to a problem is obtained if more populations, a maximum number of generations, and more sampling time are selected. However, it takes a long time to finish the simulation. If a small population and a few generations are set, a solution to a problem is quickly found, but the solution is less accurate. Therefore, the selection of a number of populations, generations, and lap time is based on the performance needed from the system.

Table 4.1 shows a breakdown of the parameters used in the gain optimization simulations for the SMC plus NPID controllers. Genetic algorithm optimization is executed in Matlab using Matlab-code or an optimization toolbox app. A Matlab code and Simulink function used for system simulation and optimization are provided in Appendices A and B. It is important to note that the overall system is optimized simultaneously for better performance and to overcome the effects of one loop on the other.

Table 4. 2: Optimization parameters for the genetic algorithm

Parameters of the Algorithm	
Maximum generation	30
Population size	30
No of parameters	11
Lower Boundary (LB)	[0 0 0 0 0 0 0 0 0 0 0]
Upper Boundary (UB)	[100 1 1 100 1 1 10 10 10 10 10]

CHAPTER FIVE

5. RESULTS AND DISCUSSION

5.1. Introduction

The mathematical modeling of TWMR with a proposed control algorithm is discussed and explained in the previous section. The Adaptive fuzzy sliding mode controller combined with an NPID controller is designed for TWMR trajectory tracking control. The proposed controller is simulated on Matlab/Simulink software, as shown in Figure 5.1. The proposed control method performance and the capability to track a given trajectory by changing the initial position.

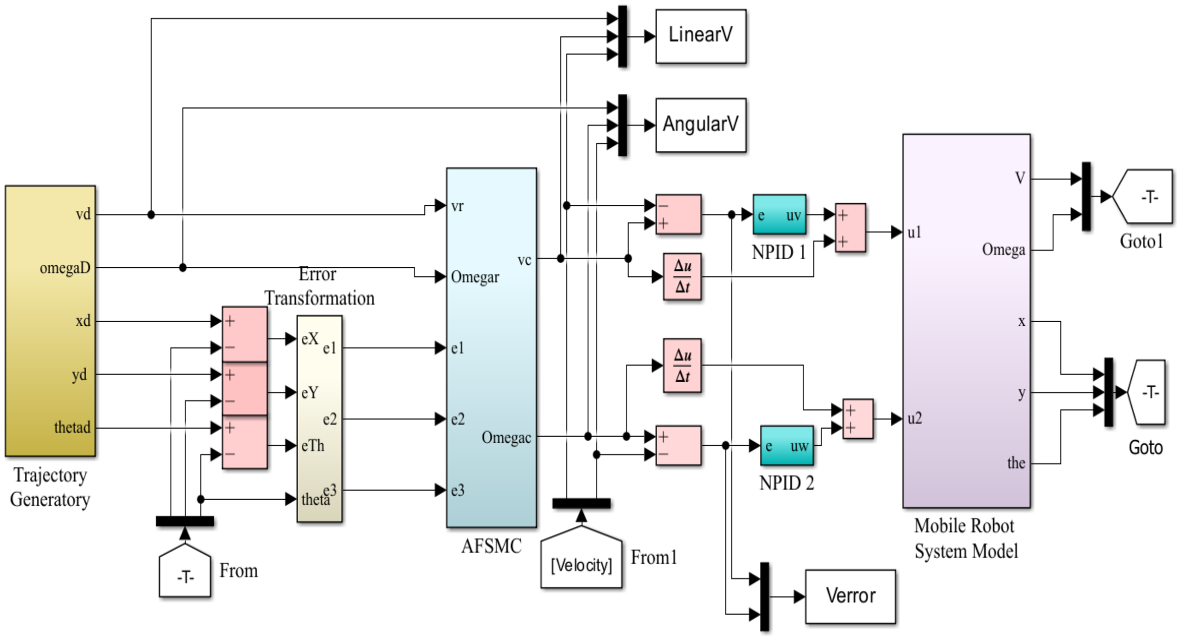

Figure 5.1: Matlab/Simulink system simulation block diagram

For a simple understanding, the designed system in Figure 5.1 is described as follows:

- The dark green color denotes a trajectory generator,
- The light-blue color denotes the proposed controllers (AFSMC and NPID controller),
- The yellow color denotes a TWMR system model.

5.2. Simulation Results

This thesis work optimizes controller gains by the genetic algorithm. The simulation is done with a solver with a fixed step size of 0.001 sec and a simulation stop time of 5 sec. The solver type is a fourth-order differential equation (Runge-Kutta). As shown in Table 5.1 and Table 5.2, the best values of controller gains.

Table 5. 1: Parameters Sliding Mode and NPID controller gains optimized by GA

Sliding Mode and NPID Controller gain parameters										
SMC					NPID 1			NPID 2		
λ_1	λ_2	λ_3	Q_1	Q_2	Kp1	Ki1	Kd1	Kp2	Ki2	Kd2
1.5	6	1	0.1	0.3	71.488	0.044	0.948	99.807	0.025	0.18

Table 5. 2: Adaptive Fuzzy Sliding Mode and NPID controller gains optimized by GA

Adaptive Fuzzy Sliding Mode and NPID Controller gain parameters										
AFSMC					NPID 1			NPID 2		
λ_1	λ_2	λ_3	K_1	K_2	Kp1	Ki1	Kd1	Kp2	Ki2	Kd2
1.5	6	1	0.1	0.3	48.12	1.598	9.177	96.315	0.019	1.137

5.2.1. Trajectory Tracking of TMWR using AFSMC

Two scenarios for the reference trajectory are taken into consideration in order to evaluate a controller's adaptability—its ability to monitor a given trajectory.

A linear reference trajectory (straight shape) is applied to a motion and is given as:

$$\begin{cases} x_r = t \\ y_r = 2 * t \end{cases}, \; for\ all\ t \geq 0 \tag{5.1}$$

Furthermore, a nonlinear reference trajectory (eight-shape) is given as:

$$\begin{cases} x_r = -5 * \sin(t/5) \\ y_r = 10 * \sin(t/10) \end{cases}, \; for\ all\ t \geq 0 \tag{5.2}$$

The reference trajectory and velocity are generated according to Equation (4.1) to Equation (4.4), derived in the previous section.

5.2.1.1. Tracking Response of a Linear Reference Trajectory (Straight-line)

Robots that use linear reference trajectory tracking move quickly along a predetermined path because the controller can predict how a tracking line or tracking error will behave in the future. The Root-Mean Square (RMS) value of the tracking error is used to compute the performance of a robot in its numerical value.

Figure 5.2 and Figure 5.3 show that a robot tracks a given reference trajectory with a minimum tracking error at $q0$ = [0, 0, 0] T initial states between 0 and 10 sec. In a given time range, a robot tracking performance response in the x-direction position has a short settling time of 0.18 sec and a peak value of 0.0112 m with a peak time of 0.029 sec, as in Figure 5.2.

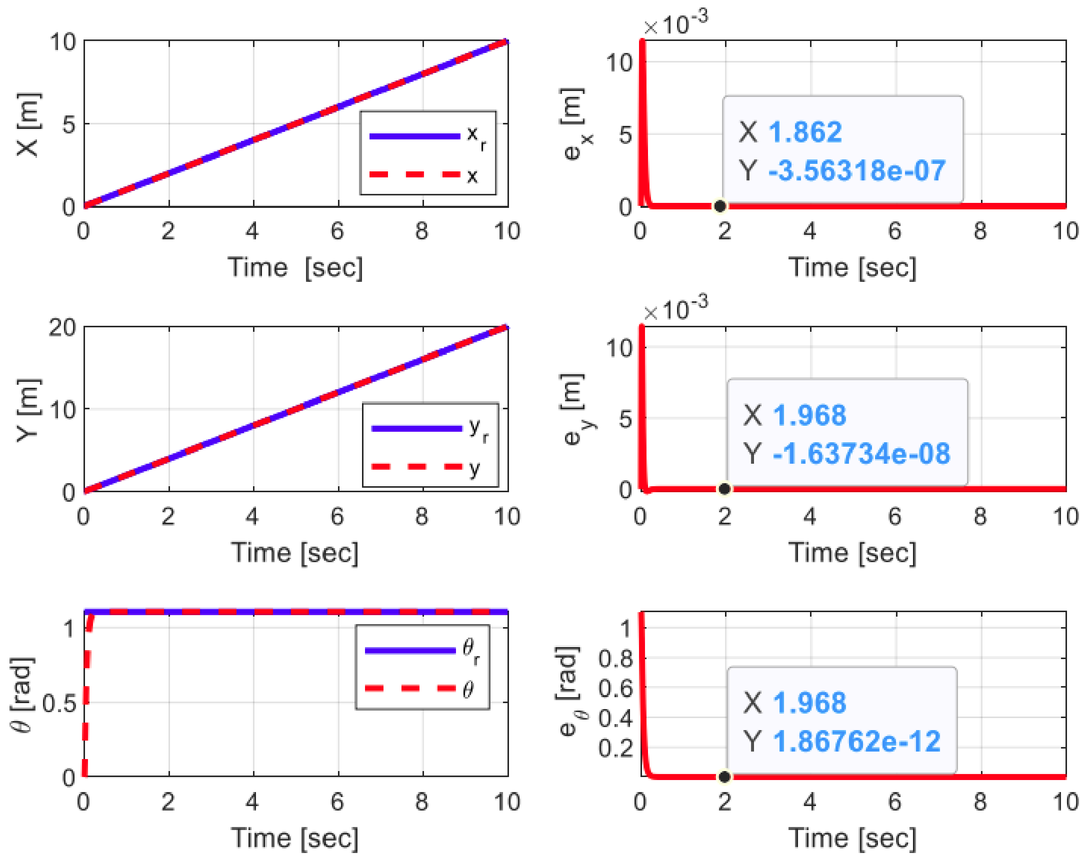

Figure 5.2: x, y, and θ variables tracking response and tracking error

In addition, the y-axis tracking error has a 0.1 sec settling time with a peak value of 0.0112 m at 0.013 sec. Regarding its heading angle, the robot needs 0.18 sec to reach a reference trajectory. On the other hand, the robot's position error (trajectory tracking error) is calculated from position tracking errors in the x and y-direction. Mathematically, expressed as $\left(e_x^2 + e_y^2\right)^{0.5}$.

As shown in Table 5.3, the RMS value of tracking errors achieved by a proposed controller in the case of linear trajectory tracking is minimum with [0, 0, 0] initial robot state.

Table 5. 3: RMS values of linear trajectory tracking error with [0, 0, 0] initial state

Tracking errors	$e_x(m)$	$e_y(m)$	$e_\theta(rad)$	$e_{pos}(m)$
RMS value	0.0007	0.0006	0.059	0.0009

The result clearly shows that a robot needs only about 0.18 sec to track a reference trajectory when it starts its movement from an [0, 0, 0] initial position, as shown by the tracking error in Figure 5.3. The trajectory tracking error has only a 0.015 m peak value at 0.018 sec. Therefore, the robot tracking performance achieved by a proposed controller with [0, 0, 0] initial states is almost perfect, regardless of the peak value at the start of a movement, which converges to zero quickly.

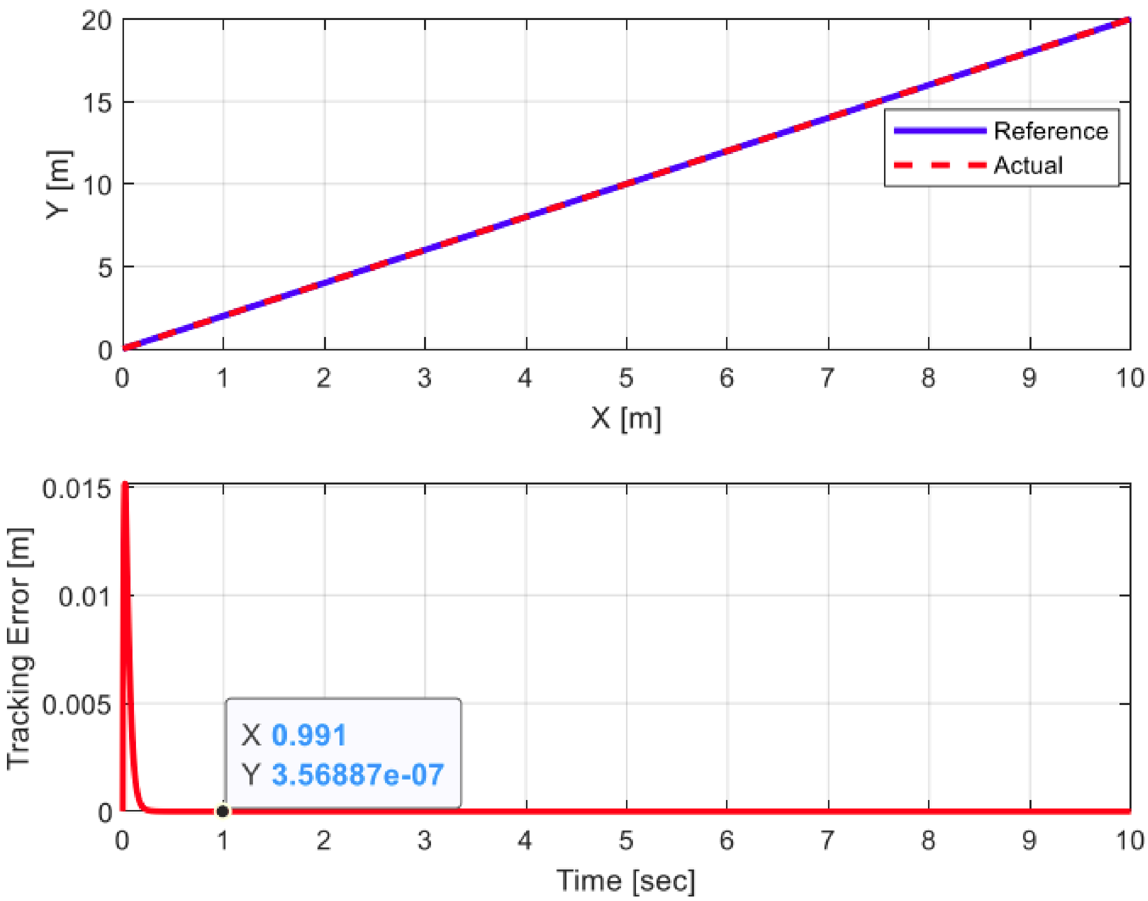

Figure 5.3: Linear trajectory tracking and position error

However, the AFSMC controller is that it produces a significant time response when the robot's initial position is changed. In this case, the robot runs at high speed. In order to test the capability of the robot to track a given reference trajectory with a proposed method, changing the initial state and applying an unknown disturbance to the system is essential.

Trajectory tracking performance with [2, 1, pi/2] initial robot position

The trajectory tracking response is shown in Figure 5.4, and Figure 5.5 shows that the robot follows its reference trajectory with [2, 1, pi/2] initial position change. From Figure 5.4, the robot has a considerable peak value at the start of motion; then, this value is decreased to zero in a short time. Thus, the proposed method rejects the effect of initial position changes on a robot's performance with a short settling time and a slight overshoot.

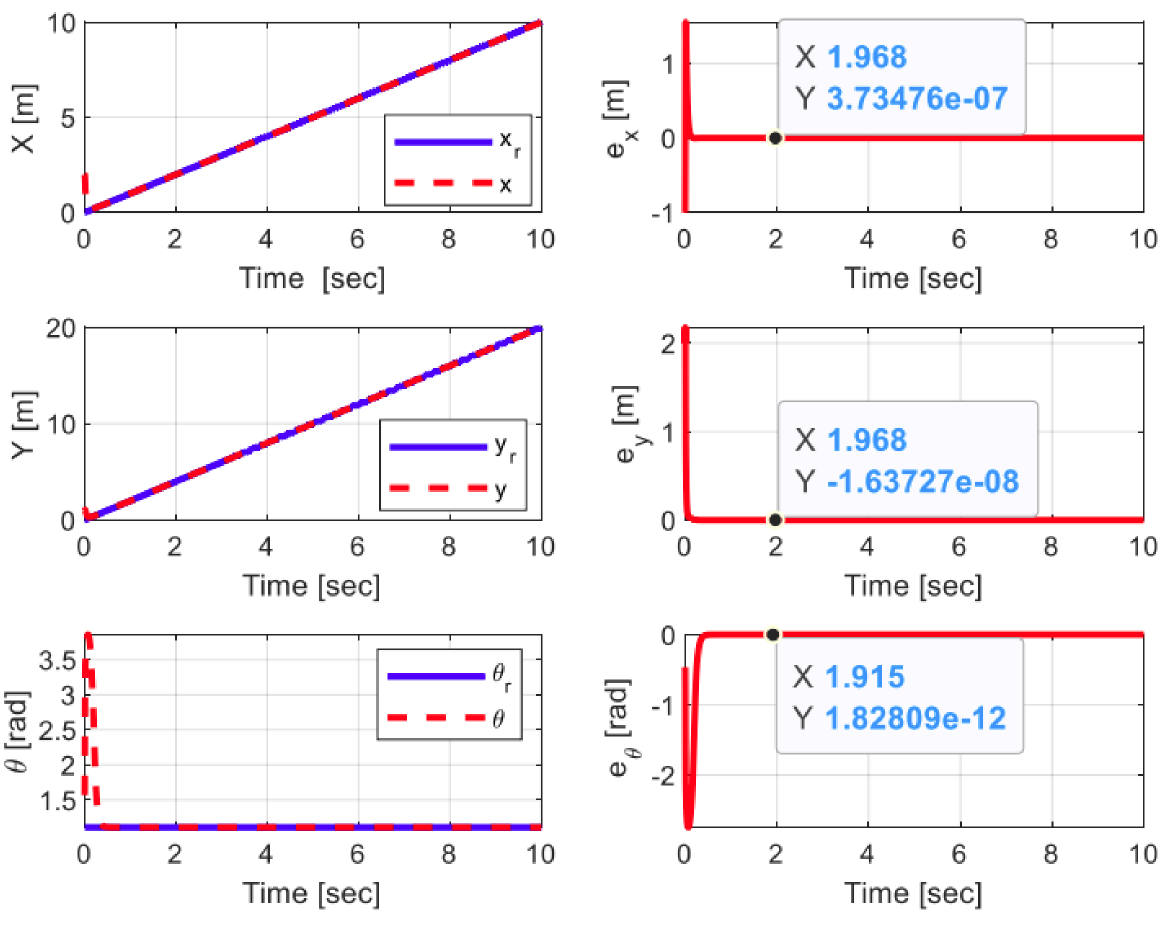

Figure 5.4: x, y, and θ variables tracking response and tracking error with the initial position of [2, 1, pi/2]

Figure 5.5 shows the tracking performance of a robot with a proposed control method when it starts its movement from [2, 1, pi/2] at the initial position. As we have seen, the obtained results are a smooth response and converge quickly to their steady-state error. The trajectory tracking error of the robot initially has a considerable peak value due to the initial position change. However, this error is reduced to almost zero in a short time.

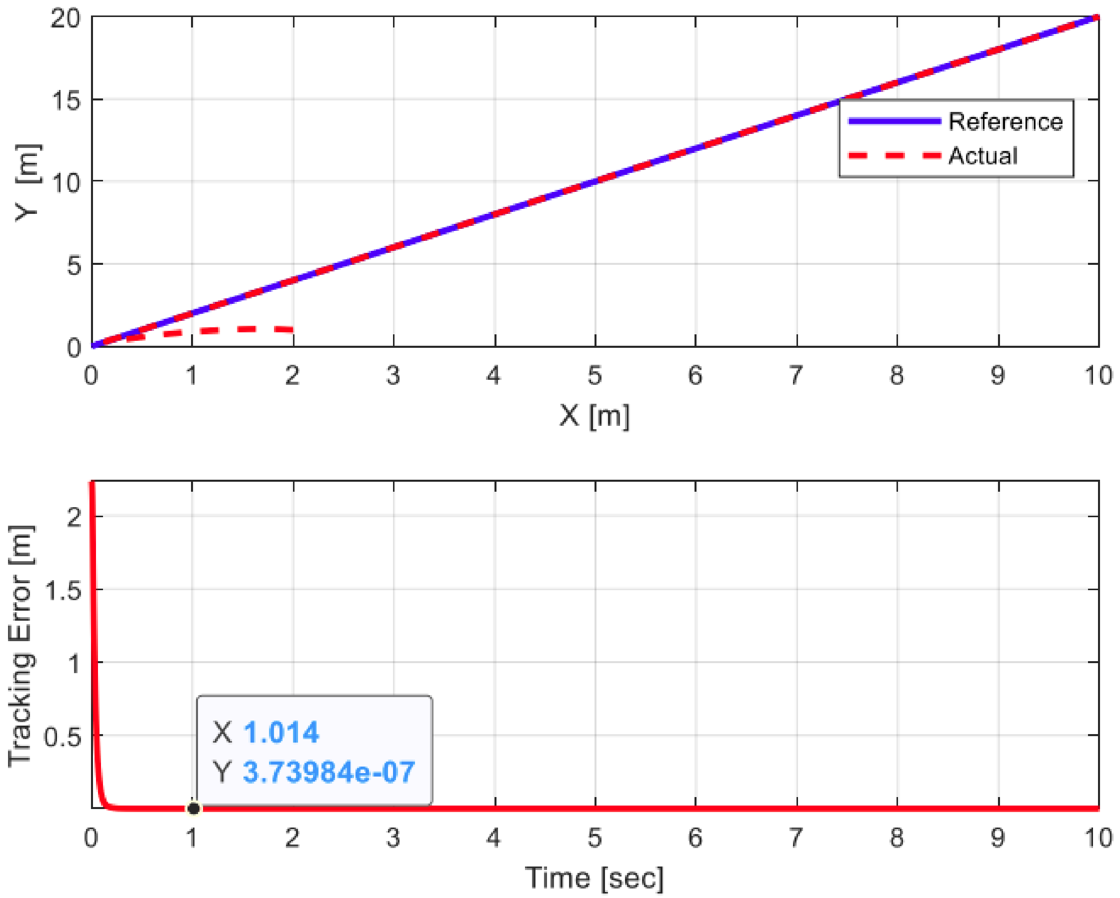

Figure 5.5: Linear trajectory tracking and tracking error with [2,1, pi/2] initial position

As a result, as shown in Table 5.4, the proposed controller has a small RMS value of tracking error, and its trajectory tracking response is also preferable and acceptable.

Table 5. 4: RMS values of linear trajectory tracking error with [2, 1, pi/2] as an initial state

Tracking errors RMS value	$e_x(m)$	$e_y(m)$	$e_\theta(rad)$	$e_{pos}(m)$
	0.0746	0.0694	0.3436	0.1018

5.2.1.2. Tracking Response of a Nonlinear Reference Trajectory (Eight-Shape)

For the second scenario, a nonlinear trajectory (eight-shape) is considered as a reference trajectory input. Since the objective of the AFSMC controller is to stabilize a robot at an equilibrium point, the robot's performance at the initial state [0, 0, 0] is shown in Figure5.6 to Figure5.10. In this case, the reference robot angle is a time-dependent function, not a constant.

As shown in Figure5.6, the robot trajectory tracking of a sinusoidal reference trajectory in the x-direction is almost perfect control with a tracking error peak value of 0.007 m at 0.018 sec of peak time regardless of undershoot at the start of the movement.

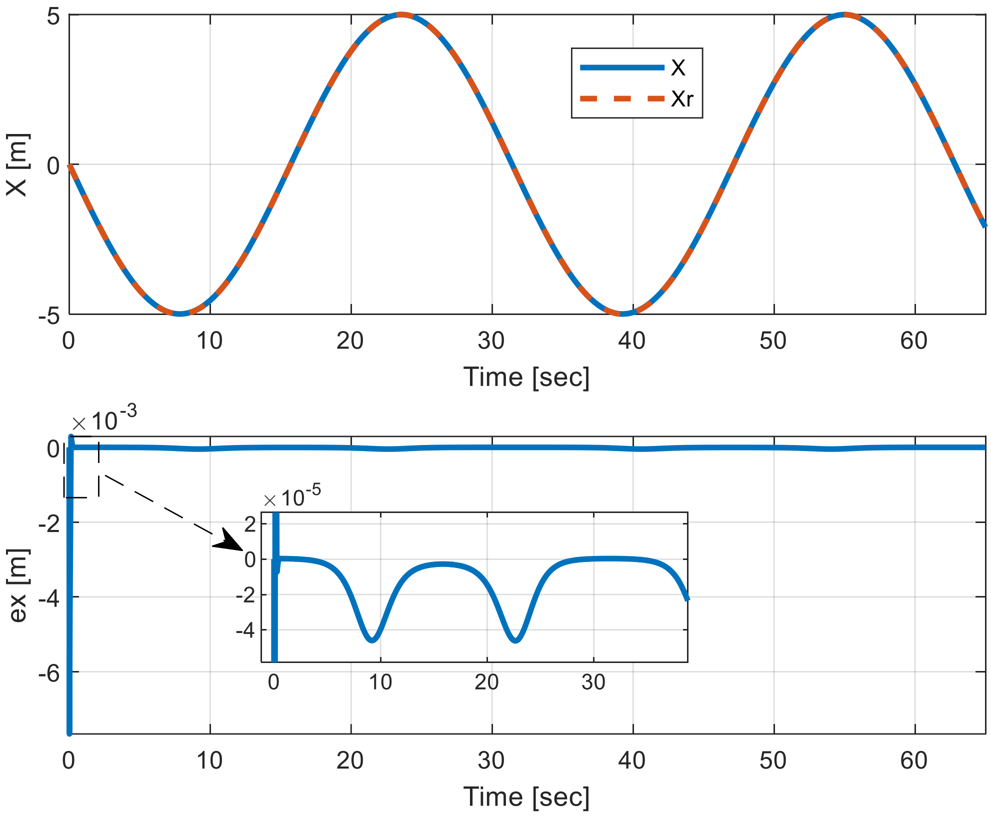

Figure 5.6: x-position tracking response and tracking error with [0, 0, 0] initial position

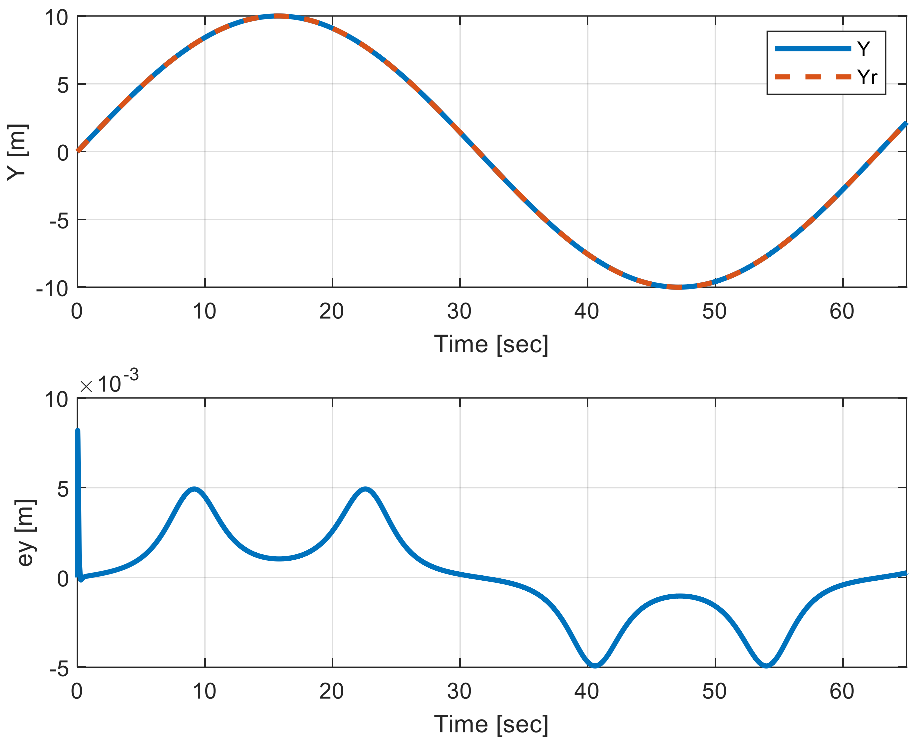

Figure 5.7: y-position tracking response and tracking error with [0, 0, 0] an initial position

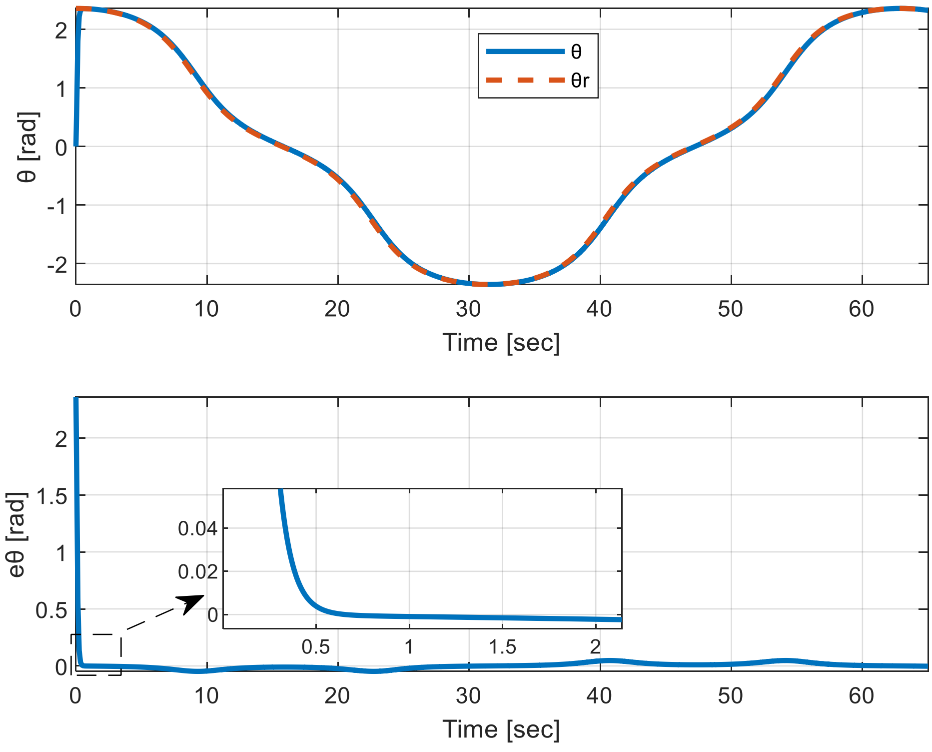

Figure 5.8: Angle tracking response and tracking error with the initial position [0, 0, 0]

As shown in Figure 5.8, since the robot's heading angle depends on position changes in y-direction and x-direction, the reference steering angle at the start of motion is not zero. The robot's steering angle quickly converges to zero. Moreover, the result in Table 5.6 shows that a robot tracks a given trajectory with small RMS tracking error values.

Table 5. 5: RMS values of the eight-shape trajectory tracking error with [0, 0, 0] initial state

Tracking errors	$e_x(m)$	$e_y(m)$	$e_\theta(rad)$	$e_{pos}(m)$
RMS value	0.0002	0.0025	0.0799	0.0025

When the robot begins its movement at the same initial point as the reference trajectory, it tracks its reference trajectory smoothly in the x and y positions with the least amount of tracking error feasible. From the trajectory tracking response shown in Figure 5.9 and the tracking error in Figure 5.10, the robot tracks an eight-shape trajectory with an error peak value of 0.015 m at a 0.018 sec peak time. According to the demonstrated results, the robot's trajectory tracking performance with a [0, 0, 0] initial state is preferable with a minimum tracking error.

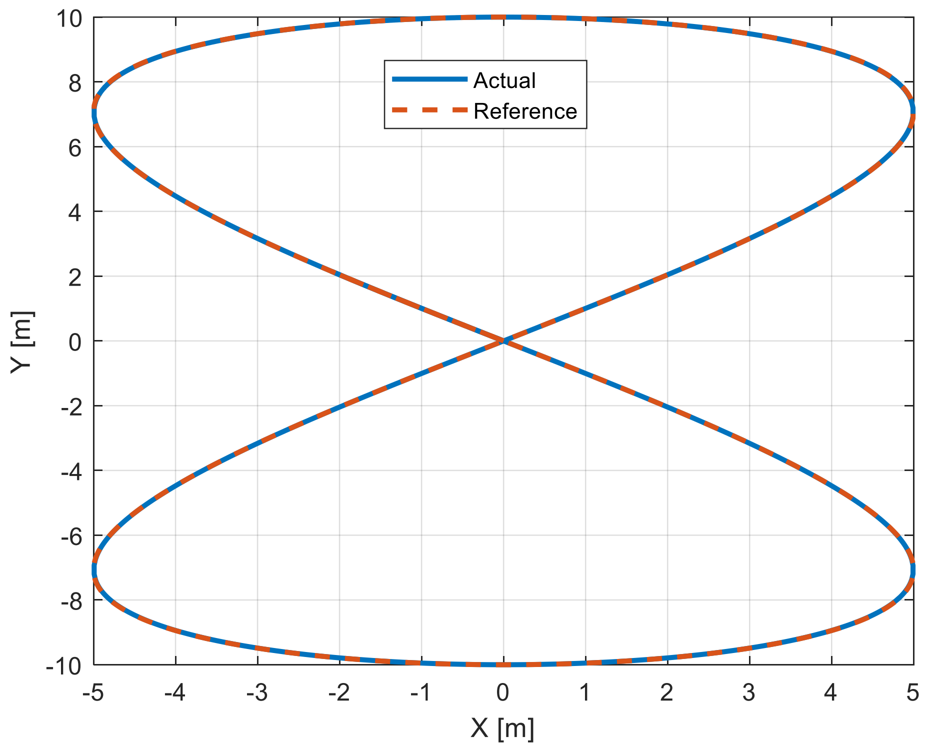

Figure 5.9: Eight-shape trajectory tracking response with initial position [0, 0, 0]

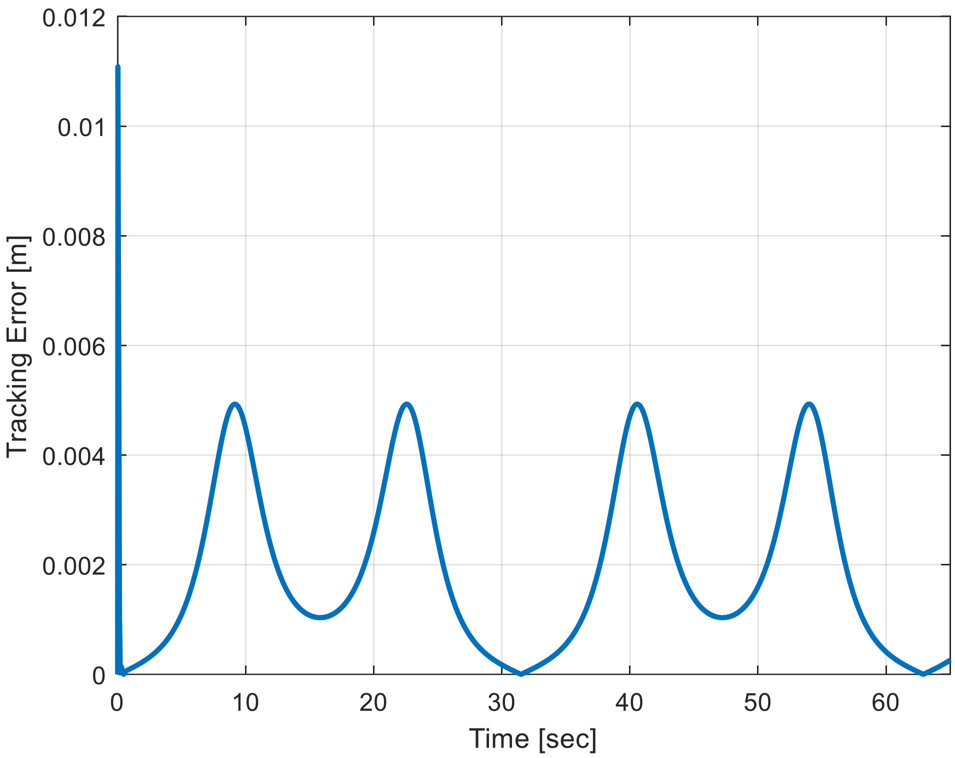

Figure 5.10: Eight-shape trajectory tracking error with the initial position [0,0,0]

In the trajectory tracking problem, if the initial position is changed, it imposes more overshoot on the system's transient response. If the change is much farther from the initial position of the reference trajectory, the system may become unable to track a given path, making it unstable and challenging to control in a reference position.

Trajectory tracking performance with [1, 2, pi/2] initial robot position

Figure 5.11 to Figure 5.15 show the trajectory tracking with the initial position [1, 2, pi/2]. The results show that the robot follows its reference trajectory with a possible minimum tracking error and a settling time of 0.1205 sec, as shown in Table 5.7.

Table 5.6: RMS values of trajectory tracking error with an initial state [1, 2, pi/2]

Tracking errors RMS value	$e_x(m)$	$e_y(m)$	$e_\theta(rad)$	$e_{pos}(m)$
	0.0439	0.0372	0.0792	0.0596

The obtained results revealed that the x position, y position, and robot angle tracking errors converged in a short settling time regardless of the significant error at the start of a motion due to the initial position change. i.e., the robot needs more energy to track a target path at the beginning and then return to its position.

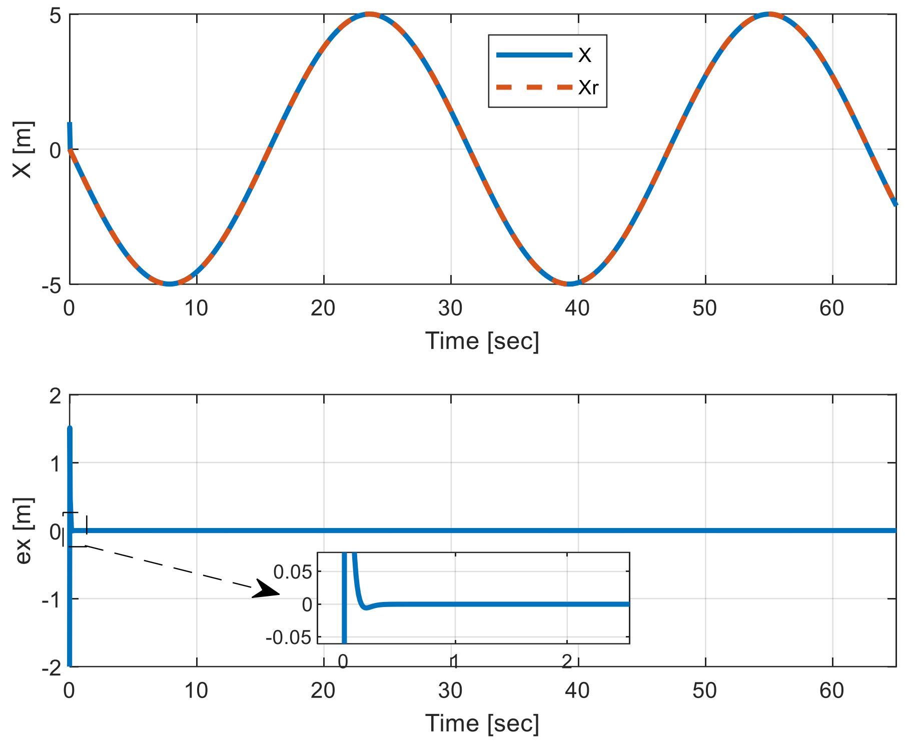

Figure 5.11: x position tracking and tracking error with initial position [1, 2, pi/2]

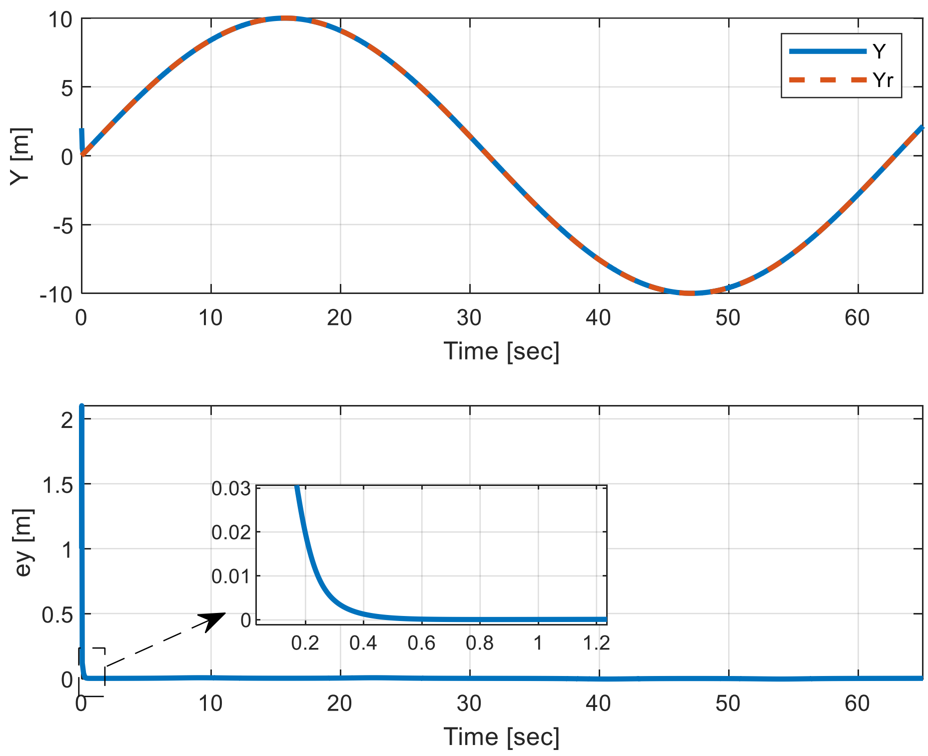

Figure 5.12: y position tracking and tracking error with initial position [1, 2, pi/2]

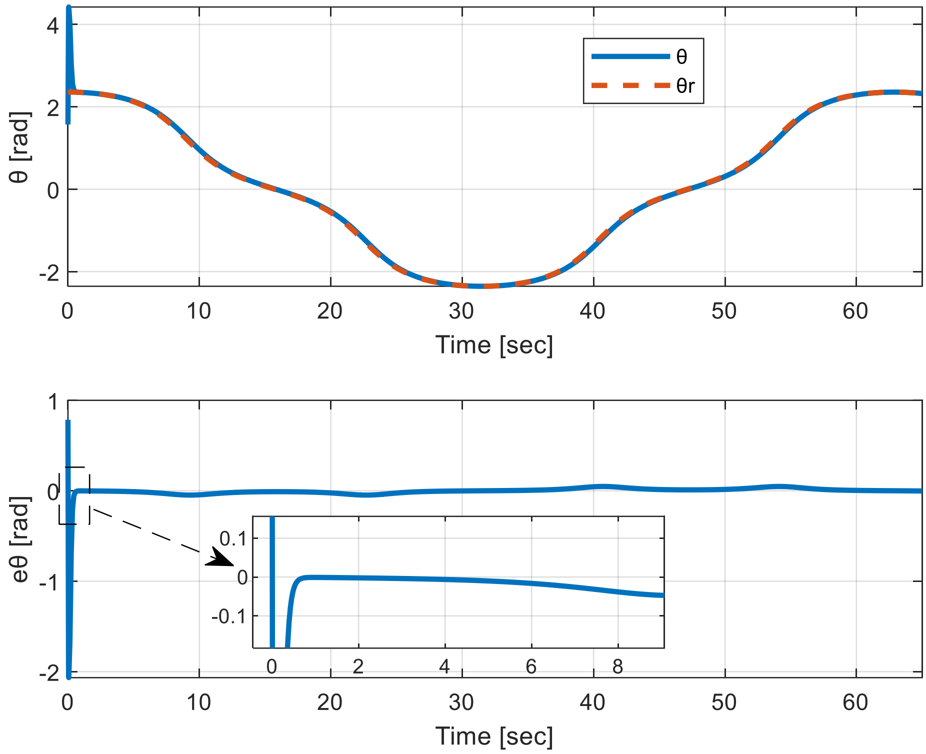

Figure 5.13: Angle tracking and tracking error with the initial position [1, 2, pi/2]

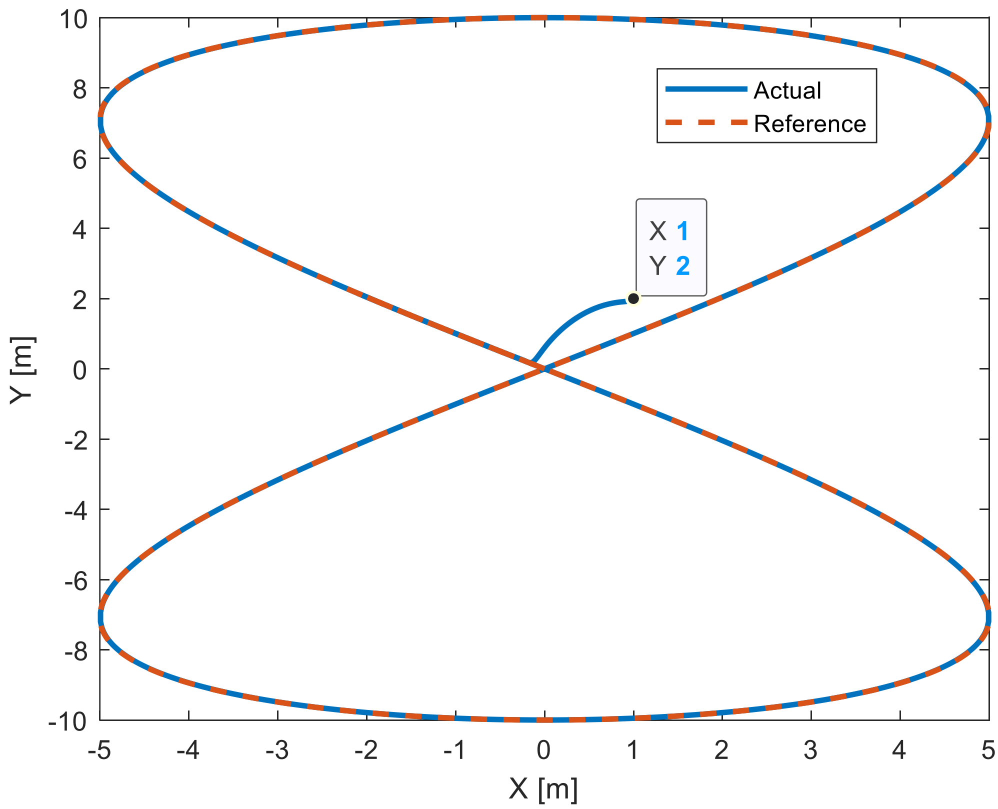

Figure 5.14: Eight-shape trajectory tracking response with the initial position [1, 2, pi/2]

In addition, the demonstrated results show that a robot tracks its reference trajectory with a minimum tracking error within an initial state change, as shown in Figure 5.15. The results show that the proposed controller is the fastest to respond to initial position changes.

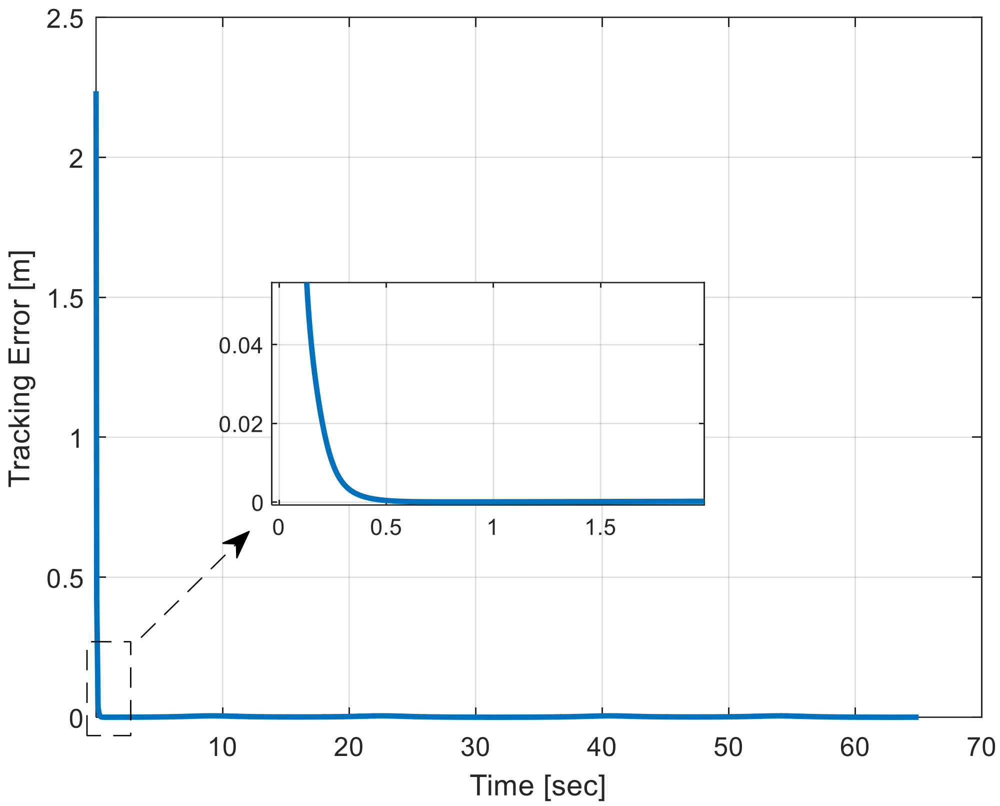

Figure 5.15: Robot trajectory tracking error with [1, 2, pi/2] initial state

Generally, the trajectory tracking of the TWMR was achieved using AFSMC+NPID controllers, as demonstrated. The results obtained by a proposed controller improve the trajectory tracking response's transient behavior and the tracking error's RMS value within a specified time. Consequently, the trajectory tracking of a TWMR is controlled and stabilized with an initial position change.

CHAPTER SIX

6. CONCLUSION AND RECOMMENDATION

6.1. Conclusion

This thesis introduces an adaptive fuzzy sliding mode controller (AFSMC) designed for precise trajectory tracking of a three-wheeled mobile robot. The proposed controller effectively handles the complexities arising from the robot's nonlinear dynamics, uncertainties in its model, and external disturbances. A versatile adaptive fuzzy logic system is integrated to accurately estimate the unknown nonlinear dynamics, while the robust sliding mode control ensures reliable tracking performance.

The simulation results show that the AFSMC performs better than the traditional sliding mode control. The suggested controller maintains stability, tracks trajectories accurately, and is resilient to unknowns and disturbances. The controller's performance is further improved by the fuzzy logic system's adaptive nature, which allows it to adjust to changing operating conditions.

The AFSMC offers significant advantages over conventional sliding mode control, including improved trajectory tracking accuracy, enhanced robustness to disturbances, and greater adaptability to varying conditions. These benefits are clearly evident in the simulation results.

6.2. Recommendation

To further validate and refine the proposed AFSMC, it is recommended to conduct comprehensive experiments under various operating conditions to assess its robustness, adaptability, and performance in the presence of different levels of noise, disturbances, and uncertainties. Additionally, comparing the performance of the AFSMC with other state-of-the-art controllers for three-wheeled mobile robots will provide valuable insights into its relative advantages and disadvantages.

Moreover, investigating methods to improve the energy efficiency of the AFSMC is crucial, especially for applications where battery life is a critical factor. Techniques such as adaptive switching between different control modes or optimizing the control parameters can be explored to enhance energy efficiency. Finally, exploring the applicability of the AFSMC to other types of mobile robots, such as wheeled inverted pendulums, quadrotors, and humanoid robots, can further broaden its potential applications and demonstrate its versatility.

Furthermore, it is recommended to explore the potential benefits of integrating the AFSMC with advanced sensing technologies, such as LiDAR or cameras, to enable the robot to perceive its environment more accurately and adapt its trajectory accordingly. This could enhance the robot's ability to navigate complex environments and perform tasks that require real-time decision-making.

REFERENCES

Ahmad Abu Hatab, R. D. (2013). Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework. *Advances in Robotics & Automation*. doi:https://doi.org/10.4172/2168-9695.1000107

Ahmed, F., Elsayed, A., & Ibrahim, A. (2016). Adaptive Sliding-mode Dynamic Controller for nonholonomic Mobile Robots. *IEEE*.

Alam, T., Qamar, S., Dixit, A., & Benaida, M. (2021). Genetic algorithm. *In International Journal of Engineering Pedagogy, 10*(6), 57–77. doi: https://doi.org/10.3991/IJEP.V10I6.14567

Al-Mayyahi, A., Wang, W., & Birch, P. (2016). Design of Fractional-Order Controller foTrajectory Tracking Control of a Non-holonomic Autonomous Ground Vehicle. *Journal of Control, Automation and Electrical Systems, 27(1)*, 29–42. doi:https://doi.org/10.1007/s40313-015-0214-2

Benaziza, W., Slimane, N., & Mallem, A. (2017). Mobile Robot Trajectory Tracking Using Terminal Sliding Mode comtrol. *the 6th International Conference on Systems and Control* (pp. 538-542). Batna,: 978-1-5090-3960-9/17/$31.00 ©2017 IEEE.

Benchouche, W. M. (2021). *The Impact of the Dynamic Model in Feedback Linearization Trajectory Tracking of a Mobile Robot.* Periodica Polytechnica Electrical Engineering and Computer Science. Retrieved from https://doi.org/10.3311/PPee.17127

Benchouche, W., Mellah, R., & Bennouna, M. S. (2021). The Impact of the Dynamic Model in Feedback Linearization Trajectory Tracking of a Mobile Robot. *Periodica Polytechnica Electrical Engineering and Computer Science, 65(4)*, 329–343. doi:https://doi.org/10.3311/PPee.17127

Cen, H., & Singh, B. K. (2021). Nonholonomic Wheeled Mobile Robot Trajectory Tracking Control Based on Improved Sliding Mode Variable Structure. *Wireless Communications and Mobile Computing*, pp. 1-9.

Chen, Li, T.-H. S., & Yeh, Y.-C. (2009). EP-based kinematie control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile rabot. *Information Sciences*, (pp. 180-195).

Fierro R., & Lewis, F. L. (1997). Control of a nonholomic mobile robot: Backstepping kinematics into dynamics. *Journal of Robotic Systems, 14(3)* , 149–163. Retrieved from https://doi.org/10.1002/(SICI)1097-4563(199703)14:3<149: AID-ROB1>3.3.CO;2-N

Fufa, L., & Ayenew, E. (2023). Trajectory Tracking of a Two-Wheeled Mobile Robot Using Backstepping and Nonlinear PID Controller. *Springer, 455*. doi:https://doi.org/10.1007/978-3-031-28725-1_18

Gao X, Y. L. (2021). *Modeling and Analysis in Trajectory Tracking Control for Wheeled Mobile Robots with Wheel Skidding and Slipping: Disturbance Rejection Perspective.* Actuators. doi:https://doi.org/10.3390/act10090222

Garbaabaa, H. A., Geda, M. G., Wase, M. G., Ranganathan, S., Jin, G.-G., & Yung-Deug, S. (2020). A Self-Tuning NPID Control Method for FOPTD Processes. *Preprints2020*, 1-13. doi:https://doi.org/(doi: 10.20944/preprints202004.0410.v1)

Guevara, L., Camacho, O., Rosales, A., Guevara, J., & Scaglia, a. G. (2019). ‘‘A linear algebra controller based on reduced order models applied to trajectory tracking for mobile robots: An experimental validation,’’. *Int. J. Autom. Control*, p. 176.

Hadi, N. H., & Younus, K. K. (2020). Path tracking and backstepping control for a wheeled mobile robot (WMR) in a slipping environment. *IOP Conference Series: Materials Science and Engineering, 671(1)*. doi:https://doi.org/10.1088/1757-

HASSANI, I., MAALEJ, I., & REKIK, C. (2020). Backstepping tracking control for nonholonomic. *4th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*. IEEE.

Hwang, E.-J., Kang, H.-S., Hyun, C.-H., & Park, M. (2013). Robust Backstepping Control Based on a Lyapunov Redesign for Skid-Steered Wheeled Mobile Robots. *International Journal of Advanced Robotic Systems, 26*. doi:https://doi.org/10.5772/55059

JIN, G.-G., & SON, Y.-D. (2019). Design of a Nonlinear PID Controller and Tuning Rulesfor First-Order Plus Time Delay Models. *Studies in Informatics and Control, 28(2)*, 157–166. doi: https://doi.org/10.24846/v28i2y201904

KALYONCU, M., & DEMİRBAŞ, F. (2017). DIFFERENTIAL DRIVE MOBILE ROBOT TRAJECTORY TRACKING WITH USING PID AND KINEMATIC BASED

BACKSTEPPING CONTROLLER. *Selcuk University Journal of Engineering, 5(1)*, 1–15. Retrieved from https://doi.org/10.15317/Scitech.2017.65

Martins, N. A., & Bertol, D. W. (2022). Wheeled Mobile Robot Control. In Studies in Systems, Decision and Control. *springer, 380*. doi:https://link.springer.com/10.1007/978-3-030-77912-2

Messom, C. (2002). Genetic Algorithms for Auto-tuning Mobile Robot Motion Control. 129–134. Retrieved from http://www.massey.ac.nz/~wwiims/research/letters/

Mohamed, M., & Hamza, M. (2019). Design PID Neural Network Controller for Trajectory Tracking of Differential Drive Mobile Robot Based on PSO. *Engineering and Technology Journal*, 574–583. doi:https://doi.org/10.30684/etj.37.12A.12

Mohareri, O. D. ((2012). *Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks.* Neurocomputing. doi:https://doi.org/10.1016/j.neucom.2011.06.035

Morales, L., Herrera, M., & Camacho, O. (2021). LAMDA Control Approaches Applied to Trajectory Tracking for Mobile Robots. *IEEE*.

Rabbani, M., & Memon, A. (2021). Trajectory Tracking and Stabilization of Nonholonomic Wheeled Mobile Robot Using Recursive Integral Backstepping Control. *Electronics*. Retrieved from https://doi.org/10.3390/electronics10161992

Rubio, F., Valero, F., & Llopis-Albert, C. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications.

Siegwart R, N. I. (2011). *Introduction to Autonomous Mobile Robots.*

Song, Y.-D. (2018). Control of Nonlinear Systems via PI, PD and PID. *In Control of Nonlinear Systems via PI, PD and PID*. doi:https://doi.org/10.1201/9780429455070

Tiep, D., Lee, K., Im, Y., Kwak, B., & Ryoo, Y. (2018). Design of Fuzzy-PID Controller for Path Tracking of Mobile Robot with Differential Drive. *INTERNATIONAL JOURNAL of FUZZY LOGIC and INTELLIGENT SYSTEMS, 18(3)*, 220–228. Retrieved from https://doi.org/10.5391/IJFIS.2018.18.3.220

Tzafestas, S. G. (2014). Introduction to Mobile Robot Control. In Introduction to Mobile Robot Control. *Elsevier*. doi:https://doi.org/10.1016/C2013-0-01365-5

Uddin, N. (2018). Trajectory Tracking Control System Design For Autonomous Two-Wheeled 19 Robot. *JURNAL INFOTEL, 10(3)*, 90.

Wang, C., & Liu, X. (2018). Trajectory Tracking of an Omni-Directional Wheeled Mobile Robot Using a Model Predictive Control Strategy. *applied sciences*.

Zaidner, G., Korotkin, S., Shteimberg, E., Ellenbogen, A., Arad, M., & Cohen, Y. (2010). Non linear PID and its application in process control. *IEEE, IEEE 26th Convention of Electrical and Electronics Engineers in Israel*, 574-577. doi:https://doi.org/10.1109/EEEI.2010.5662155

Zhao, H., H., D., K., Han, & Lee, J. M. (2015). Adaptive neural network-based PI sliding mode contral of nonholonomic mobile rabot. *12th International Conference on In Ubiquitous Robots and Ambient Intelligence (URAI)* (pp. 473-478). IEEE.

APPENDIXES

Appendix A: MATLAB Functions

Appendix A1: Dynamic Model Matlab Function

```
function [vDot, omegaDot] = Dynamics(Tr, Tl, v, omega, theta)
mt = 120; % The robot mass
r = 0.1350; % The radius of the wheels
d = 0; % The distance between the CoM and wheels axis
L = 0.330; % The lateral distance of the wheels to the center
I = 15.0656; % inertial of motor
H = [mt 0 mt*d*sin(theta); 0 mt -mt*d*cos(theta); mt*d*sin(theta) -mt*d*cos(theta) I];
C = [0 0 mt*d*omega*cos(theta); 0 0 mt*d*omega*sin(theta); 0 0 0];
S = [cos(theta) -d*sin(theta); sin(theta) d*cos(theta);0 1];
E =[cos(theta)/r cos(theta)/r; sin(theta)/r sin(theta)/r; L/r -L/r];
Sdot =[-omega*sin(theta) -d*omega*cos(theta); omega*cos(theta) -d*omega*sin(theta); 0 0];
Hhat = S'*H*S; Chat = S'*C*S+S'*H*Sdot; Ehat = S'*E;
T = [Tr; Tl]; % input Torques
w = [v; omega]; % Velocity vector
vdot = (Hhat)\(Ehat*T-Chat*w);
vDot = 4; omegaDot = vdot(2); % Accelerations Outputs
```

Appendix A2: Kinematic Model Matlab Function

```
function [xDot, yDot, thetaDot] = Kinem (v, omega, theta)
S = [cos(theta) -d*sin(theta); sin(theta) d*cos(theta); 0 1];
w = [v; omega];
qdot = S*w;
xDot = qdot (1); yDot = qdot (2); thetaDot = qdot (3);
```

Appendix A3: Nonlinear Feedback Torque Controller Matlab Function

```
function [Tr, Tl] = feedback_T (v, omega, theta, u1, u2)
H = [mt 0 mt*d*sin(theta); 0 mt -mt*d*cos(theta); mt*d*sin(theta) -mt*d*cos(theta) I];
C = [0 0 mt*d*omega*cos(theta); 0 0 mt*d*omega*sin(theta); 0 0 0];
S = [cos(theta) -d*sin(theta); sin(theta) d*cos(theta); 0 1];
E = [cos(theta)/r cos(theta)/r; sin(theta)/r sin(theta)/r; L/r -L/r];
```

```
Sdot= [-omega*sin(theta) -d*omega*cos(theta); omega*cos(theta) -d*omega*sin(theta); 0
0];
Hhat = S'*H*S;  Chat = S'*C*S + S'*H*Sdot;    Ehat = S'*E;
u = [u1; u2];
w = [v; omega];
T = (Ehat) \ (Hhat*u+ Chat*w);
Tr = T (1);  Tl = T (2);       % output torques
```

Appendix A4: Sliding Mode Controller Matlab Function

```
function [Vc, omega_c] = SMC(Vr, omega_r, qe1, qe2, qe3)
L1=1.5; L2=6; L3=1;
Q1=0.1; Q2=0.3; alpha=0.2;
Q=[Q1 0;
   0 Q2];
sigma=[-L1^2*qe1;
   L1^2*qe1*qe2-(L2*qe1+L3)*(L2*qe2+L3*qe3)];
e_Alpha= [exp(alpha*abs(sigma(1))) 0;
   0 exp(alpha*abs(sigma(2)))];
A = [L1*Vr*cos(qe3);
   L2*Vr*sin(qe3)+L3*omega_r];
B = [-1/L1 -qe2/(L2*qe1+L3);
   0 -1/(L2*qe1+L3)];

V= -B*A -Q*e_Alpha*[sign(sigma(1));sign(sigma(2))];
Vc = V(1);
omega_c = V(2);
```

Appendix A5: NPID Controller Nonlinear Gain Matlab Function

```
function ke = scale_error(e)
% ec_change = 0.5;
ec_change = 1;
% ec_change = 1.5;
ke = (exp(-e.^2/(2*ec_change^2))) *e;
```

Appendix A6: Trajectory Generation Matlab Function

```
function [xRef, yRef, tRef, vRef, omeRef] = tra(t)
```

```
xRef = -5*sin(t/5);        yRef = 10*sin(t/10);
dxRef = -cos(t/5);        yRef = cos(t/10);
ddxRef = sin(t/5)/5;        ddyRef = -sin(t/10)/10;
vRef = sqrt (dxRef. ^2 + dyRef. ^2);   tRef = atan2(dyRef, dxRef);
omeRef = (ddyRef. *dxRef - ddxRef. *dyRef)/ (dxRef. ^2+dyRef. ^2);
```

Appendix A7: Error Transformation Matlab Function

```
function [qe1, qe2, qe3] = qe(eX, eY, eTh, theta)
e=[eX eY eTh]';
qee = [cos(theta) sin(theta) 0;
    -sin(theta) cos(theta) 0;
    0 0 1];
qe = qee*e;
qe1 = qe(1);
qe2 = qe(2);
qe3 = qe(3);
```

Appendix B: MATLAB/Simulink System Block Diagrams

Appendix B1: Open-loop system simulation Simulink block

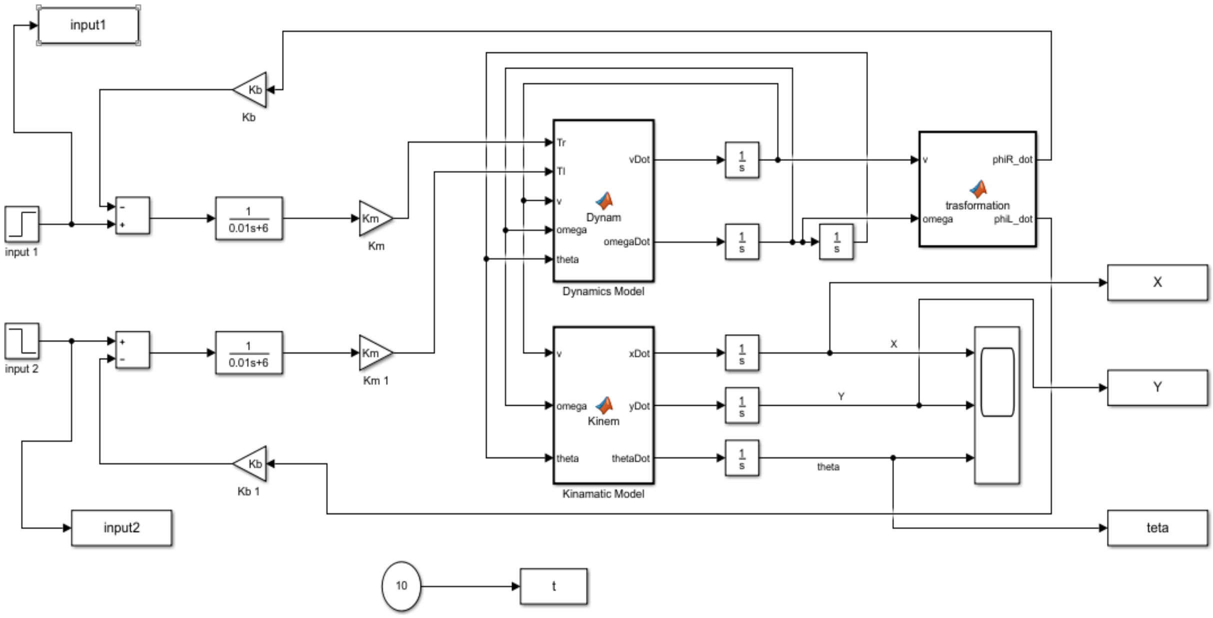

Figure B.1: Open-loop system Simulink model

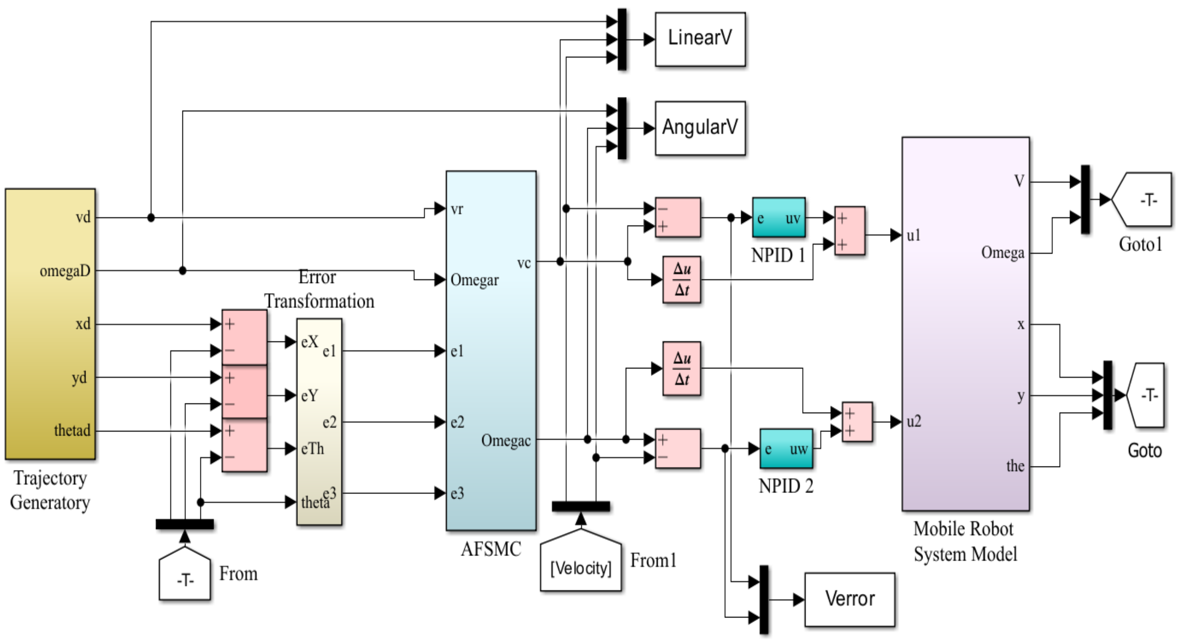

Figure B.2: Overall system simulations Simulink block

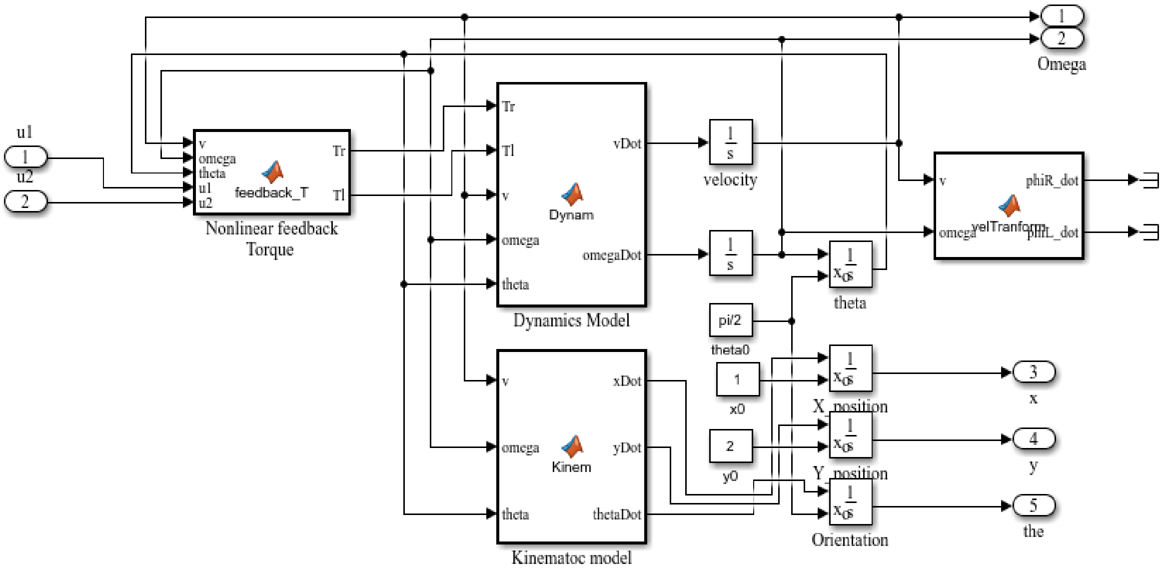

Figure B.3: Expanded mobile robot system model