Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Enhancing Trajectory Tracking Accuracy in Three Wheeled Mobile Robots using Backstepping Fuzzy Sliding Mode Control

A Thesis Submitted to School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Control Engineering

By

Yebekal Adgo

Advisor

Dr. Lebsework Negash

October, 2023

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

This is to certify that the thesis prepared by Yebekal Adgo , entitled "**Enhancing Trajectory Tracking Accuracy in Three Wheeled Mobile Robots using Backstepping Fuzzy Sliding Mode Control"**, submitted in partial fulfillment of the requirements for the degree of Master of Sciences in Control Engineering complies with the regulations of university and meets the accepted standards with respect to originality and quality.

Approved by Board of Examiners

Name	Signature	Date
Dr. Bisrat Derebssa		
(School dean)		
Dr. Lebsework Negash		
(Chairman)		
Dr. Lebsework Negash		
(Advisor)		
Dr. Challa Merga		
(External Examiner)		
Mrs.Yalemzeref Getnet		
(Internal Examiner)		

Declaration

I hereby declare that the research study entitled **"Enhancing Trajectory Tracking Accuracy in Three Wheeled Mobile Robots using Backstepping Fuzzy Sliding Mode Control"** is entirely my own work. This research has not been submitted previously for any academic degree at this institution or any other university. I have duly acknowledged all sources and references utilized in this thesis.

Yebekal Adgo		October , 2023
Name of Student	**Signature**	**Submission Date**

This thesis has been submitted for examination with my approval as a university advisor.

Name of Advisors	**Signature**
Dr. Lebsework Negash	

Dedication

To my beloved father, Adgo Wendemagegn, and my cherished mother, Birtukan Eyi, your unwavering love and support have been the cornerstone of my life's journey. I dedicate this achievements to you, with profound gratitude and love.

Yebekal Adgo

Acknowledgment

I am deeply thankful to God, whose unwavering guidance and blessings have illuminated my path and provided me with the strength to overcome challenges throughout this thesis journey. I extend my sincere gratitude to my esteemed thesis advisor, Dr. Lebsework Negash, for his invaluable guidance, unwavering support, and expert insights that have been instrumental in shaping the course of this research. I would also like to express my appreciation to the dedicated control engineering instructors at Addis Ababa Institute of Technology (AAiT) for their exceptional teachings, mentorship, and contributions to my academic growth. To my family, your unconditional love, encouragement, and belief in my potential have been the pillars of my success. Your unwavering support has motivated me to strive for excellence.To my classmates, I am grateful for the solidarity, shared experiences, and intellectual discussions that have enriched my academic journey and made it memorable. I am indebted to all those who have played a role in my educational pursuits. Your collective influence has left an indelible mark on my life, and I am honored to have your support and guidance.

Yebekal Adgo

Abstract

The rise in robotics technology has led to increased interest in three-wheeled mobile robots (TWMRs) due to their agility and adaptability across various applications. However, effectively controlling TWMRs presents a significant challenge owing to their inherent nonholonomic constraint, which restricts their independent movement in all directions. Additionally, factors like sensor noise, nonlinear system dynamics, and uncertain system parameters add to the complexity controlling of TWMRs. This research endeavors to enhance the precision of trajectory tracking in TWMRs. Specifically, it employs Backstepping Fuzzy Sliding Mode Control (BFSMC) with parameters optimized through Particle Swarm Optimization (PSO), coupled with the Extended Kalman Filter (EKF) for state estimation. The study conducts a comprehensive performance comparison between BFSMC and BSMC across various trajectory patterns, revealing substantial improvements in trajectory tracking accuracy with BFSMC. BFSMC demonstrates improved performance compared to BSMC across various trajectory types, quantified by calculating the percentage improvement in trajectory tracking using Integral Absolute Error (IAE). Specifically, it achieves a 51.97% improvement for circular trajectories, an 82.09% improvement for infinity trajectories, and an 84.073% improvement for spiral trajectories.. Moreover, BFSMC demonstrates superior robustness in the presence of disturbances, noise, parameter variations, and unmodeled dynamics compared to BSMC. The integration of the Extended Kalman Filter further improve accuracy, particularly in noisy conditions. Simulation results conducted using MATLAB/Simulink software validate the effectiveness of this approach in achieving superior trajectory tracking accuracy in TWMRs.

Keywords– Trajectory Tracking , EKF , BFSMC , BSMC , TWMR , PSO , Nonholonomic Constraint, Disturbances , Sensor Noise.

Contents

Declaration i

Acknowledgment iii

Abstract iv

List of Figures xi

List of Tables xii

List of Acronyms 1

1 Introduction 1

1.1 Background of the Study 1

1.2 Problem Statement 3

1.3 Objectives of the Research Study 4

1.3.1 General Objective 4

1.3.2 Specific Objectives 4

1.4 Methodology 4

1.5 Contribution of the Thesis 5

1.6 Scope of the Thesis 5

1.7 Structure of the Thesis 6

2 Literature Review 7

2.1 Theoretical Background 7

2.1.1 Wheeled Mobile Robot 7

2.1.1.1 Classification Wheels 7

2.1.1.2 Types of Locomotion of WMR 9

2.1.2 Controller and Algorithms 10

2.1.2.1 Fuzzy Logic Control 10
2.1.2.2 Sliding Mode Control 11
2.1.2.3 Extended Kalman Filter 12
2.1.2.4 Backstepping Control 14
2.1.2.5 Particle Swarm Optimization 15
2.2 Overview of the Existing Surveys 16
2.2.1 The Proposed BFSMC System 19
2.2.2 Control Perspective of Three Wheeled Mobile Robot 19

3 Mathematical Modeling of Three Wheeled Mobile Robot 21
3.1 Introduction 21
3.2 Tricycle Wheeled Mobile Robot 21
3.2.1 Coordinate Frames Transformation 22
3.2.2 Kinematics Modeling of Three Wheeled Mobile Robot 23
3.2.3 Dynamics Modeling of Three Wheeled Mobile Robot 28
3.2.4 Actuator Dynamics Modeling 32
3.2.5 Model Verification for Three Wheeled Mobile Robot 34

4 Controller Design 37
4.1 Introduction 37
4.2 Backstepping Controller Design 38
4.2.1 Stability Analysis for Backstepping Controller Using Lyapunov Stability Theorm 41
4.3 Sliding Mode Controller Design 41
4.3.1 Lyapunov Stability Analysis of SMC 43
4.4 Fuzzy Sliding Mode Controller Design 45
4.5 Extended Kalman Filter Design for Position and Orientation Estimation of TWMR 47
4.5.1 Steps for Designing of Extended Kalman Filter 48
4.6 Particle Swarm Optimization 49
4.6.1 Steps for PSO-Based Control Gain Tuning in MATLAB-Simulink . . 49

5 Simulation Results and Discussion 51
5.1 Introduction 51
5.2 Simulation Result for Trajectory Tracking of TWMR using BSMC and BFSMC 52

5.2.1 Circle Trajectory Tracking 52
5.2.1.1 Circle Trajectory Tracking using BSMC 52
5.2.1.2 Circle Trajectory Tracking using BFSMC 53
5.2.2 Infinity Trajectory Tracking 55
5.2.2.1 Infinity Trajectory Tracking using BSMC 55
5.2.2.2 Infinity Trajectory Tracking using BFSMC 56
5.2.3 Stair Trajectory 58
5.2.3.1 Stair Trajectory Tracking using BSMC 58
5.2.3.2 Stair Trajectory Tracking using BFSMC 59
5.2.4 Spiral Trajectory 60
5.2.4.1 Spiral Trajectory Tracking using BSMC 60
5.2.4.2 Spiral Trajectory Tracking using BFSMC 61
5.2.5 Square Trajectory 63
5.2.5.1 Square Trajectory Tracking Using BSMC 63
5.2.5.2 Square Trajectory Tracking Using BFSMC 64
5.3 Extended Kalman Filter Simulation Result 66
5.3.1 State Estimations for Circular Trajectory using EKF 66
5.4 Simulation Result Comparison BSMC and BFSMC 72
5.4.1 Chattering Comparison 72
5.4.2 Under Nominal Conditions Trajectories Comparison 73
5.4.2.1 Performance Indices Comparison between BFSMC and BSMC for Various Trajectories 74
5.4.2.2 Trajectory Tracking Improvement Comparison of BSMC and BFSMC 75
5.4.3 Disturbance and Uncertainty of Unmodeled Dynamics 75
5.4.4 Robustness Analysis Between BSMC and BFSMC 76
5.4.4.1 Infinity Trajectory 76
5.4.4.2 Square Trajectory 78
5.4.4.3 Circular Trajectory 80

6 Conclusion and Future Works **82**
6.1 Conclusion 82
6.2 Future Work 83

Bibilography **84**

A MATLAB Code **89**

A.1 PSO Code 89

A.2 Rectangular Trajectory Generation 90

B MATLAB® Script **91**

C Appendix® Formula **95**

List of Figures

1.1 Wheeled Mobile Robot [12] . . . 3

2.1 a) Conventional wheel b) Pivot wheel c) Swedish Wheel d) Spherical Wheel[13] 8
2.2 Different Kinds of Locomotion Types for Wheeled Mobile Robots [13] 9
2.3 Fuzzy Logic Control System Architecture . . . 10
2.4 Reaching Phase , Sliding Phase and Chattering Phenomena in SMC 12

3.1 The Global Reference Frame and the Robot Reference Frame of TWMR. . . 22
3.2 Nonholonomic Constraint on the Robot Motion . . . 25
3.3 Wheel Velocities and Inertial frame. . . . 25
3.4 DC Motor Architecture . . . 32
3.5 Various Modes of Motion for TWMRs[55] . . . 35
3.6 Mathematical Model of TWMR . . . 35
3.7 Simulation Result of Model Verification . . . 36

4.1 General Block Diagram of the System . . . 37
4.2 Posture Error Description . . . 38
4.3 Fuzzy Logic Controller Block Diagram . . . 45
4.4 Input and Output Membership Function for FLC . . . 46

5.1 Simulation Result of Pose for Circular Trajectory using BSMC 52
5.2 Torque and Pose Error for Circular Trajectory using BSMC 52
5.3 X-Y Plot of Circular Trajectory Simulation Result using BSMC 53
5.4 Simulation Result of Pose for Circular Trajectory using BFSMC 53
5.5 Pose Error and Torque for Circular Trajectory using BFSMC 54
5.6 X-Y Plot for Circular Trajectory using BFSMC 54
5.7 Simulation Result for Pose of Infinity Trajecyory using BSMC 55
5.8 Torque and Pose Error for Infinity Trajectory using BSMC 55

5.9 X-Y Plot of Infinity Trajectory using BSMC 56
5.10 Pose for Infinity Trajectory Tracking Result using BFSMC 56
5.11 Trajectory Tracking Error and Torque for Infinity Trajectory using BFSMC 57
5.12 X-Y Plot for Infinity Trajectory Tracking Result using BFSMC 57
5.13 Simulation Result for Pose for Stair Trajectory using BSMC 58
5.14 Pose Error and Torque for Stair Trajectory using BSMC 58
5.15 Pose for Stair Trajectory Tracking Result of WMR using BFSMC 59
5.16 Trajectory Tracking Error for Pose and Torque for WMR using BFSMC for Stair Trajectory 59
5.17 Simulation Result for Pose of Spiral Trajectory using BSMC 60
5.18 Pose Error and Torque for Spiral Trajectory using BSMC 60
5.19 X-Y Plot of Spiral Trajectory using BSMC 61
5.20 Pose for Spiral Trajectory Tracking Result of WMR using BFSMC 61
5.21 Trajectory Tracking Error for Pose and Control Effort for WMR using BFSMC for Spiral Trajectory 62
5.22 X-Y Plot for Spiral Trajectory Tracking Result using BFSMC 62
5.23 Simulation Result for Pose of Square Trajectory using BSMC 63
5.24 Pose Error and Torque using BSMC for Square Trajectory 63
5.25 X-Y Plot of Square Trajectory using BSMC 64
5.26 Pose for Square Trajectory Tracking Result using BFSMC 64
5.27 Trajectory Tracking Error and Torque for Square Trajectory using BFSMC . 65
5.28 X-Y Plot of Square Trajectory Tracking Using BFSMC 65
5.29 Pose Estimation of TWMR Without Addition of Noise 66
5.30 Pose Estimation Error of TWMR Without addition of Noise 67
5.31 Pose Estimation TWMR after Addition of Noise 67
5.32 Pose Estimation Error of TWMR after Addition of Noise 68
5.33 Pose Actual Trajectory versus EKF Output Trajectory after Noise Addition 68
5.34 Pose Actual Trajectory versus EKF Output Trajectory Without Noise Addition 69
5.35 Circular Trajectory Tracking with Presence of Noise using BSMC without EKF 70
5.36 Circular Trajectory Tracking with Presence of Noise using BSMC with EKF 70
5.37 Circular Trajectory Tracking without Noise using BSMC with EKF 71
5.38 Circular Trajectory Chattering Comparison between BSMC and BFSMC . . 73
5.39 Infinity Trajectory Tracking Performance Comparison BSMC and BFSMC . 73
5.40 Spiral Trajectory Tracking Performance Comparison BSMC and BFSMC . . 74

5.41 White Noise , External disturbance , and Disturbance Torque 76
5.42 Infinity Trajectory Tracking Comparison between BSMC and BFSMC with Parameter Variation and Disturbance 77
5.43 Square Trajectory Tracking after Disturbance Added to the System using BSMC 78
5.44 Square Trajectory Tracking After Disturbance Added to the System using BFSMC 79
5.45 Circle Trajectory Tracking after Friction Introduced to the System using BF-SMC and BSMC Comparison 80

B.1 General System Design in MATLAB/SIMULINK Software 91
B.2 Dynamics Controller System Design in MATLAB/SIMULINK Software . . . 92
B.3 Backstepping Controller and Kinematics Controller Design in MATLAB/SIMULINK Software 92
B.4 Rotational Matrix Design of the System in MATLAB/SIMULINK Software 93
B.5 Kinematics and Dynamics Model of TWMR Design in MATLAB/SIMULINK Software 93
B.6 Extended Kalman Filter Design in MATLAB/SIMULINK Software 94
B.7 Fuzzy Logic Controller Design in MATLAB/SIMULINK Software 94

List of Tables

2.1 PSO Algorithms Mathematical Expressions 15

3.1 System Parameter Values[54] 34

4.1 Rules Base of the Fuzzy Logic Controller 46

5.1 Performance Comparison of BSMC and BFSMC Under Nominal Conditions. 74

5.2 Percentage of performance Improvement by BFSMC over BSMC 75

5.3 Performance Comparison of BSMC and BFSMC for Infinity Trajectory Tracking 77

5.4 Performance Comparison of BSMC and BFSMC for Circular Trajectory Tracking with the Presence of Friction Forces 80

C.1 Parameters used for Stair Trajectories Simulation 95

Chapter 1

Introduction

1.1 Background of the Study

A robot is an autonomous or semi-autonomous mechanical device designed to perform tasks with a certain degree of autonomy. These machines are typically equipped with sensors, actuators, and a programmable control system to observe their surroundings, make decisions, and execute actions to achieve specific objectives [1].

Robots vary widely, from simple machines that perform repetitive tasks to advanced systems capable of complex interactions and decision-making. The concept of robotics has a long history, with roots tracing back to the mid-20th century, when early pioneers like Alan Turing and John von Neumann laid the theoretical foundations [2]. However, the development of robots as we know them today took a significant step forward in 1954 when the first industrial robot, named Unimate, was created by George Devol and Joseph Engelberger. This robot featured a wheeled base and played a crucial role in revolutionizing factory automation [3]. As the 1960s and 1970s unfolded, notable creations like Shakey the Robot emerged from Stanford Research Institute, utilizing wheels for navigation [4].

Three-wheeled mobile robots (TWMRs) as shown in figure 1.1, also known as differential-drive robots, have garnered significant attention due to their unique design and agility. These robots typically consist of two main driving wheels, usually located on the rear axle, and a caster wheel at the front, enabling them to maneuver swiftly and efficiently in constrained spaces. Over the years, TWMRs have evolved in terms of their kinematics and control systems, making them suitable for various applications [5].

Despite their early development, mobile robots have come a long way. With advancements in control algorithms, including Proportional-Integral-Derivative (PID) control and

Model Predictive Control (MPC), these robots have gained enhanced autonomy . Additionally, the integration of advanced sensors like light detection and ranging (LiDAR) and cameras has significantly improved perception and navigation capabilities [6]. As a result, wheeled mobile robots have diversified in their applications, finding roles in logistics, agriculture, healthcare, search and rescue, and more. They play a vital role in warehouse automation, exemplified by Amazon's Kiva robots, autonomous vehicles, and even planetary exploration missions like National Aeronautics and Space Administration(NASA's) Mars rovers [7].

This thesis primarily focuses on Three Wheeled Mobile Robots (TWMRs) due to their extensive utilization in the realm of mobile robots. TWMRs are known for their swift maneuverability, straightforward control mechanisms, and energy-efficient attributes [8, 9]. However, controlling TWMRs poses several challenges. These robots are subject to nonholonomic constraints, limiting their motion and requiring sophisticated control strategies to follow desired paths. Additionally, they often operate in uncertain and dynamic environments, necessitating robust control algorithms. Sensor noise can affect control accuracy, particularly in applications requiring precise navigation .

The field of wheeled mobile robot trajectory tracking control has been the focus of extensive research efforts [10]. Achieving effective position estimation techniques is essential for mobile robots to navigate efficiently, especially in environments where sensor measurements are corrupted by noise. Localization is a critical aspect of mobile robotics, allowing robots to interpret sensory information, determine their location, plan their actions, and control their motion effectively [11].

To navigate successfully, robots go through various stages, including perception, localization, cognition, and motion control.

In conclusion, the control of TWMRs is a complex but vital field in robotics. These robots have evolved significantly since their early days and are now integral to various industries. This thesis explores the use of Backstepping Fuzzy Sliding Mode Control (BFSMC) and Extended Kalman Filter (EKF) to address the challenges of controlling TWMRs. BFSMC combines fuzzy logic control, sliding mode control, and backstepping control principles, providing robustness and adaptability. EKF, on the other hand, enhances state estimation in the presence of sensor noise. The integration of these techniques aims to improve the performance and stability of TWMRs, allowing them to navigate effectively in various environments.

Figure 1.1: Wheeled Mobile Robot [12]

1.2 Problem Statement

The demand for autonomous mobile robots is on the rise, but the environments they navigate are highly dynamic. While wheeled robots are widely used, creating effective motion controllers for them remains challenging due to various complexities. These complexities stem from factors like nonlinear dynamics, uncertain system parameters, actuator dynamics and delays, sensing and perception challenges, robustness, real-time control demands, and accurate system identification. One significant hurdle is the non-holonomic constraint, arising from a robot's mechanical structure or locomotion mechanism. For instance, a differential drive robot's sideways movement is constrained unless specific control strategies or mechanisms are employed. Sensor noise is another critical issue. Noisy sensor measurements, like odometry or position sensors, can introduce uncertainties in estimating the robot's state variables - its position, velocity, and orientation. This uncertainty can affect control inputs and lead to unstable robot behavior. This study focuses on controlling a three wheeled mobile robot (TWMR) using a combination of Backstepping fuzzy sliding mode controller (BFSMC) and extended kalman filter (EKF). The aim is to create a control strategy that blends the robustness of BFSMC with EKF's state estimation capabilities. This strategy aims to achieve precise and stable motion control for the TWMR in the face of uncertainties, nonlinearities, and disturbances that are inherent to the system.

1.3 Objectives of the Research Study

1.3.1 General Objective

- Enhancing trajectory tracking accuracy in three wheeled mobile robots using backstepping fuzzy sliding mode control.

1.3.2 Specific Objectives

- To model kinematics and dynamics of three wheeled mobile Robot.
- To estimate the robot's states and compensate for sensor noise and measurement inaccuracies using extended kalman filter(EKF).
- To design backstepping fuzzy sliding mode control for tracking trajectories of three wheeled mobile robot.
- To simulate and verify the proposed system.
- To compare BFSMC with BSMC

1.4 Methodology

For the accomplishment of designing proposed trajectory tracking and pose estimation the following procedures have been performed:

- Define research problem and set the general objective of the study.
- Literature survey to three wheeled mobile robot approach.
- Derive more precise mathematical model of three wheeled mobile robot kinematics and dynamics Model.
- Model verification of kinematics and dynamics model.
- Set up three wheeled mobile robot control strategy.
- Design backstepping controller for kinematics model and sliding mode control for dynamics Model.
- Design of extended kalman filter for position and orientation estimation.

- Interfacing BFSMC and EKF.
- Compare performance of proposed system with different approaches.
- Check the robustness and performance of the System.
- Analysis and optimization.

1.5 Contribution of the Thesis

Major contribution of this research study are listed below:

- Advancement of knowledge, in robotics , control system and artificial intelligence.
- Integration position and orientation estimation algorithm to enhance controller performance.
- Integrated Control Strategy
- Comparative Evaluation.

1.6 Scope of the Thesis

The scope of this thesis are the design and validation of a kinematics and dynamics model for a three Wheeled Mobile Robot (TWMR). The kinematic model serves as the basis for developing a kinematic controller, known as the backstepping controller . Additionally, the dynamic model is utilized for the design of a dynamic controller called backstepping fuzzy sliding mode control (BFSMC). To address the impact of sensor noise on the system, an extended kalman filter (EKF) is employed. The integration of the EKF and BFSMC aims to enhance the performance of the control system. The stability of the designed controller is assessed based on the Lyapunov stability theorem. The performance of the controller is evaluated through experiments involving the introduction of white noise, parameter variations, and disturbances into the system. The overall designed system is simulated and analyzed using MATLAB/SIMULINK software. It is important to note that practical implementation of the system is not included in this thesis.

1.7 Structure of the Thesis

This thesis comprises six chapters, commencing with an introductory section. In **Chapter Two**, an extensive literature review is conducted, surveying prior research and investigations within the field. **Chapter Three** encompasses the development and validation of the system's mathematical model, involving conversions from the local reference frame to the global reference frame through orthogonal transformations. This chapter also entails an analysis of the dynamics model, kinematics model, and actuator model, with confirmation of model accuracy via MATLAB software. **Chapter Four** centers on the controller design for trajectory tracking and pose estimation. Specifically, it outlines the creation of the backstepping Controller for the TWMR's kinematics and the backstepping fuzzy sliding mode control for the dynamics model. Additionally, the chapter delves into the design of the extended kalman filter for pose estimation. **Chapter Five** offers a presentation of simulation results, which assess the performance of the designed controllers across various trajectories. A careful analysis is conducted to evaluate the control system's effectiveness and efficiency. Finally, **Chapter Six** provides a thesis conclusion, summarizing the research's key findings and contributions. Moreover, it identifies and discusses potential avenues for future studies and progress.

Chapter 2

Literature Review

2.1 Theoretical Background

2.1.1 Wheeled Mobile Robot

A mobile robot is a type of robotic system designed to move and operate in various environments, both indoors and outdoors, without being physically constrained to a fixed location. It is equipped with locomotion capabilities that allow it to navigate and explore its surroundings autonomously or under remote control.The wheel has been by far the most popular locomotion mechanism in mobile robotics and in man-made vehicles in general. It can achieve very good efficiencies, and does so with a relatively simple mechanical implementation[13].

2.1.1.1 Classification Wheels

- **Standard Wheel**: Often used in combination with other types of wheels, such as driven wheels or swivel wheels, to achieve desired functionality and maneuverability. For instance, a robot may have standard wheels at the front and rear for stability while using driven wheels for propulsion or swivel wheels for enhanced maneuverability. This wheel has Two degrees of freedom; rotation around the (motorized) wheel axle and the contact point[13].

- **Castor Wheel**: It is designed to rotate freely in any direction around a vertical axis, allowing the object it is attached to to move and change direction easily. The swivel bearing enables the wheel to rotate 360 degrees, providing omnidirectional movement. This wheel has two degrees of freedom; rotation around an offset steering joint[13].

- **Swedish Wheel**: Also known as the Mecanum wheel or Ilon wheel, is a specialized

omnidirectional wheel design that enables a mobile robot to move in any direction without the need for additional steering mechanisms. The Swedish wheel consists of a regular wheel with a series of small, angled rollers mounted around its circumference. These rollers are oriented at a specific angle, typically 45 degrees, and have a free-rotating capability. By independently controlling the speed and direction of each wheel, a mobile robot equipped with Swedish wheels can achieve omnidirectional movement. wheel axle, revolving around the rollers, and rotating around the contact point.This wheel consists of three degrees of freedom[13].

- **Ball or Spherical Wheel**: The primary purpose of a ball wheel is to provide support and facilitate movement in any direction, making it especially useful for applications where maneuverability and omnidirectional motion are required. The ball wheel can rotate freely in any direction, allowing the robot or object to move smoothly and change direction without resistance.realization technically difficult[13].

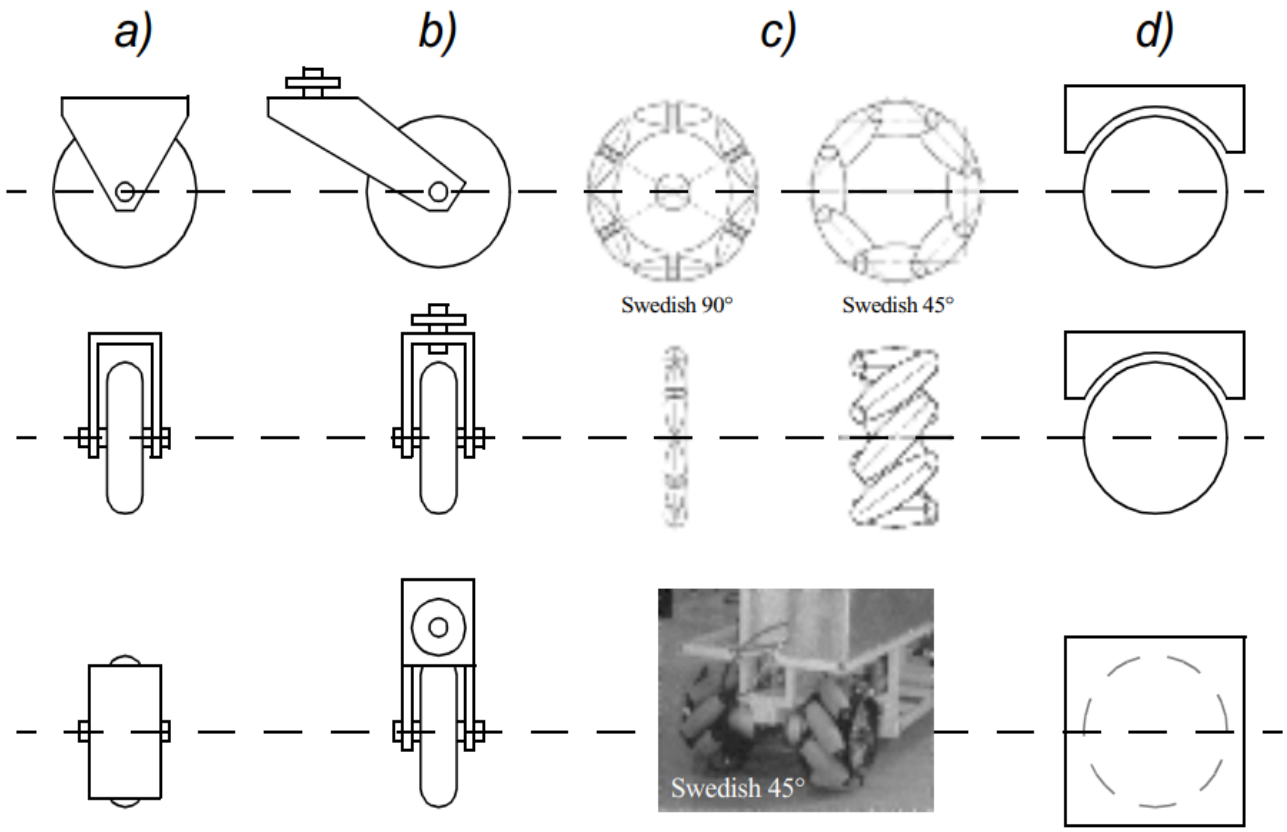


Figure 2.1: a) Conventional wheel b) Pivot wheel c) Swedish Wheel d) Spherical Wheel[13]

2.1.1.2 Types of Locomotion of WMR

Wheeled mobile robots can have the following five types of locomotion system:

- **Differential Drive**:- is a common propulsion mechanism used in mobile robots and vehicles to achieve motion control and maneuverability. It involves the use of two or more wheels, typically on the same axle, that are driven independently and at different speeds to control the direction and movement of the robot. In a differential drive system, each wheel is connected to a separate motor or drive mechanism. By varying the speed and direction of rotation of the two wheels, the robot can achieve different types of motion, including forward and backward movement, turning, and rotation in place.

- **Omnidirectional Drive** : use multiple wheels arranged in specific configurations, such as Mecanum wheels or omni wheels, to achieve high maneuverability and the ability to move in any direction without changing their orientation.

- **Car drive (Ackerman steering)**:- use a traditional steering mechanism with a front-wheel steering angle. They are often used in applications where precise turning and path following are required, such as automated guided vehicles (AGVs) in warehouses or assembly lines.

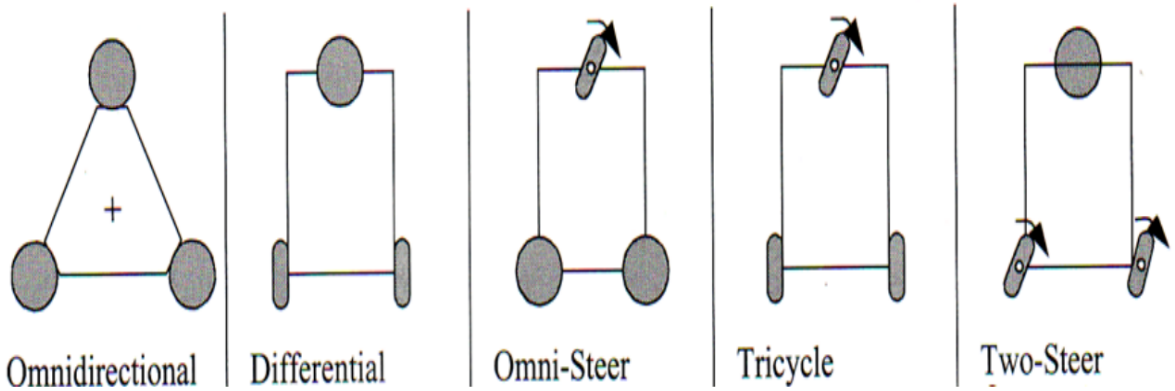


Figure 2.2: Different Kinds of Locomotion Types for Wheeled Mobile Robots [13]

2.1.2 Controller and Algorithms

2.1.2.1 Fuzzy Logic Control

Fuzzy Logic Principles: Fuzzy logic, introduced by Lotfi Zadeh (1965), provides a mathematical framework for dealing with uncertainty. In FLC, linguistic variables, such as "hot," "cold," "high," and "low," are used to represent system parameters and states. These linguistic variables are described by fuzzy sets, which capture the gradual transition between membership and non-membership[14]. A Fuzzy Logic Controller (FLC) is a control system that utilizes fuzzy logic to replicate human reasoning and decision-making processes. Unlike traditional control methods that rely on precise mathematical models, FLC operates using imprecise and uncertain data by employing linguistic variables and rules. Fuzzy Logic Controllers are especially useful in systems with nonlinearities, uncertainties, and imprecise information, as they can capture intricate relationships between input and output variables that may be challenging for conventional control techniques. Designing an effective FLC involves carefully choosing linguistic terms, membership functions, and rules to create a reliable control strategy.

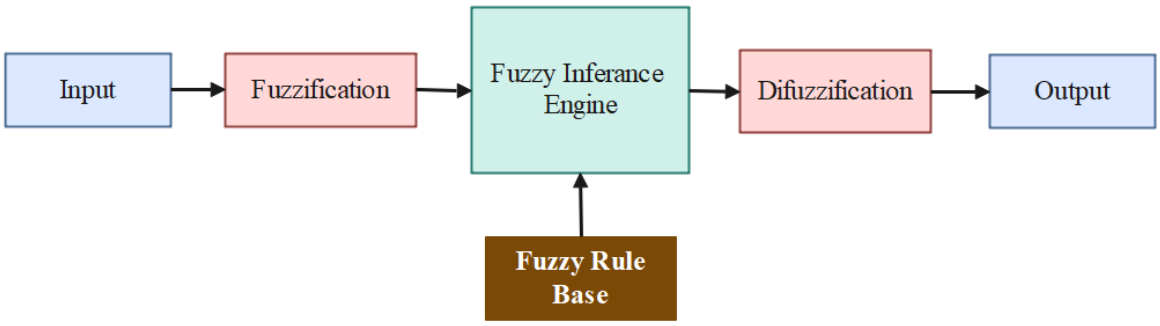


Figure 2.3: Fuzzy Logic Control System Architecture

Fuzzy Logic Controller Architecture

- **Fuzzification**:- is the process of converting crisp input data into fuzzy linguistic terms using membership functions. These membership functions define the degree of membership of a value to a fuzzy set[15].
- **Defuzzification**:- is the inverse process of fuzzifications, converting fuzzy output into a crisp control action. The centroid or weighted average method is often used for defuzzification[15].
- **Fuzzy Rule Base**: The fuzzy rule base contains IF-THEN rules that specify the system's behavior in a premise-consequent form, providing the foundation for approximate

(imprecise) reasoning[16].

- **Fuzzy Inference Engine**: The inference engine evaluates the fuzzy rules based on the current input values. It calculates the firing strength of each rule, reflecting the degree to which each rule's conditions are satisfied. Various methods, including the minimum, product, or maximum, are used to aggregate rule outputs [17].

FLC has found applications in a wide range of fields. In industrial automation, it has been used for process control, robotics, and quality control. In automotive systems, FLC is employed in engine control, ABS braking systems, and automatic transmissions. Other applications include medical diagnosis, consumer electronics, and decision support systems[15]. Advantages: FLC excels in handling systems with uncertainty, imprecision, and nonlinearity. Its ability to capture complex relationships between variables, adapt to changing conditions, and provide interpretable control rules makes it a valuable tool in various domains [18].

2.1.2.2 Sliding Mode Control

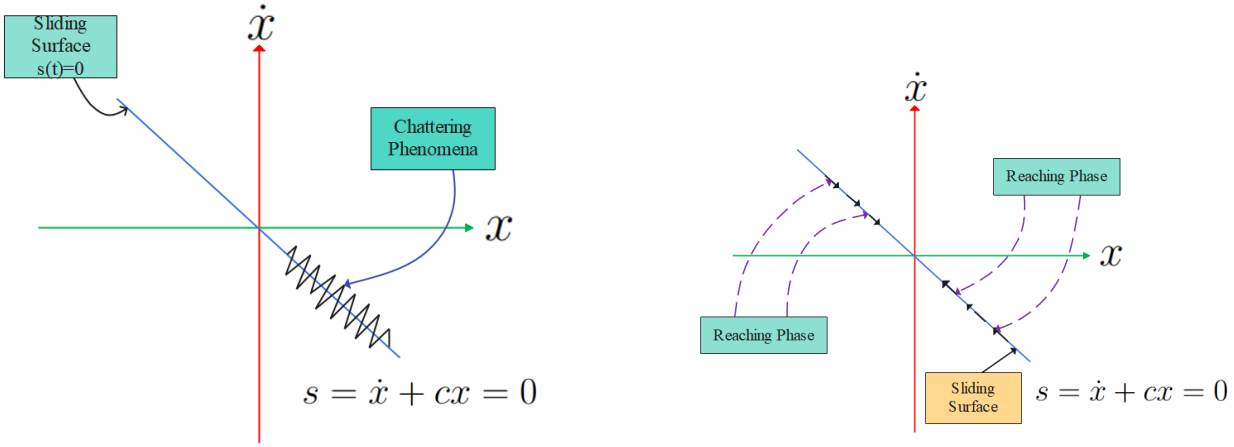
Sliding mode control (SMC) is a robust control technique that has garnered significant attention in the field of control systems due to its ability to effectively handle complex and nonlinear systems. This control strategy was first introduced by V. Utkin in the late 20th century [19] and has since evolved into a powerful tool for controlling a wide range of dynamic systems. SMC operates on the principle of creating a sliding surface, also known as a manifold, within the state space of the system. The objective is to drive the system's state trajectory onto this sliding surface and then maintain it there, ensuring that the system exhibits the desired behavior. This sliding surface acts as a boundary that separates different system behaviors, such as stability and instability. One of the key features of SMC is its robustness to uncertainties, disturbances, and parameter variations in the system. It accomplishes this by employing a discontinuous control law, which actively adjusts the control input based on the system's current state and the sliding surface. This results in rapid and precise control responses even in the presence of external disturbances or model uncertainties [20].

Sliding Phase: In the sliding phase, the control system aims to drive the system's state trajectory onto a designated manifold called the sliding surface as shown in figure 2.4b. This surface represents the desired behavior or equilibrium point. Once the state trajectory reaches this sliding surface, it follows it exactly, ensuring precise tracking of the desired behavior. The sliding phase is the ideal operating condition for SMC, as it guarantees excellent

tracking performance.

Reaching Phase: Before the system enters the sliding phase, it typically goes through the reaching phase. During this phase, the control law generates control inputs to steer the system toward the sliding surface as shown in figure 2.4b. The objective is to reach the sliding surface in finite time. The reaching phase is characterized by transient dynamics as the system approaches the sliding surface.

Chattering Phenomenon: One of the distinctive features of SMC is chattering as shown in figure 2.4a. Chattering occurs when the control input rapidly switches between values to maintain the trajectory precisely on the sliding surface. This rapid switching can lead to high-frequency oscillations in the control signal. While chattering ensures exact tracking of the sliding surface, it can also introduce high-frequency noise and wear and tear on actuators. Chattering is a well-known issue in SMC, and various techniques have been developed to reduce its adverse effects, such as introducing boundary layers around the sliding surface or using switching control with hysteresis [21].



(a) Chattering Phenomena in SMC

(b) Sliding Phase and Reaching Phase in SMC

Figure 2.4: Reaching Phase , Sliding Phase and Chattering Phenomena in SMC

2.1.2.3 Extended Kalman Filter

The kalman filter stands as one of the most influential and extensively applied estimation techniques. It excels in predicting hidden variables, leveraging imperfect and biased observations. Additionally, the Kalman Filter anticipates the forthcoming state of a system based on prior estimations. The filter derives its name from Rudolf E. Kalman (May 19, 1930 – July 2, 2016), who, in 1960, introduced his renowned paper outlining a recursive solution to the discrete-data linear filtering problem. In the present day, the Kalman filter finds

widespread utility across diverse domains, including target tracking (as in Radar systems), location and navigation systems, control systems, computer graphics, and many more [22]. The Extended Kalman Filter (EKF) is a recursive Bayesian filter that extends the Kalman Filter to non-linear systems. It is used for state estimation in dynamic systems where the relationships between variables are described by non-linear equations. The EKF operates by linearizing the system dynamics at each time step, updating the state estimate, and refining its predictions based on sensor measurements. This filter is particularly useful for applications such as navigation, control systems, and sensor fusion in various fields.

EKF Algorithms

- **Prediction Step:**

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k)$$
$$P_k^- = A_k P_{k-1} A_k^T + Q_k$$

- **Correction Step:**

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-))$$
$$P_k = (I - K_k H_k) P_k^-$$

Where:

$\hat{x_k^-}$: represents the a priori estimate of the state at time k.

$f(\dot{})$: is the state transition function.

u_k : represents the control input at time k

p_k^- : is the a priori estimate error covariance matrix.

A_k : is the Jacobian matrix of the state transition function.

z_k : represents the measurement at time k.

Q_k : is the process Noise Covariance Matrix.

k_k : is the kalman gain.

$h(.)$: the measurement prediction function.

H_k : is jacobian matrix of measurement prediction function.

R_k : is the measurement noise covariance matrix.

$\hat{x_k}$: represent the updated estimate of the state at a time k.

p_k : is the updated estimate error covariance matrix.

2.1.2.4 Backstepping Control

Control systems play a pivotal role in enabling machines and autonomous systems to accomplish tasks with precision and accuracy. Among the diverse set of control strategies available, Backstepping control, also known as Backstepping control design, stands out as an effective and widely employed technique. It is particularly well-suited for controlling complex systems characterized by nonlinearity, uncertainties, and stringent performance requirements. The concept of Backstepping control emerged as an extension of traditional control methods to tackle challenging nonlinear systems [23]. It provides a systematic framework for designing controllers that can achieve desired performance levels even in the presence of intricate dynamics and disturbances. Backstepping control's effectiveness in dealing with complex systems can be attributed to its recursive nature and its ability to generate control laws by iteratively designing controllers for each subsystem, thus breaking down the control problem into more manageable parts [24]. The backstepping control strategy has found applications in various domains, including robotics, aerospace, automotive control, and more [25]. Its ability to handle uncertain parameters, mitigate external disturbances, and adapt to changing system dynamics makes it an attractive choice for real-world applications. The backstepping approach has also been enhanced and combined with other control techniques, such as fuzzy logic, sliding mode control, and adaptive control, to further improve its robustness and performance [26] [27].

This introduction provides an overview of the backstepping control strategy and its significance in modern control systems. It highlights its applicability in challenging scenarios and its role in enhancing the performance and robustness of complex systems [28]. The general mathematical expression for backstepping control can be summarized as follows. Consider a dynamic system described by the following equation:

$$\dot{x} = f(x) + g(x)u \tag{2.1}$$

where:

x : is the system state vector

u : is the control input

The goal of backstepping control is to design a control input u that drives the system state x to a desired reference trajectory x_d. The control input is recursively designed, with each subsystem's control law being based on the error between the current actual state and the desired state of the previous subsystem. The general expression for the backstepping control

law for the i-th subsystem is given by:

$$u_i = -k_i(x_i - x_{i,d}) + u_{i+1} \tag{2.2}$$

Where:

x_i : State of the i-th subsystem

$x_{i,d}$: Desired state of the i-th subsystem

u_{i+1} : Control input from the previous subsystem

k_i : Gain parameter for the i-th subsystem

2.1.2.5 Particle Swarm Optimization

Particle swarm optimization (PSO) is a nature-inspired optimization algorithm that draws its inspiration from the collective behavior of social organisms, such as bird flocking and fish schooling [29]. Developed by Kennedy and Eberhart in 1995. PSO has gained prominence as a powerful heuristic optimization technique [29]. Its popularity stems from its ability to efficiently explore complex search spaces, making it well-suited for solving a wide range of optimization problems. PSO has found applications in a myriad of fields, including engineering, finance, biology, and machine learning [30]. It has been adapted and extended to address various optimization challenges, such as multi-objective optimization, constrained optimization, and dynamic optimization [31].

Table 2.1: PSO Algorithms Mathematical Expressions

Step	**Mathematical Representation**
Initialization	Randomly initialize positions $\mathbf{X}$ and velocities $\mathbf{V}$ of particles.
Fitness Evaluation	Evaluate fitness values $f(\mathbf{X})$ for each particle's current position.
Personal Best Update	Update personal best positions $\mathbf{P}$ if $f(\mathbf{X})$ is better than previous best.
Global Best Update	Update global best position $\mathbf{G}$ as the best $\mathbf{P}$ among all particles.
Velocity and Position Update	Update velocities $\mathbf{V}$ and positions $\mathbf{X}$ using formulas: $\mathbf{V} = \omega\mathbf{V} + \phi_1\mathbf{R}_1(\mathbf{P} - \mathbf{X}) + \phi_2\mathbf{R}_2(\mathbf{G} - \mathbf{X})$ $\mathbf{X} = \mathbf{X} + \mathbf{V}$
Termination	Repeat steps 2-5 until termination condition is met.

2.2 Overview of the Existing Surveys

Tracking Control Based on the Robot Kinematic Model

The trajectory tracking controller introduced in [32] the main emphasis was on developing a reliable control strategy for the precise tracking of nonholonomic vehicles. A Lyapunov function is employed to assess the stability of the controller. It takes inputs such as the vehicle's posture and reference velocities, producing target linear and rotational velocities as outputs. Notably, this controller relied on the vehicle's kinematic model and did not account for the effects of robot dynamics. This characteristic makes it suitable for various mobile robots equipped with dead reckoning capabilities, as it is not robot-specific.

The controller's parameters were determined by linearizing the system's differential equations and identifying conditions for critical damping. However, it should be noted that this approach assumes perfect velocity tracking, which is not typically achievable in practice. Additionally, it overlooks disturbances and requires complete knowledge of the system dynamics [33]. In order to improve the performance of this proposed control algorithm, other controllers which include the vehicle dynamics and can deal with disturbances are proposed.

Adaptive Based Approach

Adaptive control methods aiming at achieving trajectory tracking for wheeled mobile robots have been introduced in [34, 35, 36, 37]. In [34], a dual adaptive dynamic controller is presented specifically for the trajectory tracking of non-holonomic wheeled mobile robots. Notably, this controller is designed entirely in discrete-time, making it suitable for situations where the robot's nonlinear dynamic functions are unknown. To approximate these unknown functions, a Gaussian radial basis function neural network is employed, with real-time stochastic weight estimation. What sets this controller apart from previously published adaptive certainty equivalence controllers is its consideration of estimation uncertainties, which results in improved tracking performance. Extensive validation is performed through realistic simulations and Monte Carlo analysis.

In [36], a novel and straightforward adaptive tracking controller is introduced, primarily based on kinematic models. This controller utilizes an artificial potential field for guiding the wheeled mobile robot. Its key advantages lie in its ease of design, rapid convergence, and adaptability to various non-holonomic mobile robots. The stability of this controller is established through Lyapunov function analysis.

In [37], adaptive control rules operating at the dynamics level are proposed for non-holonomic mobile robots characterized by unknown dynamic parameters. These adaptive

controls, developed using the backstepping technique, address both trajectory tracking and posture stabilization. For trajectory tracking, the controller guarantees the asymptotic convergence of the tracking error to zero. For posture stabilization, the problem is transformed into an equivalent tracking problem through the use of time-varying error feedback, followed by the application of tracking control. This design includes a velocity/acceleration limiter, preventing wheel slippage in the robot.

PID Controller Based Approach

[38] in this paper introduces a particle swarm optimization (PSO) approach to find the best parameters for PID controller. The PID controller is employed to regulate a nonholonomic mobile robot engaged in path tracking. The method involves optimizing two separate PID controllers: one for speed control and another for azimuth control.A good tracking effect is obtained, However the robustness of the controller is poor , and the system is unstable due to external inference.

A Robust Tracking Control for a Nonholonomic Wheeled Mobile Robot using Fuzzy Logic Control is proposed in [[39], [40], [41], and [42]]. In reference to [39], the proposed algorithm focuses on the development of a controller for a 2-DOF Wheeled Mobile Robot, employing a combination of fuzzy logic and genetic algorithms. The controller takes into consideration global inputs, including time-variable reference position and velocity, to estimate the current position of the mobile robot using a dead-reckoning algorithm. This algorithm continuously updates the robot's real-time position by accumulating incremental position data over each sampling period. The primary goal of the tracking controller is to minimize the position error, necessitating the introduction of compensatory velocities along the trajectory. To achieve this, a controller utilizing a fuzzy-genetic algorithm is introduced, where the input variables for two fuzzy logic controllers (FLCs) are the position errors observed during each sampling interval. The FLCs then produce compensation velocities, and genetic algorithms (GAs) are employed to fine-tune the output gain of the fuzzy logic controllers.

In reference to [40], a control algorithm is proposed that operates based on posture errors of a mobile robot. This algorithm generates corrective signals for adjusting the speeds of the left and right motors. The control strategy is founded on a feed-forward velocity profile and the correction signal in the velocity, which is derived from a Fuzzy Logic Controller (FLC) considering posture errors. Simulation results illustrate the effectiveness of this algorithm, showcasing its capability to achieve precise tracking and robust performance even in the presence of uncertainties within the system model. The fuzzy logic controller presented

in [41] utilizes a backstepping strategy to achieve asymptotic stability in positioning and orienting a robot along a specified trajectory. This approach takes into account both the vehicle's kinematics and dynamics. The controller implements a Mamdani inference system, comprising nine if-then rules and employing the centroid of area technique for defuzzification. Input torques and velocities are associated with linguistic variables. Notably, this work distinguishes itself from prior methods that exclusively relied on the kinematic model (the robot's steering system) for addressing tracking control challenges. Instead, it incorporates the robot's dynamic model into its control strategy.

User in reference to [42], the controller presents an innovative tracking method for mobile robots using fuzzy logic control in combination with predictive control . Tracking the trajectory in self-navigating mobile robots often involves nonlinear, time-varying characteristics and is subject to perturbations from additive noise. To mitigate time delays caused by slow sensor response, the algorithm employs predictive control to anticipate the robot's position and orientation. Additionally, fuzzy control is introduced to address the inherent nonlinearities within the system. As described in [42], this paper introduces a novel approach to tracking mobile robots using predictive control in conjunction with FLC. Autonomous mobile robot trajectory tracking usually involves nonlinear time-varying characteristics and is prone to noise perturbations. To counteract delays resulting from sluggish sensor responses, the algorithm utilizes predictive control to forecast the robot's position and orientation. Furthermore, fuzzy control is incorporated to manage the inherent non-linear aspects of the system.The design utilizes triangle and trapezoidal-shaped membership functions, incorporates three fuzzy partitions, and defines nine rules.As described in [42], The controller combines fuzzy logic control with predictive control to present a new method of tracking mobile robots. Trajectory tracking in autonomous mobile robots usually involves nonlinear time-varying characteristics and is prone to noise perturbations. To counteract delays resulting from sluggish sensor responses, the algorithm utilizes predictive control to forecast the robot's position and orientation. Furthermore, fuzzy control is incorporated to manage the inherent non-linear aspects of the system.

2.2.1 The Proposed BFSMC System

Combining Fuzzy Logic Control (FLC), Sliding Mode Control (SMC), and Backstepping Controller in the context of three-wheeled mobile robots offers several advantages:

- Robustness: FLC excels in handling uncertainties and imprecise information, which are common in real-world robotic applications [43]. SMC, known for its robustness against disturbances and uncertainties [21], further enhances the system's ability to cope with unpredictable external factors.
- Precise Tracking: Backstepping control focuses on tracking desired trajectories accurately [44]. When combined with FLC and SMC, it ensures that the mobile robot follows its intended path with high precision.
- Nonlinearity Handling: Mobile robot dynamics are inherently nonlinear. FLC and SMC can effectively deal with nonlinear systems [45]. By integrating backstepping control, the control system becomes even more capable of managing complex nonlinearities.
- Reduced Control Effort: FLC can reduce the control effort required to achieve desired performance [46]. When integrated with SMC and backstepping, it leads to efficient control with minimal energy consumption.

2.2.2 Control Perspective of Three Wheeled Mobile Robot

Three wheeled mobile robots (TWMRs) represent a fascinating and versatile class of robotic systems that have gained significant attention in recent years.TWMRs are characterized by their ability to move swiftly and navigate through tight spaces, making them well-suited for a wide range of applications. The design and control of TWMRs present intriguing challenges and opportunities, motivating extensive research in the field of robotics [47]. The concept of three-wheeled mobile robots has its roots in the principles of differential drive systems, where the robot moves by independently controlling the speeds of its two drive wheels. This differential drive configuration imparts nonholonomic constraints on the robot's movement, resulting in a kinematic system that requires careful control strategies for precise trajectory tracking. Additionally, TWMRs often operate in environments with uncertainties, sensor noise, and parameter variations, further emphasizing the need for robust control solutions. Three-wheeled mobile robots (TWMRs) are a popular choice in robotics

due to their maneuverability and efficiency. In this configuration, two wheels serve as the driving wheels, while one wheel acts as a caster wheel, typically unpowered and free to swivel [48]. This arrangement offers simplicity in control and mechanical design. The driving wheels are responsible for generating motion, while the caster wheel provides stability and ease of steering [49].

Additionally, the control of TWMRs is well-established, with various control algorithms designed to exploit their unique characteristics. For instance, differential drive control, a common approach for TWMRs, enables precise control over the individual driving wheel speeds, allowing for accurate maneuvering and trajectory tracking [50]. This control perspective emphasizes the role of the two driving wheels in determining the robot's motion, while the caster wheel plays a crucial role in ensuring stability during movement.

Chapter 3

Mathematical Modeling of Three Wheeled Mobile Robot

3.1 Introduction

In this chapter, a comprehensive examination of the mathematical model of a three wheeled mobile robot (TWMRs) is undertaken. Three key aspects are focused on: the kinematics model, which describes the robot's motion without considering forces and torques; the dynamics model, which accounts for the forces and torques influencing the robot's behavior; and the actuator model, which characterizes the robot's control system. By integrating these three models using the versatile MATLAB software, the derived model is carefully validated and verified, ensuring its accuracy and effectiveness in representing the behavior and performance of the TWMR . This chapter presents the analysis and verification of the general mathematical model of TWMR.

3.2 Tricycle Wheeled Mobile Robot

A tricycle mobile robot is a type of wheeled robot that utilizes three wheels for locomotion. In this configuration, the front wheel, also known as the caster wheel, plays a critical role in balancing the robot the two rear wheel known as driving wheel their task is driving the robot. The caster wheel is designed to pivot freely in any direction, allowing it to rotate around its vertical axis. This characteristic enables the robot to maintain stability and balance during its movement. As the robot navigates, the caster wheel helps prevent tipping and allows for smooth turns and maneuverability. The tricycle mobile robot with a front caster wheel is

commonly employed in various applications, including indoor navigation, material handling, and surveillance, due to its simplicity, stability, and ease of control.

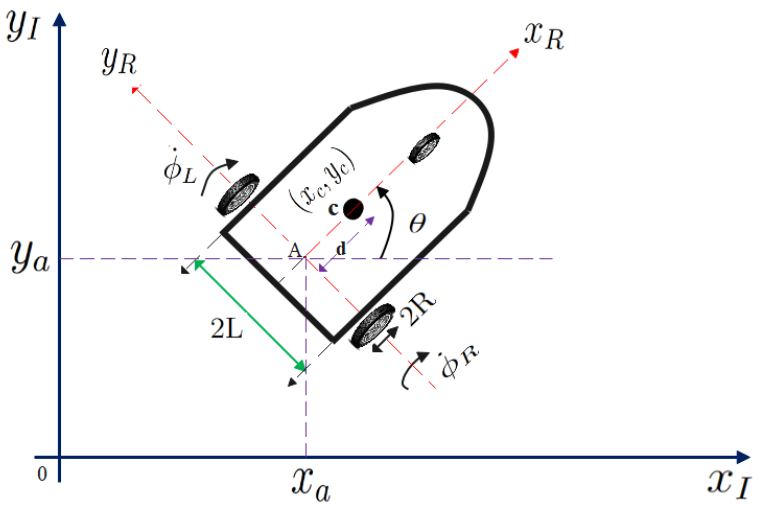


Figure 3.1: The Global Reference Frame and the Robot Reference Frame of TWMR.

In this thesis, the following notations are utilized:

A: Indicates the intersection point of the axis of symmetry and the axis of the driving wheels.

c: Represents the center of mass of the platform.

d: Signifies the distance between the center of mass and the axis of the driving wheels in the x-direction.

L: Signifies the separation between each driving wheel and the axis of symmetry of the robot in the y-direction.

$\dot{\phi}_R$: Indicates the rotational velocity of the right wheel.

R: Stands for the radius of each driving wheel.

$\dot{\phi}_L$: Represents the rotational velocity of the left wheel.

v: Corresponds to the translational velocity of the platform in the local frame.

w: Denotes the rotational velocity of the platform in both the local and global frames.

3.2.1 Coordinate Frames Transformation

The main function of the coordinate systems is to represent the position of the robot. To describe the motion of a mobile robot, it is must to specify and arrange the reference frames. If the robot viewed as a rigid body, there are two basic frames. They are the world-fixed global reference frame and the robot-fixed local reference frame.

Inertial Reference Frame:- This coordinate system is also known as the global frame,

which remains stationary within the environment or plane where it operates. TWMR operates, and it is symbolized as (x_I, y_I).

$$q_I = \begin{bmatrix} x_I & y_I & \theta_I \end{bmatrix}^T \tag{3.1}$$

Robot Reference Frame:- This coordinate system is also recognized as the local frame, which is affixed to the TWMR and travels along with it. This frame is represented as (x_R, y_R).

$$q_R = \begin{bmatrix} x_R & y_R & \theta_R \end{bmatrix}^T \tag{3.2}$$

With the definitions provided earlier, it becomes possible to convert elements expressed in the local frame into the global frame and vice versa. For instance, it allows for the conversion of computed motion from the global frame into the robot's local frame, Additionally, the mapping of obstacles detected by the robot in the global reference frame is facilitated by the orthogonal rotation matrix, which functions based on the robot's current pose.

Orthogonal Rotation Matrix:- Used to map robot reference frame and global reference frame. Orthogonal rotation matrix($R(\theta)$) given as :-

$$R(\theta) = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Robot reference frame and inertial reference frame related by this two transformations equation as shown below.

$$\begin{aligned} \dot{q_I} &= R(\theta)^{-}\dot{q_R} \\ \dot{q_R} &= R(\theta)\dot{q_I} \end{aligned} \tag{3.4}$$

Equation (3.4) defines relations between velocities in the global and the local reference frame.

3.2.2 Kinematics Modeling of Three Wheeled Mobile Robot

In the field of mobile robotics, kinematics plays a crucial role in recognizing and measuring the limitations imposed by the design of the robot. This design, treating the robot as a rigid body with the freedom to move in space, is characterized by six degrees of freedom represented as $[x, y, z, \theta, \phi, \psi]^T$. Nonetheless, due to the influence of a gravitational field that restricts the robot's movement to a Euclidean plane, the variables x and y are indicative of the robot's position, while θ defines its orientation within that plane. This two-dimensional

space, where the robot maneuvers, is commonly known as the C-space or configuration space. TWMRs fall into a class of mechanical systems distinguished by kinematic constraints that are non-integrable, meaning they cannot be eliminated from the model equations.[51].

Kinematics Constraints

- Each wheel remain perpendicular about its plane.
- No slipping, skidding or sliding.
- There is only one contact point between plane and wheel.
- Pure rolling motion without slipping occurs.
- Friction is absent for rotation around contact points.
- The wheels are not deformable.
- The wheels are connected by a rigid frame (chassis).
- Movement on a horizontal plane.

Absence of lateral slip motion: This limitation implies that the robot is capable of curved movements (forward and backward) but does not have the ability to move sideways.Considering the previously mentioned assumptions regarding wheel motion, the robot will be constrained by a specific type known as a nonholonomic constraint. This constraint restricts the potential velocities of the robot, implying that it can move in specific directions (e.g., forward and backward) but not in others (e.g., sideward motion), as depicted in Figure 3.2. $\dot{y}_c$ and $\dot{x}_c$ are the robot velocity components in the inertial frame. This constraint means that the velocity of the robot center point will be in the direction of the axis of symmetry and the motion in the orthogonal plane will be zero.

$$\dot{y}_c \cos\theta - x_c \sin\theta - \dot{\theta}_a = 0 \tag{3.5}$$

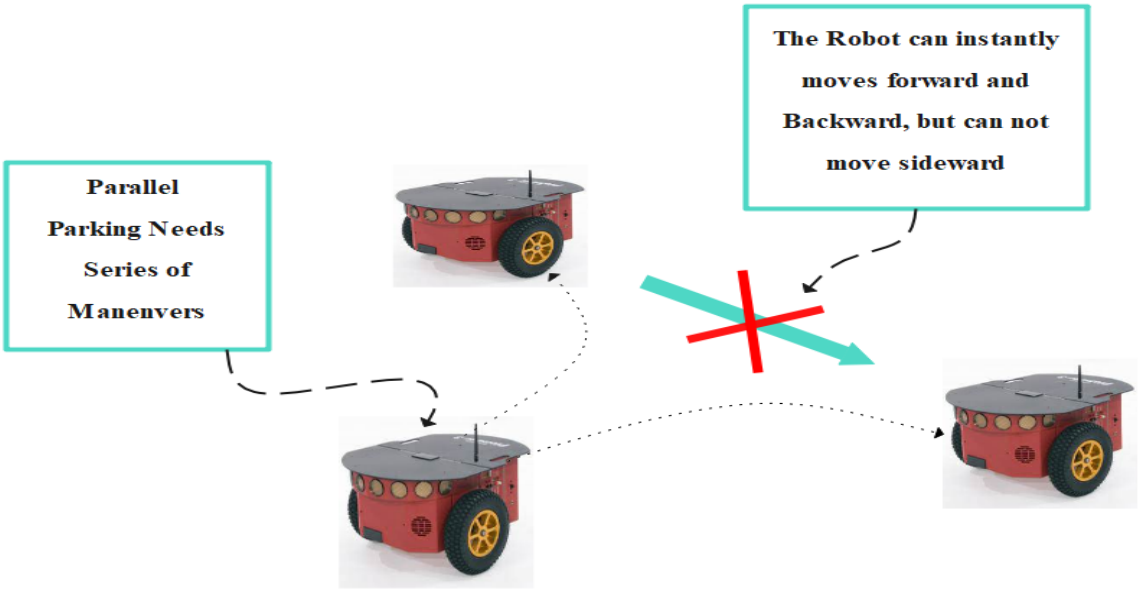


Figure 3.2: Nonholonomic Constraint on the Robot Motion .

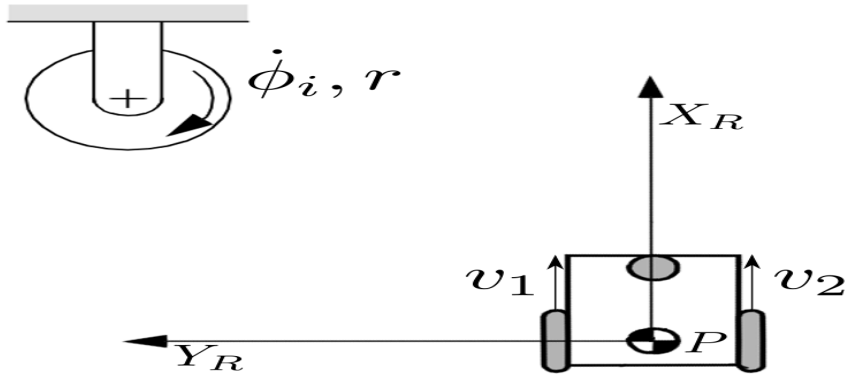


Figure 3.3: Wheel Velocities and Inertial frame.

Constraint of pure rolling: The pure rolling constraint indicates that each wheel sustains a single contact point P with the ground, as illustrated in figure.3.3.

$$\begin{aligned} \dot{x}_c \cos\theta + \dot{y}_c \sin\theta + L\dot{\theta} &= R_a \dot{\phi}_R \\ \dot{x}_c \cos\theta + \dot{y}_c \sin\theta - L\dot{\theta} &= R_a \dot{\phi}_L \end{aligned} \tag{3.6}$$

This constraint shows that the driving wheels do not slip.

Forward Kinematics:- can be expressed as the following function.

$$\dot{q}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\phi}_R, \dot{\phi}_L, L, R_a, \theta) \tag{3.7}$$

Since the wheels contribute to the robot's mobility individually, their contribution separately will be examined. [52].

$$\begin{aligned} v_i &= \tfrac{\dot{\phi}_i r}{2} \\ \omega_i &= \tfrac{\dot{\phi}_i r}{2l} \\ \text{where } i &= \{1, 2\} \end{aligned} \tag{3.8}$$

The TWMR's linear velocity in the robot frame is the mean of the linear velocities of its two wheels.

$$v = \frac{v_R + v_L}{2} = R\frac{(\dot{\phi}_R + \dot{\phi}_L)}{2} \tag{3.9}$$

The angular velocity of the TWMR is given as

$$\omega = \frac{v_R - v_L}{2L} = R\frac{(\dot{\phi}_R - \dot{\phi}_L)}{2L} \tag{3.10}$$

The velocities of the TWMRs in the robot frame can be expressed in terms of the velocities of the center-point A in the robot frame as follows:-

$$\begin{aligned} \dot{x}_R = v &= \frac{v_R + v_L}{2} = R\frac{(\dot{\phi}_R + \dot{\phi}_L)}{2} \\ \dot{y}_R &= 0 \\ \dot{\theta} = \omega &= \frac{v_R - v_L}{2L} = R\frac{(\dot{\phi}_R - \dot{\phi}_L)}{2L} \end{aligned} \tag{3.11}$$

Expressing equation (3.11)in matrix format :-

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \tag{3.12}$$

To obtain wheel velocities in inertial frames multiplying velocities of robot frame as indicated in equation (3.3) by orthogonal transformation matrix as indicated in equation(3.3) . TWMRs velocities in the inertial frame can be obtained as below equation (3.13)

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{2L} & \frac{R}{-2L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \tag{3.13}$$

After multiplication of above equation (3.13) equation (3.14) resulted which represents the forward kinematics model of TWMR:

$$\dot{q}_I = \begin{bmatrix} \frac{R}{2}\cos(\theta) & \frac{R}{2}\cos(\theta) \\ \frac{R}{2}\sin(\theta) & \frac{R}{2}\sin(\theta) \\ \frac{R}{2L} & \frac{R}{-2L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \tag{3.14}$$

Another alternative representation of the kinematic model can be derived by expressing the TWMR velocities in relation to the linear and angular velocities of the TWMR in the robot frame, as illustrated in the equation (3.15).

$$\dot{q}_I = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{3.15}$$

Since the relationship between spin speeds and velocities are known by using forward kinematics modeling,What about the position and orientation of the robot in the global frame? By integrating equation (3.14) it is possible to obtain pose of WMR in global reference frame:-

$$\dot{q}_I = \int_0^t \begin{bmatrix} \frac{R}{2}\cos(\theta) & \frac{R}{2}\cos(\theta) \\ \frac{R}{2}\sin(\theta) & \frac{R}{2}\sin(\theta) \\ \frac{R}{2L} & \frac{R}{-2L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} dt \tag{3.16}$$

$$\begin{aligned} x(t) &= \frac{R}{2}\int_0^t (\dot{\phi_R}(t) + \dot{\phi_L}(t))cos(\theta(t))dt + x(0) \\ y(t) &= \frac{R}{2}\int_0^t (\dot{\phi_R}(t) + \dot{\phi_L}(t))sin(\theta(t))dt + y(0) \\ \theta(t) &= \frac{R}{2L}\int_0^t (\dot{\phi_R}(t) - \dot{\phi_L}(t))dt + \theta(0) \end{aligned} \tag{3.17}$$

3.2.3 Dynamics Modeling of Three Wheeled Mobile Robot

The inputs of a kinematic model do not directly correspond to physical inputs such as forces or torques. In other words, the dynamics of the system are disregarded when dealing with a kinematic model. Therefore, it is essential to derive the dynamic model and examine its characteristics.

There are two methods for obtaining dynamic models.

- **The Newton-Euler approach**: Which is based on direct interpretations of Newton's second law of motion, describes the system by accounting for all the forces and momentum acting on it.
- **The Lagrange approach**: Alternatively, it employs the concepts of work and energy to indirectly derive the equations of motion. The Lagrange technique is used in this case because of its more systematic nature and automated removal of work-less and constraining forces. A system's Lagrangian is defined as the difference between its kinetic and potential energy[52].

The Lagrange equation for a non-holonomic system is formulated as:-

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}})^T - (\frac{\partial L}{\partial q})^T = S(q) + V(q)\lambda \tag{3.18}$$

The Lagrange equation in the case of a non-holonomic system is given as:

$$S(q)\dot{q} = V(q)$$

where $S(q)$ is a matrix of size $(n \times m)$, mapping the external inputs $(m = nk)$ denoted by τ to generalized forces, and $V(q)$ represents the transpose of $V^T(q)$.

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q) = 1/2\dot{q}^T I(q)\dot{q}U(q) \tag{3.19}$$

Where, $T(q, \dot{q})$ and $U(q)$ are the kinetic and potential energy, respectively, and $I(q)$ is the inertia matrix of the mechanical system.

A non-holonomic TWMR with n generalized coordinates $(q_1, q_2, ..., q_n)$ and subject to m constraints can be described by the following equations of motion:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - \Lambda^T(q)\lambda \tag{3.20}$$

where:

$M(q)$: an nxn symmetric positive definite inertia matrix.

$V(q,\dot{q})$: is the centripetal and coriolis matrix.

$G(q)$: is the gravitational vector.

τ_d : is the vector of bounded unknown disturbances.

$B(q)$: is the input matrix.

τ : is the input vector.

$\Lambda^T(q)$: is the matrix associated with the kinematic constraints.

λ : is the Lagrange multipliers vector.

The initial step in formulating the dynamic model through the Lagrange approach involves determining the kinetic and potential energies influencing the TWMR's motion. As the TWMR moves within the (x_I, y_I) plane, the potential energy is assumed to be zero. For the TWMR, the chosen generalized coordinates are:-

$$\dot{q} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} & \dot{\phi}_R & \dot{\phi}_L \end{bmatrix}^T \tag{3.21}$$

The parameter matrices and vector in equation (3.20) are given by[53]:

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \Lambda^T(q) = \begin{bmatrix} -sin(\theta) \\ cos(\theta) \\ 0 \end{bmatrix} B(q) = \frac{1}{R} \begin{bmatrix} cos(\theta) & cos(\theta) \\ sin(\theta) & sin(\theta) \\ L & L \end{bmatrix} \tag{3.22}$$

$$\tau = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix}$$

$$\lambda = -m\dot{\theta}(\dot{x}_c cos(\theta) + \dot{y}_c sin(\theta)) \tag{3.23}$$

where m represents the mass of the TWMR, I is the moment of inertia of the TWMR about its center. The torque control inputs τ_R and τ_L are generated by the right and left DC motors, respectively. For control purposes, the constraint term $\Lambda^T(q)\lambda$ in equation (3.20) needs to be eliminated through an appropriate transformation.

The primary objective is to eliminate the constraint term $\Lambda^T(q)\lambda$ in equation (3.20), given that the Lagrange multipliers λ_i are unknown. This is initially achieved by introducing the reduced vector as follows.

$$\eta = \begin{bmatrix} \dot{\phi}_R & \dot{\phi}_L \end{bmatrix}^T \tag{3.24}$$

Subsequently, the generalized coordinates velocities are expressed in terms of the forward kinematics equation using the forward kinematic model equation (3.14).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = \frac{1}{2} \begin{bmatrix} Rcos(\theta) & Rcos(\theta) \\ Rsin(\theta) & Rsin(\theta) \\ \frac{R}{L} & \frac{-R}{L} \\ 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \tag{3.25}$$

$$\dot{q} = S(q)\eta \tag{3.26}$$

This can be expressed as

$$\dot{q} = S(q)\Lambda^T(q) = 0 \tag{3.27}$$

It can be verified that the transformation matrix $S(q)$ is in the null space of the constraint matrix $\Lambda(q)$.

This can be written in the form of:-

$$S^T(q)\Lambda^T(q) = 0 \tag{3.28}$$

Next taking the time derivative of equation (3.26) yields:-

$$\ddot{q} = \dot{S}(q)\eta + S(q)\dot{\eta} \tag{3.29}$$

Substituting equation (3.26) and equation (3.29) into main equation (3.20) yields:

$$M(q)[\dot{S}(q)\eta + S(q)\dot{\eta}] + V(q,\dot{q})[S(q)\eta] + F(\dot{q})[S(q)\eta] = B(q)\tau - \Lambda^T(q)\lambda \tag{3.30}$$

Next, rearranging the equation and multiplying both sides of above Equation by $S^T(q)$ as result below equation obtained.

$$\begin{aligned} S^T(q)M(q)S(q)\dot{\eta} + S^T(q)[M(q)\dot{S}(q) + V(q,\dot{q})S(q)]\eta + F(\dot{q})[S(q)\eta] = S^T(q)B(q)\tau \\ -S^T(q)\Lambda^T(q)\lambda \end{aligned} \tag{3.31}$$

where the last term is zero by definition. Defining the new matrices and reducing the dynamic model results in the following equation.

$$\hat{M(q)}\dot{\eta} + \hat{V}(q,\dot{q})\eta + \hat{F}(\dot{q}) = \hat{B}(q)\tau \tag{3.32}$$

Equation (3.32) written as when kinematics model expressed in terms linear and angular velocity rather than left and right wheel speed as below equation (3.33)

$$\hat{M}(q)\dot{v} + \hat{V}(q,\dot{q})v + \hat{F}(\dot{q}) = \hat{B}(q)(\tau + \hat{\tau}_d) \tag{3.33}$$

$$\begin{aligned} \hat{M} &= S^T M S \in R^{2\times 2} \\ \hat{F} &= S^T F \in R^{2\times 1} \\ \hat{\tau_d} &= S^T \tau_d \\ \hat{B} &= S^T B \\ \hat{V}_m &= S^T(M\dot{S} + V_m S) \in R^{2\times 2} \end{aligned} \tag{3.34}$$

- In view of the distance between center of mass and the coordinate center of WMR is zero in this case $\hat{V}(q,\dot{q}) = 0$.
- By considering surface friction F($\dot{q}$) and the disturbance torque(τ_d) as the external disturbances. The dynamics model equation (3.33) can be reduced into nominal dynamic model as below equation:

$$\hat{M}(q)\dot{v} = \hat{B}(q)\tau \tag{3.35}$$

From equation (3.35) $\dot{v}$ becomes as

$$\dot{v} = \hat{M}^{-1}(q)\hat{B}(q)\tau \tag{3.36}$$

Where as $\hat{M}(q)$ and $\hat{B}(q)$ given as

$$\hat{M}(q) = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \qquad \hat{B}(q) = \frac{1}{R}\begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix} \tag{3.37}$$

Let for simplicity $E = \hat{M}^{-1}(q)\hat{B}(q)$, nominal dynamics model becomes

$$\dot{v} = E\tau \tag{3.38}$$

The dynamic equation in presence of uncertainties and disturbances given as below equation.

$$\dot{v} = E\tau + \tau_d(t) = \hat{E}\tau(t) + \Delta E\tau(t) + \tau_d(t) \tag{3.39}$$

$\hat{E}$ represents the nominal component of the system matrix E, influenced by the parameters of WMRs, namely m, r, I, and b. ΔE signifies the uncertainties associated with the system matrix E. τ_d stands for the vector of external disturbances.

3.2.4 Actuator Dynamics Modeling

In general cases, the driving system of mobile robots is based on an armature-controlled DC motors which are considered as servo actuators. These actuators are responsible for instructing and propelling the mobile robot's two wheels through torque control inputs. The control inputs for the motors are the armature voltages v_a, with the field circuit conditions being held constant.

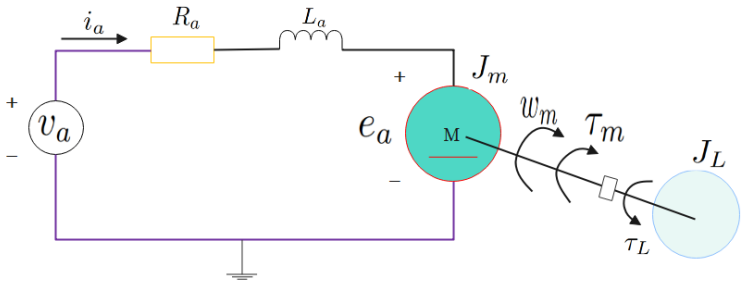


Figure 3.4: DC Motor Architecture

The given below equations represent the characteristics of the armature circuit for a permanent-magnet DC motor.

$$v_a = R_a i_a + L_a \frac{di_a}{dt} + e_a \tag{3.40}$$

$$\begin{aligned} e_a &= k_b w_m \\ \tau_m &= k_t i_a \\ \tau &= N\tau_m \end{aligned} \tag{3.41}$$

where i_a represents the armature current, (R_a, L_a) denote the resistance and inductance of the armature winding, e_a is the back emf, w_m stands for the rotor angular speed, T_m represents the motor torque, and (k_t, k_b) are the torque constant and back emf constant, respectively. N is the gear ratio, and τ is the output torque applied to the wheel. Given

that the motors in the TWMR are mechanically connected to the robot wheels through the gears, the mechanical equations of motion for the motors are directly related to the mechanical dynamics of the TWMR. As a result, each DC motor contributes to:

$$\begin{aligned} w_{m_R} &= N\dot{\phi}_R \\ w_{m_L} &= N\dot{\phi}_L \end{aligned} \tag{3.42}$$

By applying newtons second law the mechanical equation of motor becomes as follows:

$$J_m \frac{d\omega_m}{dt} = T_m - T_L - b_m\omega_m \tag{3.43}$$

Where J_m is rotor inertia , b_m is equivalent damping constant of the rotor, T_m is the rotor torque , and T_L is the load torque of the motor rotor. The mechanical and electrical equation of actuator can be written us by applying Laplace transform to equation (3.43) and equation (3.40).

$$\begin{aligned} I_a(s) &= \frac{v_a - e_a}{L_a s + R_a} \\ \omega(s) &= \frac{T_m - T_L}{sJ_m + b_m} \end{aligned} \tag{3.44}$$

Calculating torque from equation (3.40) by assuming static model below equation (3.45) resulted in:

$$v_a = R_a i_a + k_b\omega_m \tag{3.45}$$

Calculating armature current(i_a) from equation (3.45) and calculating torque ($\tau = k_t i_a$). Finally torques of TWMRs can be given by below equation (3.46).

$$\tau = k_t \frac{v_a - k_b w_m}{R_a} \tag{3.46}$$

Torque for left and right wheel of mobile robot can be given as:-

$$\begin{aligned} \tau_R &= k_t \frac{V_R - k_b w_R}{R_a} \\ \tau_L &= k_t \frac{V_L - k_b w_L}{R_a} \end{aligned} \tag{3.47}$$

Table 3.1: System Parameter Values[54]

SI. No	Parameter	Value	Unit	Description
1	I	5	$kg \cdot m^2$	The mass moment of inertia about the center of mass
2	b_m	0.5	Nms/rad	Viscous Friction
3	R_a	8	ohm	Resistance of the DC motor Armature
4	L_a	0	H	Motor Inductance
5	k_m	0.35	Nm/A	Motor Torque Constant
6	k_b	0.35	Vs/rad	Motor Back-EMF Constant
7	m	10	kg	The Robot Weight
8	L	20	cm	The distance between the Drive Wheel and the axis of symmetry
9	A	5	cm	The distance between the center of mass and the drive Wheel axis
10	R	7.5	cm	Wheel Radius

3.2.5 Model Verification for Three Wheeled Mobile Robot

Under this section the TWMR model is verified by applying voltages for right and left wheel of mobile robot. By applying equal , different , opposite and equal left wheel voltage(V_L) and right wheel(V_R) voltage. The correctness of the model analyzed.

1. When $V_R = V_L$, the robot moves straight forward in a linear motion. The radius of curvature, R, becomes infinite, resulting in no rotation, which means ω is zero as shown in figure 3.5a.

2. When different values of V_R and V_L are applied, the mobile robot doesn't move in a straight line but instead follows a curved path around a point located at a distance R from CR. This situation can be observed in figure 3.5b and figure 3.5c, where it turns around one of the wheels, causing changes in both the robot's position and orientation.

3. In the case of $V_R = -V_L$, R equals 0, resulting in the robot rotating about the midpoint of the wheel axis. This behavior is illustrated in figure 3.5d.

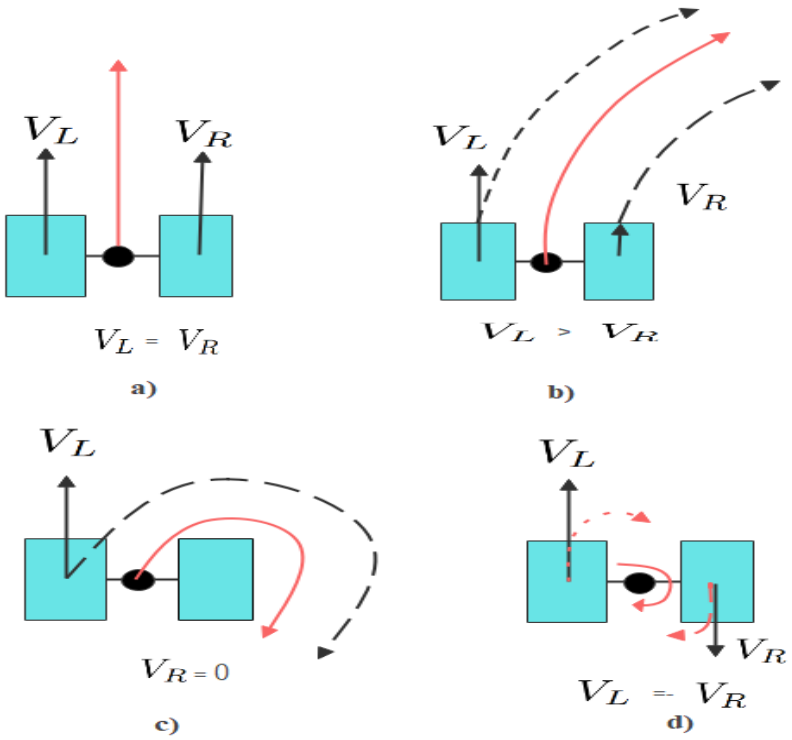


Figure 3.5: Various Modes of Motion for TWMRs[55]

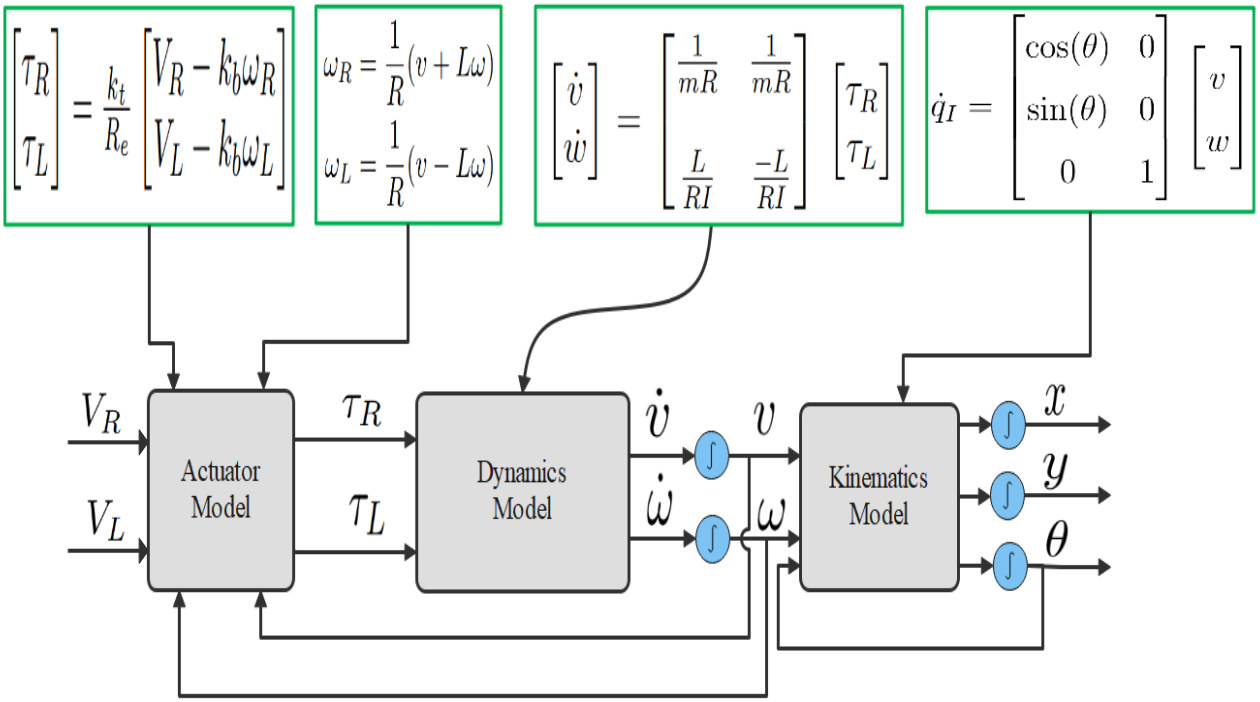


Figure 3.6: Mathematical Model of TWMR

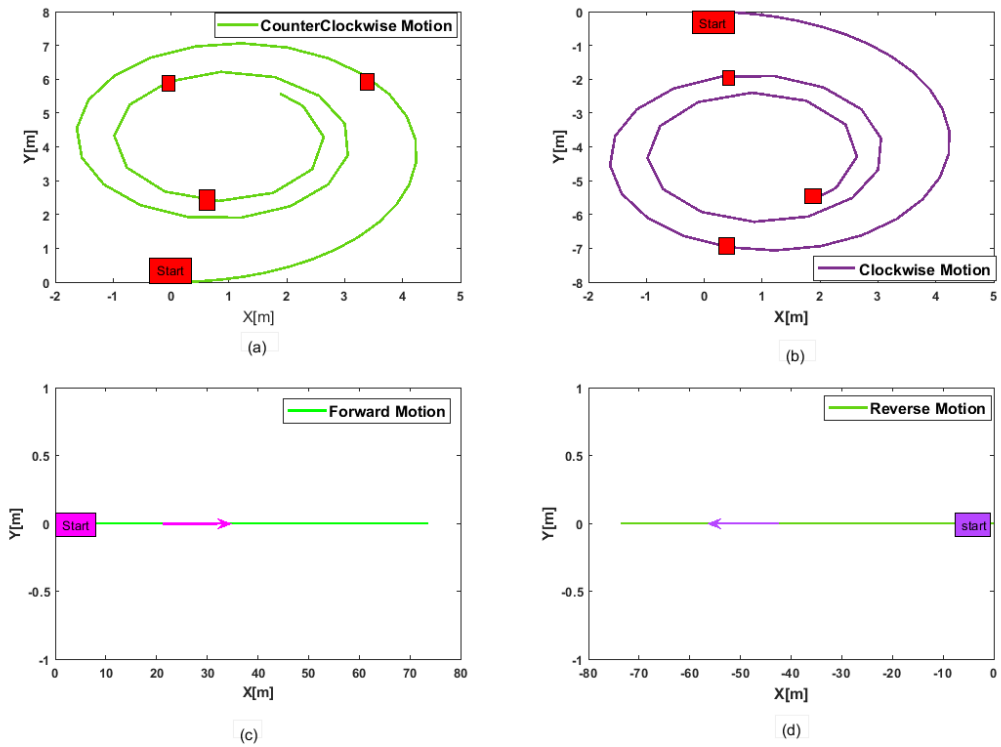


Figure 3.7: Simulation Result of Model Verification

Simulation Result Description

The model of TWMR verified or checked by using MATLAB/Simulink Software's by using parameter values in table 3.1. The behavior of the wheeled mobile robot was studied under different voltage configurations for its right and left wheels. When the voltage applied to the right wheel was lower than that of the left wheel, the robot exhibited clockwise motion, as depicted in figure 3.7a . Conversely, with a higher voltage on the right wheel compared to the left wheel, the robot moved in the counterclockwise direction, as shown in figure 3.7b. Furthermore, when equal voltages were applied to both the left and right wheels, illustrated in figure 3.7c, the robot moved forward. On the other hand, when equal-magnitude but opposite voltages were applied to the robot's wheels, it traveled backward, as demonstrated in figure 3.7d. Based on the observed results, it can be confidently concluded that the derived model accurately represents the behavior of the wheeled mobile robot under different control inputs, confirming the validity of the model

Chapter 4

Controller Design

4.1 Introduction

In this chapter, the controller design for both the inner and outer loops is presented, using the kinematics and dynamics model of three wheeled mobile robot (TWMR). The design involves the design of a backstepping controller for the outer loop and a fuzzy sliding mode controller (FSMC) for the inner loop. Additionally, a backstepping sliding mode controller (BSMC) is introduced for comparison purpose. An extended kalman filter (EKF) is also formulated to improve the overall system efficiency by reducing sensor noise. Finally, particle swarm optimization (PSO) techniques are employed to fine-tune the controller gains of both BFSMC and BSMC.

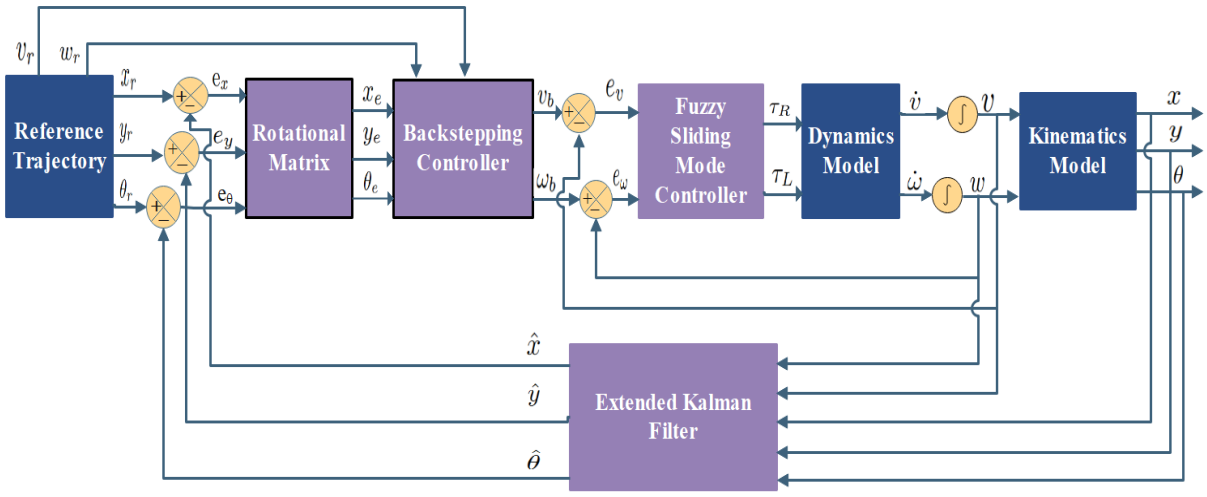


Figure 4.1: General Block Diagram of the System

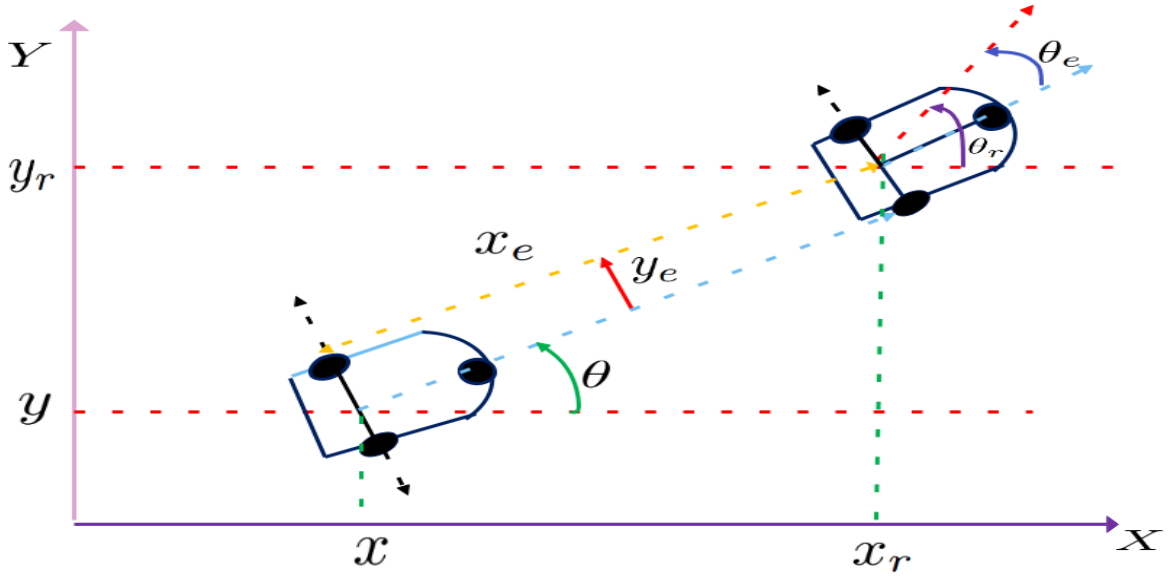


Figure 4.2: Posture Error Description

4.2 Backstepping Controller Design

To simplify the derivation of the backstepping controller, assuming that the EKF is replaced by a unity gain. However, it's important to note that in the simulation, the EKF will be reintroduced into the simulink block. The movement of mobile robot begin from position $p = (x, y, \theta)$ to desired position $p_r = (x_r, y_r, \theta_r)$ [56]. Figure 4.2 shows that movement of mobile robot.

The objective of backstepping controller is to provide the reference robot trajectory.

$$x = x_r(t), y = y_r(t), \theta = \theta_r(t) \tag{4.1}$$

Using forward kinematics model of TWMR from equation(3.15).

$$\begin{aligned} \dot{x} &= vcos(\theta) \\ \dot{y} &= vsin(\theta) \\ \dot{\theta} &= \omega \end{aligned} \tag{4.2}$$

where $x = x_r(t), y = y_r(t), \theta = \theta_r(t)$ are continuously differentiable function such that the following holds.

$$\begin{aligned} \dot{x}_r(t) &= v_r(t)cos(\theta_r(t)) \\ \dot{y}_r(t) &= v_r(t)sin(\theta_r(t)) \\ \dot{\theta}_r(t) &= \omega_r(t) \end{aligned} \tag{4.3}$$

Tracking error can be expressed as

$$q_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{4.4}$$

where x_e , y_e and θ_e is the longitudinal position, lateral position and angle errors between reference position and the actual position transformed into the robot frame[56].

$$x_e = cos(\theta)(x_r - x) + sin(\theta)(y_r - y) \tag{4.5}$$

$$y_e = -sin(\theta)(x_r - x) + cos(\theta)(y_r - y) \tag{4.6}$$

$$\theta_e = \theta_r - \theta \tag{4.7}$$

Taking the time derivative of equation (4.5)

$$\begin{aligned} \dot{x}_e &= -\dot{\theta}sin(\theta)(x_r - x) + (\dot{x}_r - \dot{x})cos(\theta) + \dot{\theta}cos(\theta)(y_r - y) + (\dot{y}_r - \dot{y})sin(\theta) \\ &= \dot{\theta}(cos(\theta)(y_r - y) - sin(x_r - x)) + (\dot{x}_r - \dot{x})cos(\theta) + (y_r - y)sin(\theta) \\ &= \dot{\theta}y_e + (x_r - x)cos(\theta) + (y_r - y)sin(\theta) \\ &= \dot{\theta}y_e + cos(\theta)(v_r cos(\theta_r) - vcos(\theta)) + sin(\theta)(v_r sin(\theta_r) - vsin(\theta)) \\ &= \dot{\theta}y_e + v_r cos(\theta_r)cos(\theta) - vcos^2(\theta) + v_r sin(\theta_r)sin(\theta) - vsin^2(\theta) \end{aligned} \tag{4.8}$$

Rearranging above equation and using trigonometric property finally equation (4.9) resulted in:

$$\dot{x}_e = \dot{\theta}y_e + v_r cos(\theta_e) - v \tag{4.9}$$

Differentiating equation (4.6) hence $\dot{y}_e$ becomes.

$$\begin{aligned} \dot{y}_e &= -\dot{\theta}(cos(\theta)(x_r - x) + sin(y_r - y)) + (\dot{x}_r - \dot{x})(-sin(\theta) + (\dot{y}_r - \dot{y})cos(\theta) \\ &= -\dot{\theta}x_e + (v_r cos(\theta) - vcos(\theta)) * -sin(\theta) + (v_r sin(\theta_r) - vsin(\theta)) * cos(\theta) \\ &= \dot{\theta}x_e + v_r(sin(\theta_r)cos(\theta) - cos(\theta_r)sin(\theta)) + v(cos(\theta)sin(\theta) - sin(\theta)cos(\theta)) \\ &= -\dot{\theta}x_e + v_r sin(\theta_e) \end{aligned} \tag{4.10}$$

Finally $\dot{y}_e$ becomes:

$$\dot{y}_e = -\dot{\theta}x_e + v_r sin(\theta_e) \tag{4.11}$$

By differentiating equation (4.7) $\dot{\theta}_e$ becomes:

$$\dot{\theta}_e = \dot{\theta}_r - \dot{\theta} \tag{4.12}$$

After derivation trajectory tracking error becomes as equation (4.13):

$$\begin{aligned} \dot{x}_e &= \dot{\theta} y_e + v_r cos(\theta_e) - v \\ \dot{y}_e &= -\dot{\theta} x_e + v_r sin(\theta_e) \\ \dot{\theta}_e &= \dot{\theta}_r - \dot{\theta} \end{aligned} \tag{4.13}$$

In order to derive backstepping controller for kinematics model of TWMR. Firstly selecting the Lyapunov candidate function as follows which is positive definite:

$$V = \frac{1}{2}(x_e^2 + y_e^2) + \frac{1 - cos(\theta_e)}{k_2} \tag{4.14}$$

To show Lyapunov candidate function V is positive definite(PD) substituting the value of (x_e, y_e, θ_e)=0 into equation (4.14) it is clear $V \geq 0$ and $V = 0$, therefore the above V function is a positive definite function.

Differentiating equation (4.14)

$$\dot{V} = x_e \dot{x}_e + y_e \dot{y}_e + \frac{\dot{\theta}_e sin(\theta_e)}{k_2} \tag{4.15}$$

Substituting equation (4.13) and (4.5),(4.6),(4.7) into equation (4.15) gives equation (4.16).

$$\dot{V} = (\dot{\theta} y_e + v_r cos(\theta_e) - v_1) x_e + (-\dot{\theta} x_e v_r sin(\theta_e) y_e + \frac{(\dot{\theta}_r - \dot{\theta}) sin(\theta_e)}{k_2} \tag{4.16}$$

$$\dot{V} = (u_2 y_e + v_r cos(\theta_e) - u_1) x_e + (-\dot{\theta} x_e v_r sin(\theta_e) y_e + \frac{(\dot{\theta}_r - u_2) sin(\theta_e)}{k_2} \tag{4.17}$$

Choosing control signal(u) =$[u_1 \quad u_2]^T$ from equation (4.16) in a way the derivative of Lyapunov candidate function is less than or equal to zero ($\dot{V} \leq 0$).

$$\begin{aligned} u_1 &= v_r cos(\theta_e) + k_1 x_e \\ u_2 &= \dot{\theta}_r + v_r (k_2 ye + k_3 sin(\theta_e)) \end{aligned} \tag{4.18}$$

Since $u = [u_1 \quad u_2]^T = [v \quad w]^T$

Backstepping controller for kinematics model or outer loop controller for the system is given as:-

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} v_r cos(\theta_e) + k_1 x_e \\ \theta_r + v_r(k_2 y_e + k_3 sin(\theta_e)) \end{bmatrix} \tag{4.19}$$

4.2.1 Stability Analysis for Backstepping Controller Using Lyapunov Stability Theorm

The condition for the system to be stable the candidate Lyapunov function V must be positive definite and the derivative of Lyapunov function V must be negative definite. Equation (4.14) is positive definite , to check $\dot{V} \leq 0$ substituting the value of v and w from equation (4.19) into equation (4.16).

$$\dot{V} = -k_1 x_e^2 - \frac{k_3}{k_2} sin^2(\theta_e) \tag{4.20}$$

For $\dot{V} \leq 0$ the parameter k_1, k_2 and k_3 must be greater than or equal to zero , mathematically $\dot{V} \leq 0$ when $k_1 \geq 0, k_2 \geq 0$ and $k_3 \geq 0$

The designed controller is stable for positive value of k_1, k_2 and k_3.

4.3 Sliding Mode Controller Design

In this section SMC method employed to enable dynamic tracking controller auxiliary tracking error to zero.

- Auxiliary velocity tracking error given as:-

$$e_c(t) = [e_v \quad e_w]^T \tag{4.21}$$

$$\dot{e}_c(t) = [\dot{e}_v \quad \dot{e}_\omega]^T \tag{4.22}$$

where as $e_v = [v_\beta - v]$ and $e_\omega = [\omega_b - \omega]$

Steps in Designing Sliding Mode Control for Dynamics of the System

- Selecting PI sliding surface.

$$s(t) = [s_1(t) \quad s_2(t)]^T = e_c(t) + \beta \int e_c(t)dt \tag{4.23}$$

where β is positive sliding surface constant. If the system is on sliding surface $s(t) = 0$. When the system is on slide surface equation (4.23) becomes:

$$e_c(t) = -\beta \int e_c(t)dt \tag{4.24}$$

Tracking error from equation (4.24) given as:-

$$e_c(\infty) = -\beta \int e_c(\infty)dt \tag{4.25}$$

$e_c(\infty)$ becomes zero, as t $\rightarrow \infty$ since $\beta > 0$.

Differentiating equation (4.23) becomes:

$$\dot{s}(t) = [\dot{s}_1(t) \quad \dot{s}_2(t)]^T = \dot{e}_c(t) + \beta e_c(t) \tag{4.26}$$

From the concept of equivalent control law (τ_{eq}) is stated by recognizing that $\dot{s}(t) = 0$ is a necessary condition for state trajectory to stay in the sliding surface[53]. Substituting equation (3.38) into equation (4.26) results equation (4.27).

$$\dot{s}(t) = (\dot{v}_c(t) - E\tau) + \beta e_c(t) = 0 \tag{4.27}$$

Rearranging equation (4.27) and solving for equivalent torque(τ_{eq}), equivalent torque becomes:

$$\tau_{eq(t)} = E^{-1}[\dot{v}_c(t) + \beta e_c(t)] \tag{4.28}$$

where

$$E^{-1} = -\frac{R}{2L}\begin{bmatrix} -Lm & -I \\ -Lm & I \end{bmatrix} \tag{4.29}$$

- Discontinuous controller or switching torque (τ_{sw}) given as equation (4.36)

$$\tau_{sw} = E^{-1}k\tanh(\frac{s}{\alpha}) \tag{4.30}$$

- The overall control law is determined by adding together the equivalent torque and the discontinuous torque.

  So the total control law given as:

$$\tau(t) = \tau_{eq} + \tau_{sw} = E^{-1}[\dot{v}_c(t) + \beta e_c(t) + k\tanh(\frac{s}{\alpha})] \tag{4.31}$$

- The total control law in presence of uncertainties and disturbances becomes:

$$\tau(t) = \tau_{eq} + \tau_{sw} = (E + \Delta E)^{-1}[\dot{v}_c(t) + \beta e_c(t) + k\tanh(\frac{s}{\alpha}) - \tau_d] \tag{4.32}$$

Where as

$$k = \begin{bmatrix} k_{11} & 0 \\ 0 & k_{22} \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_{11} & 0 \\ 0 & \beta_{22} \end{bmatrix} \tag{4.33}$$

$\tau = [\tau_R \quad \tau_L]^T$, $v_c = [v_b \quad \omega_b]^T$, $v_a = [v \quad \omega]^T$, $\tau_d = [\tau_{d1} \quad \tau_{d2}]^T$ $\quad \tanh(\frac{s}{\alpha}) = [\tanh(\frac{s_1}{\alpha}) \quad \tanh(\frac{s_2}{\alpha})]^T$

4.3.1 Lyapunov Stability Analysis of SMC

Step One: The sliding mode control with proportional integral(PI) sliding surface is designed to stabilize the inner loop such that the sliding surface ($s(t)$) satisfies:

$$s(t) = [s_1(t) \quad s_2(t)]^T = e_c + \beta \int e_c dt \tag{4.34}$$

where as e_c and $\dot{e}_c$ represented in equation (4.21) and equation (4.22) respectively. β are positive sliding surface integral constant.

Step Two: Define candidate Lyapunov function which is positive definite for the inner loop sliding mode control with PI sliding surface :

$$V = \frac{1}{2}s^2 + \frac{1}{2}e_c^2 \tag{4.35}$$

Step Three: Calculate the time derivative of V along the system trajectories.

$$\dot{V} = s\dot{s} + e_c\dot{e}_c \tag{4.36}$$

Substituting s and $\dot{s}$ from equation (4.23) and (4.26) into equation (4.36) gives below equation.

$$\dot{V} = (e_c + \beta \int e_c dt)(\dot{e}_c + \beta e_c) + e_c\dot{e}_c \tag{4.37}$$

After multiplication $\dot{V}$ becomes

$$\dot{V} = 2e_c\dot{e}_c + (\beta \int e_c dt)(\dot{e}_c + \beta e_c) + e_c^2\beta \tag{4.38}$$

Step Five Removing integral term from above equation of the error e_c, and by assuming that the sliding surface error e_c converges to zero as $t \to \infty$ due to the sliding mode control. $\dot{V}$ becomes:

$$\dot{V} = e_c(2\dot{e}_c + \beta e_c) \tag{4.39}$$

where as $e_c = (v_c - v_a)$

By substituting e_c in equation (4.38) and multiplying and rearranging, and also substituting total control law from equation (4.31) into equation (4.38) gives below equation resulted in:

$$\begin{aligned}
\dot{V} &= (v_c - v_a)(2(\dot{v}_c - \dot{v}_a) + \beta(v_c - v_a)) \\
&= (v_c - v_a)(2(\dot{v}_c - E\tau) + \beta e_c) \\
&= (v_c - v_a)(2(\dot{v}_c - E((E^{-1}(\dot{v}_c + \beta e_c + k\tanh\frac{s}{\alpha}))) + \beta e_c) \\
&= (v_c - v_a)(2(\dot{v}_c - \dot{v}_c - \beta e_c - k\tanh\frac{s}{\alpha})) + \beta e_c) \\
&= (v_c - v_a)(2(-\beta e_c - k\tanh\frac{s}{\alpha})) + \beta e_c) \\
&= (e_c)(2(-\beta e_c - k\tanh\frac{s}{\alpha})) + \beta e_c) \\
&= (e_c)(-\beta e_c - 2k\tanh\frac{s}{\alpha})
\end{aligned} \tag{4.40}$$

After derivation $\dot{V}$ becomes

$$\dot{V} = -e_c^2\beta - 2ke_c\tanh(\frac{s}{\alpha}) \tag{4.41}$$

properties of $\tanh(-\frac{s}{\alpha}) = -\tanh(\frac{s}{\alpha})$

Step Six: From equation (4.41) for $\dot{V} \leq 0$ the value of beta(β) and gain(k) must greater than or equal to zero, or $\dot{V} \leq 0$ when $\beta \geq 0$ and $k \geq 0$. So its possible to conclude that the designed SMC is stable for positive value of beta(β) , and gain(k).

4.4 Fuzzy Sliding Mode Controller Design

Fuzzy sliding mode control (FSMC) is a control strategy that combines elements of both fuzzy logic control and sliding mode control. The most prominent drawback associated with conventional SMC is chattering, characterized by high-frequency switching of the control signal near the sliding surface. Chattering can lead to excessive mechanical wear, increased energy consumption, and audible noise, making it undesirable in practical applications[21][57]. This drawback is due to discontinuous controller part of SMC equation (4.24) shows discontinuous part of SMC. In previous section 4.3 SMC was designed, in this section fuzzy logic controller implemented in order to mitigate chattering phenomena associated with conventional SMC of discontinuous controller part. By replacing discontinuous controller part of SMC by FLC in equation (4.31) the total control law becomes:

$$\tau(t) = \tau_{eq} + \tau_{sw} = E^{-}[\dot{v}_c(t) + \beta e_c(t) + Ufuzzy] \tag{4.42}$$

where as $s = [s_1 \quad s_2]^T$, $\dot{s} = [\dot{s}_1 \quad \dot{s}_2]^T$, $Ufuzzy = [u_{fuzzy1} \quad u_{fuzzy2}]^T$.
To achieve this, two separate FLCs are utilized: one to control the torque applied to the right wheel and another for the left wheel. The initial FLC takes state variables (s_1, $\dot{s}_1$) as inputs and generates the output (u_{fuzzy1}). Similarly, the second FLC takes state variables (s_2, $\dot{s}_2$) as inputs and produces an output (u_{fuzzy2}). To ensure smooth control, seven membership functions are employed for both the input and output variables in each FLC. In total, a rule base of 49 rules is utilized to govern the decision-making process within the FLCs.

The configuration of a fuzzy control system is presented in the figure 4.3. Table 4.1 indicates that the rules base used for FLC, and also figure 4.4 shows that input membership function and output membership function used for FLC system.

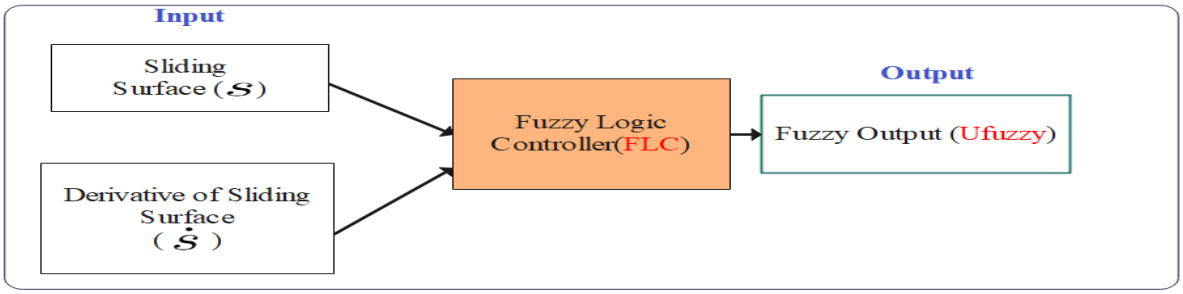


Figure 4.3: Fuzzy Logic Controller Block Diagram

Table 4.1: Rules Base of the Fuzzy Logic Controller

S \ $\dot{S}$	NB	NM	NS	Z	PS	PM	PB
NB	PB	PB	PM	PM	PS	PS	Z
NM	PB	PM	PM	PS	PS	Z	NS
NS	PM	PM	PS	PS	Z	NS	NS
Z	PM	PS	PS	Z	NS	NS	NM
PS	PS	PS	Z	NS	NS	NM	NM
PM	PS	Z	NS	NS	NM	NM	NB
PL	Z	NS	NS	NM	NM	NB	NB

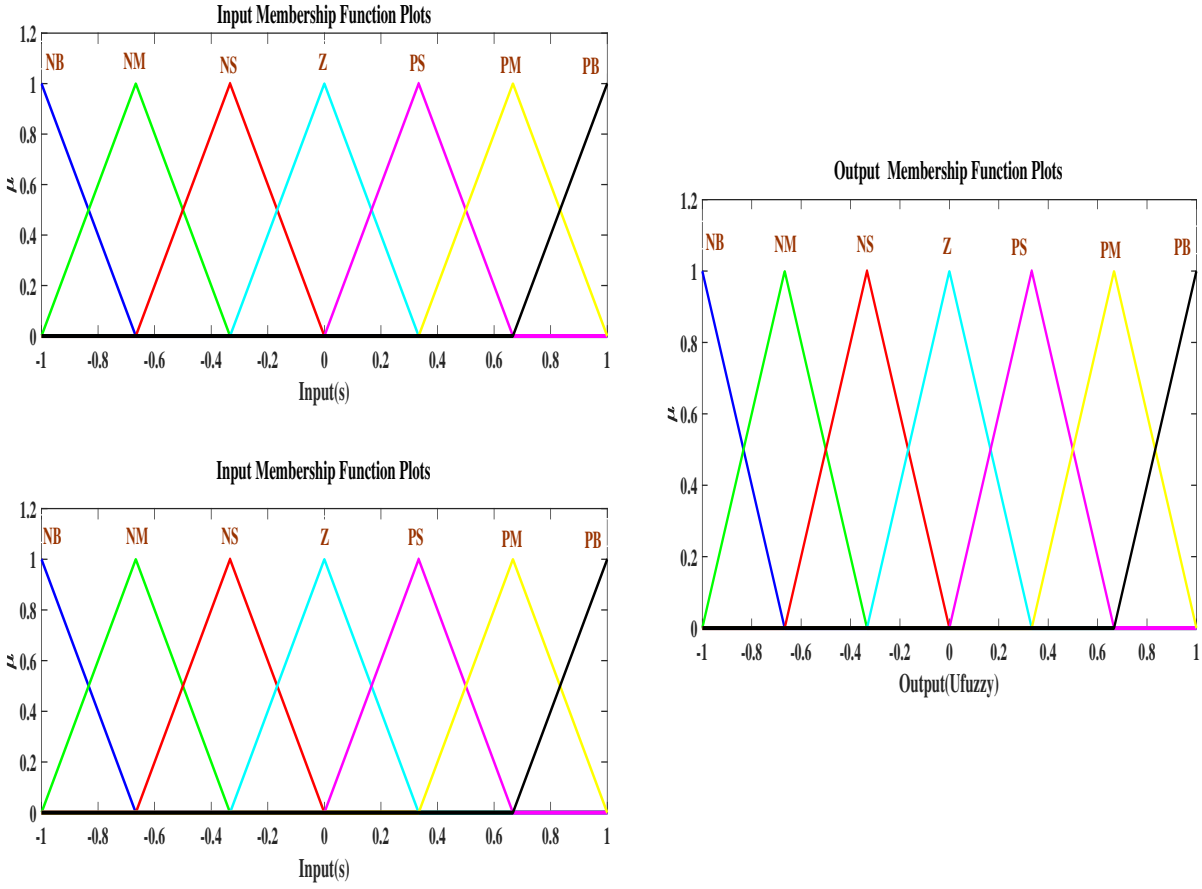


Figure 4.4: Input and Output Membership Function for FLC

4.5 Extended Kalman Filter Design for Position and Orientation Estimation of TWMR

The majority of modern systems are built with a variety of sensors that allow estimating hidden (unknown) variables based on a sequence of measurements. For example, the global positioning system (GPS) receiver estimates location and velocity, where location and velocity are hidden variables, and measurements are the differential time of satellite signal arrival. One of the most challenging issues for tracking and control systems is providing accurate and precise estimation of hidden variables in the presence of uncertainty. The GPS receiver's measurement uncertainty is affected by various external factors, including thermal noise, atmospheric effects, small variations in satellite locations, receiver clock precision, and many more.

It is employed to integrate the internal position estimation with external measurements for the localization of a wheeled mobile robot. The Extended Kalman Filter (EKF) is a widely adopted method for addressing challenges in mobile robot applications that involve sequential localization and mapping. The effectiveness of EKF hinges on the accurate modeling of covariance matrices for system and measurement noise. Inaccuracies in these statistical analyses can significantly impact performance [58]. EKF offers a solution to the estimation problem in the presence of nonlinear models. The approach relies on a Gaussian probability density representation of the robot's position, and it can be applied with various sensors to achieve precise pose estimation in environments affected by random noise.

The assumptions of linear state transitions and linear measurements with added Gaussian noise are rarely fulfilled in practice [59]. For example, a robot that moves with a constant transnational and rotational velocity typically moves on a circular trajectory, which cannot be described by linear next-state transitions [59]. This observation, along with the assumption of unimodal beliefs, makes the plain kalman filter inapplicable to all but the most trivial robotics problems [59].

Moreover, the Kalman filter is implemented in the simplest system, involving a two-wheeled mobile robot, to enhance accuracy during data collection for position estimation in indoor navigation [60]. The discrete-time model of the Three-Wheeled Mobile Robot (TWMR) can be expressed as follows:

$$\begin{aligned} x_{k+1} &= f(x_K + u_k) + w_k \\ z_k &= h(x_k) + v_k \end{aligned} \tag{4.43}$$

In this section, Extended Kalman Filter for the TWMR in designed. To create the Extended Kalman Filter, the plant and measurements are linearized by eliminating high-order terms from the Taylor expansion.

4.5.1 Steps for Designing of Extended Kalman Filter

Step1: Discertization of kinematic model of TWMR equation (3.15) becomes:-

$$x_k = f(x_{k-1}, u_{k-1}) = \begin{bmatrix} x_{k-1} + v_{k-1} cos\theta_{k-1} \Delta T \\ y_{k-1} + v_{k-1} sin\theta_{k-1} \Delta T \\ \theta_{k-1} + w_{k-1} \Delta T \end{bmatrix} \tag{4.44}$$

Equation (4.44) is discrete forward kinematics model of TWMR.
Where $X_{k-1} = (x_{k-1}, y_{k-1}, \theta_{k-1})^T$, ΔT is sampling time, and $u_{K-1} = (v_{k-1}, w_{k-1})^T$ are the position and control signal of the robot at a time $k-1$ and $X_k = (x_k, y_k, \theta_k)^T$ is the robot position at the time k .
Step2: Linearizing the plant and measurement model using first order Taylor series approximation. Applying Taylor first order expansion to equation (4.44) resulted in:-

$$x_k = x_k^- + A(x - x_{k-1}) + W_{w_{k-1}} \tag{4.45}$$

where x_k^- is the predicted position through control signal.
First order taylor series approximation to measurement model given as below equation (4.46).

$$Z_k = Z_k^- + H(x - x_k^-) + V u_k \tag{4.46}$$

where A , W , H and V are jacobian matrix.
Step3: From equation (4.44) finding jacobian metrices for A and W.

$$A = \frac{\partial f}{\partial x_{k-1}} = \begin{bmatrix} 1 & 0 & -v_{k-1} sin\theta_{k-1} \Delta T \\ 0 & 1 & v_{k-1} cos\theta_{k-1} \Delta T \\ 0 & 0 & 1 \end{bmatrix} \tag{4.47}$$

$$W = \frac{\partial f}{\partial u_{k-1}} = \begin{bmatrix} cos\theta_{k-1} \Delta T & 0 \\ sin\theta_{k-1} \Delta T & 0 \\ 0 & 1 \end{bmatrix} \tag{4.48}$$

Process noise covariance (Q) and measurement noise covariance(R) given as below equation:

$$Q = \begin{bmatrix} \sigma_{\text{vel}}^2 & 0 & 0 \\ 0 & \sigma_{\text{vel}}^2 & 0 \\ 0 & 0 & \sigma_{\omega}^2 \end{bmatrix} \tag{4.49}$$

where as σ_{vel} represents the noise in linear velocity, and σ_{ω} represents the noise in angular velocity.

$$R = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_{\theta}^2 \end{bmatrix} \tag{4.50}$$

where as σ_x represents the measurement noise in position, and σ_{θ} represents the noise in orientation measurement.

4.6 Particle Swarm Optimization

Achieving optimal performance in BFSMC heavily relies on appropriately setting the controller gains. Conventional tuning methods, such as manual adjustment or trial and error, can be time-consuming, inefficient, and may lack a systematic approach. To address this challenge and ensure that BFSMC operates at its best, advanced optimization techniques are employed. PSO is one such technique that has shown promise in optimizing the controller gains for both BFSMC and BSMC.

4.6.1 Steps for PSO-Based Control Gain Tuning in MATLAB-Simulink

Step 1: Initialization: Create or import control system model in Simulink. Include the parameters or gains that require tuning.

Step 2: Algorithm Setup: Implement the PSO algorithm in MATLAB. Configure the algorithm by specifying parameters such as:

- Defining lower and upper bounds for each gain or parameter requiring tuning.
- Determining the number of particles in the swarm.
- Setting the number of iterations for the PSO algorithm.

- Defining the fitness function, often chosen as the Integral of Time-weighted Absolute Error (ITAE).

Step 3: Fitness Function Implementation: Create a MATLAB fitness function to compute the ITAE for the control system. Develop a Simulink model for evaluating the control system's performance over time.

Step 4: Integration of Simulink and MATLAB: Establish connectivity between Simulink and MATLAB using the Simulink API. Utilize the "sim" command to simulate the control system with different sets of gains or parameters.

Step 5: PSO Optimization Loop: Construct a loop in your MATLAB code to execute the PSO algorithm for the specified number of iterations. During each iteration, the PSO algorithm assesses the fitness of each particle's position by simulating the control system in Simulink.

Step 6: Particle Position Updates: Modify the positions of particles in the swarm based on fitness values, following the update rules of the PSO algorithm. Adjust the gains or parameters related to Backstepping control, Sliding Mode Control (SMC), and Fuzzy Logic Control (FLC).

Step 7: Iterative Optimization: Iterate through the PSO optimization loop for the predetermined number of iterations. The algorithm should progressively approach a set of gains that minimize the ITAE.

Step 8: Optimized Gains Retrieval: Upon completion of the PSO optimization process, obtain a set of optimized gains for your control system.

Step 9: Application of Optimized Gains: Implement the optimized gains back into your Simulink control system model.

Step 10: Validation and Testing: Validate the control system's performance by simulating it with the optimized gains. Ensure that the ITAE is minimized, indicating enhanced control system behavior.

Chapter 5

Simulation Results and Discussion

5.1 Introduction

This chapter provides an examination of the simulation outcomes to the trajectory tracking control of a three wheeled mobile robot using two advanced control strategies: backstepping sliding mode control (BSMC) and backstepping fuzzy sliding mode control (BFSMC). The performance of BSMC and BFSMC is individually examined, with and without the presence of disturbance parameter and variations. Moreover, a comparative analysis of BFSMC and BSMC is presented to assess their respective capabilities. Additionally, the implementation of an extended kalman filter (EKF) is explored for sensor noise reduction. By conducting these simulations and comparative studies, valuable insights are gained into the trajectory tracking performance and robustness of these control strategies, especially in the presence of disturbances , parameter variations, and unmodeled dynamics.

5.2 Simulation Result for Trajectory Tracking of TWMR using BSMC and BFSMC

5.2.1 Circle Trajectory Tracking

5.2.1.1 Circle Trajectory Tracking using BSMC

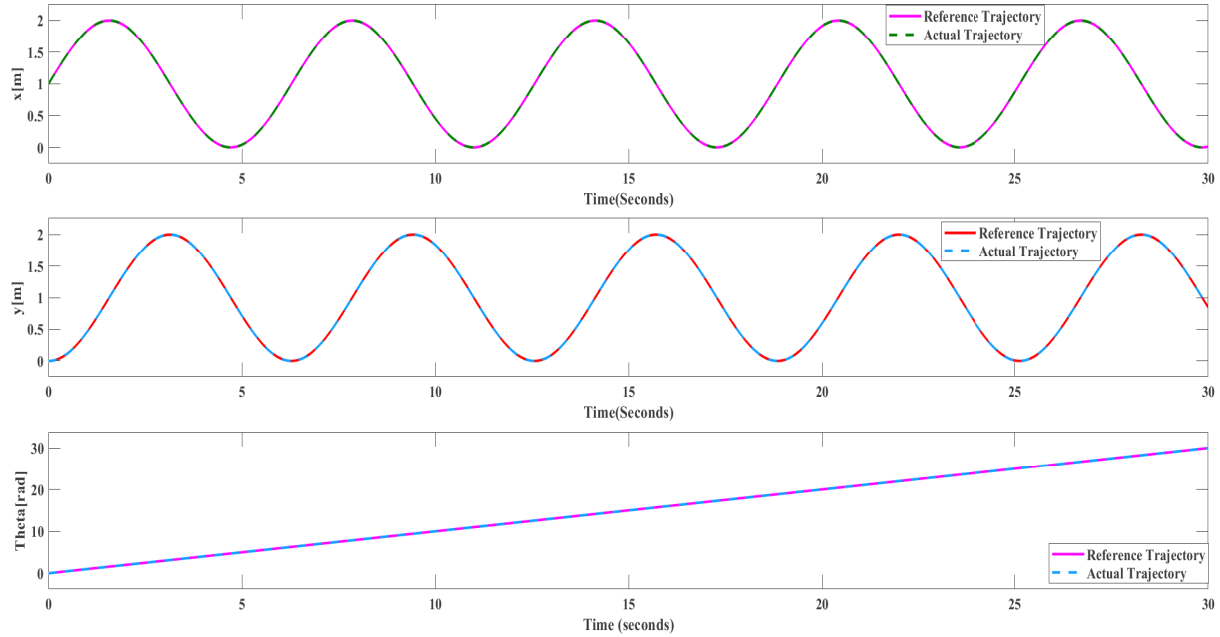


Figure 5.1: Simulation Result of Pose for Circular Trajectory using BSMC

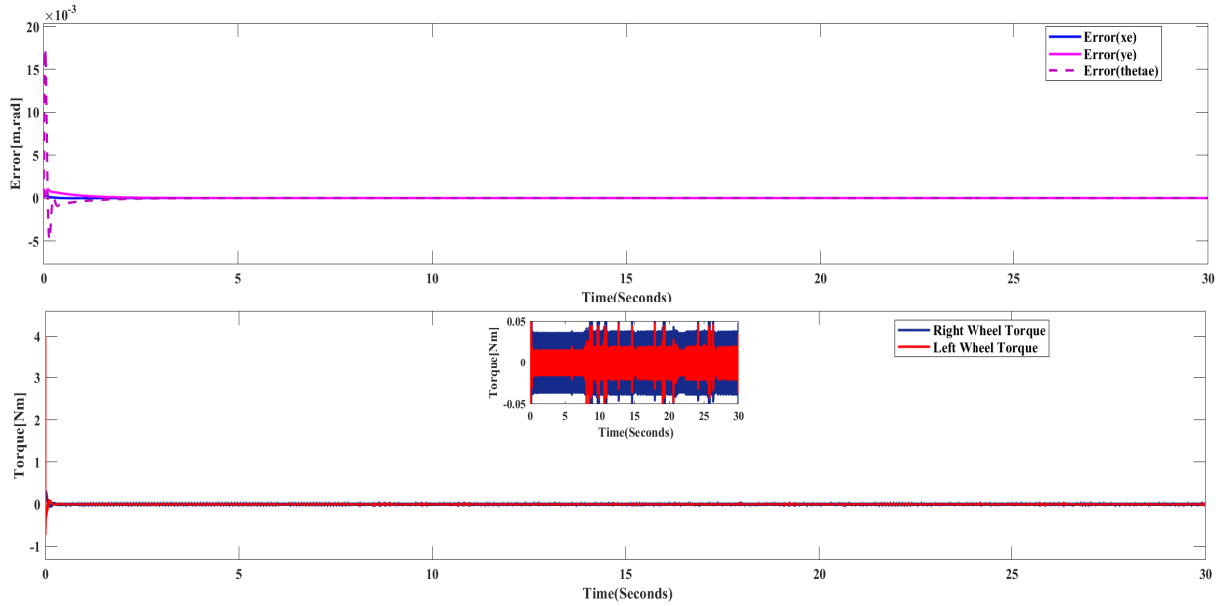


Figure 5.2: Torque and Pose Error for Circular Trajectory using BSMC

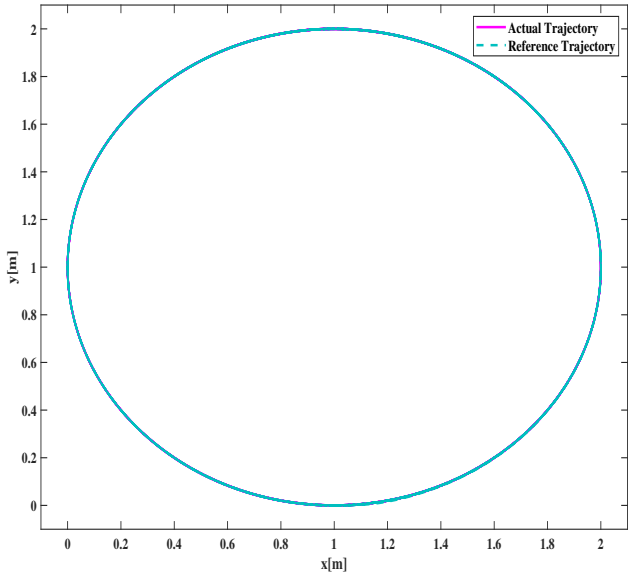


Figure 5.3: X-Y Plot of Circular Trajectory Simulation Result using BSMC

5.2.1.2 Circle Trajectory Tracking using BFSMC

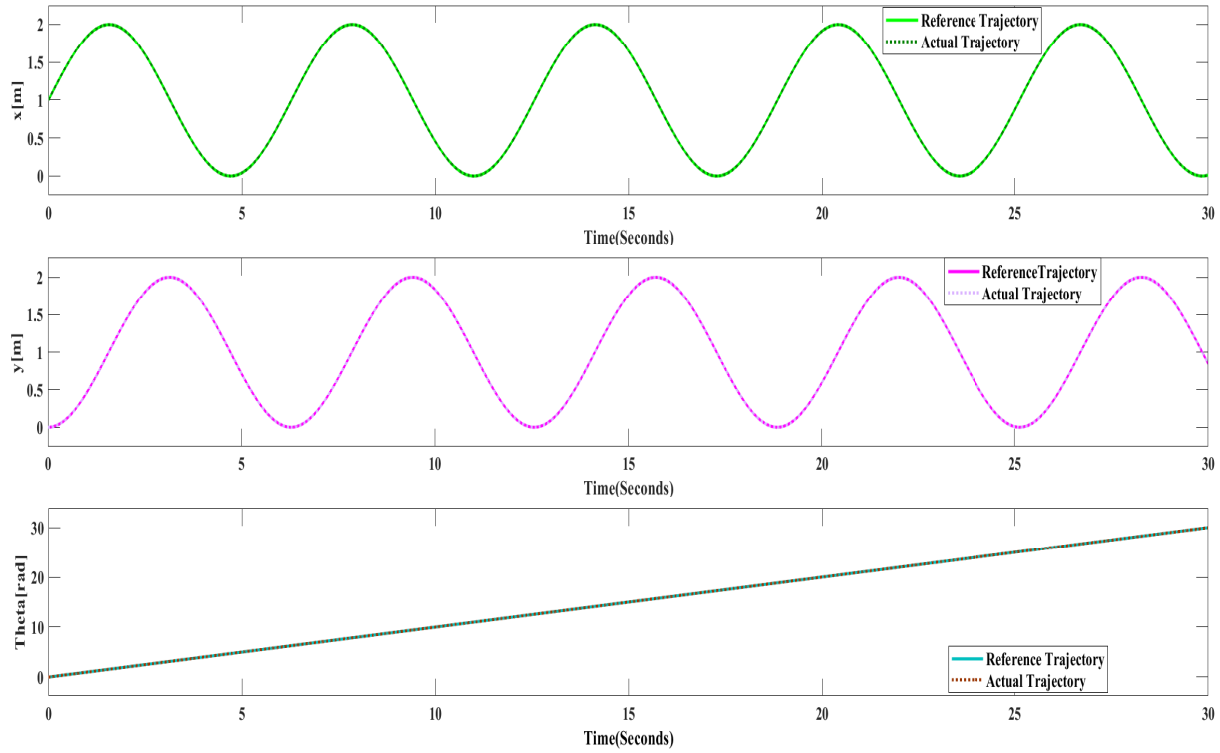


Figure 5.4: Simulation Result of Pose for Circular Trajectory using BFSMC

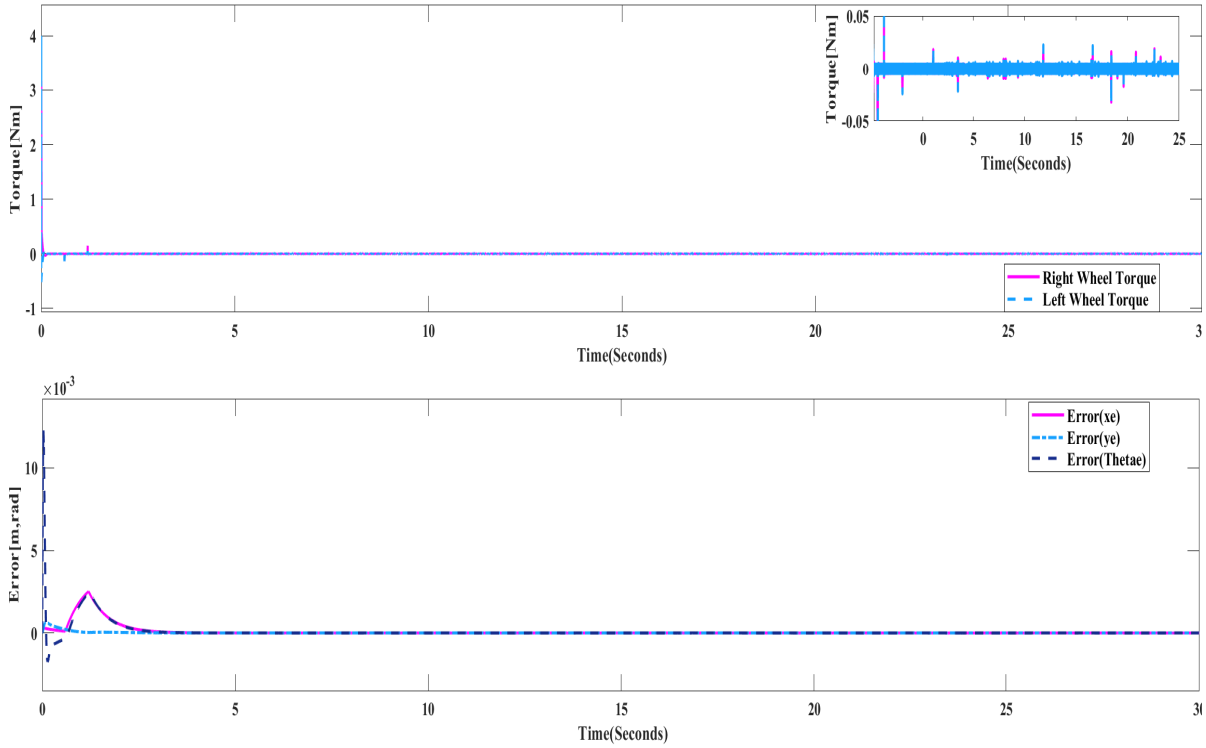


Figure 5.5: Pose Error and Torque for Circular Trajectory using BFSMC

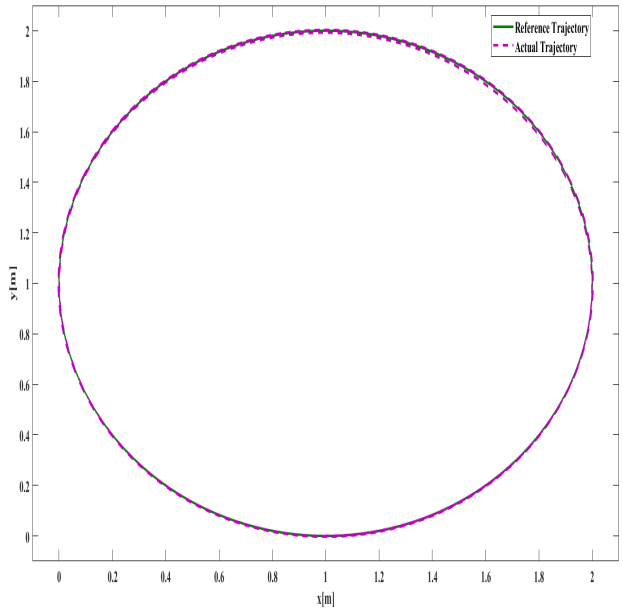


Figure 5.6: X-Y Plot for Circular Trajectory using BFSMC

5.2.2 Infinity Trajectory Tracking

5.2.2.1 Infinity Trajectory Tracking using BSMC

$$x_r = \cos t$$
$$y_r = \sin 2t \tag{5.1}$$

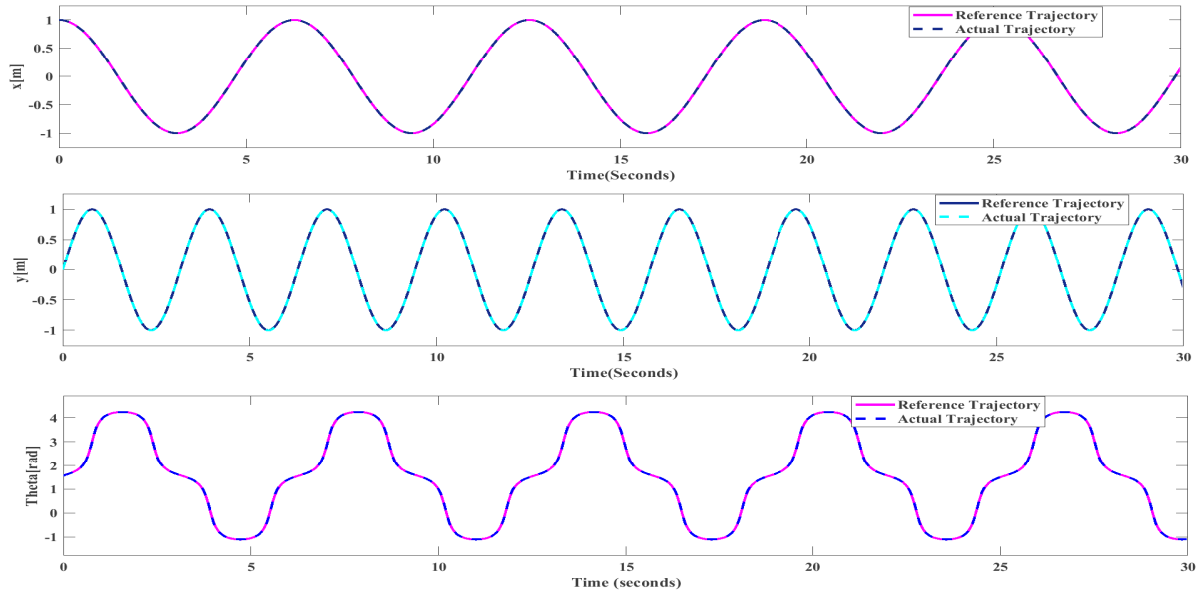


Figure 5.7: Simulation Result for Pose of Infinity Trajecyory using BSMC

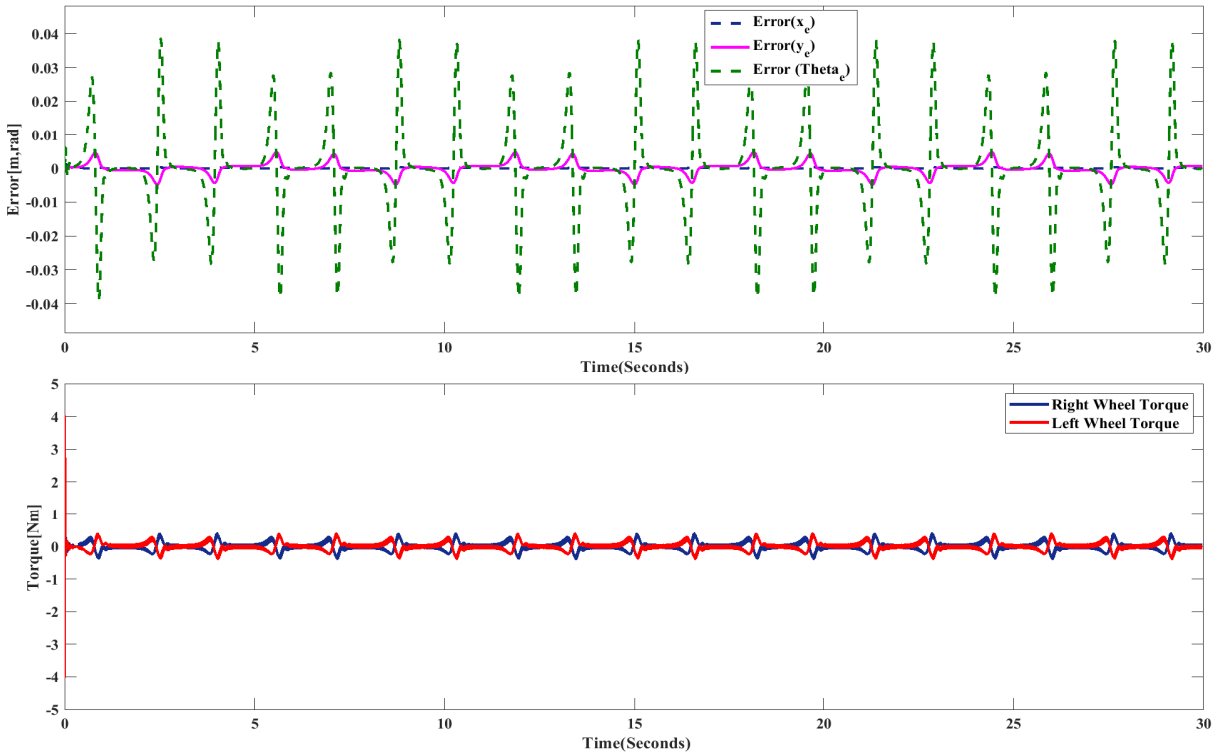


Figure 5.8: Torque and Pose Error for Infinity Trajectory using BSMC

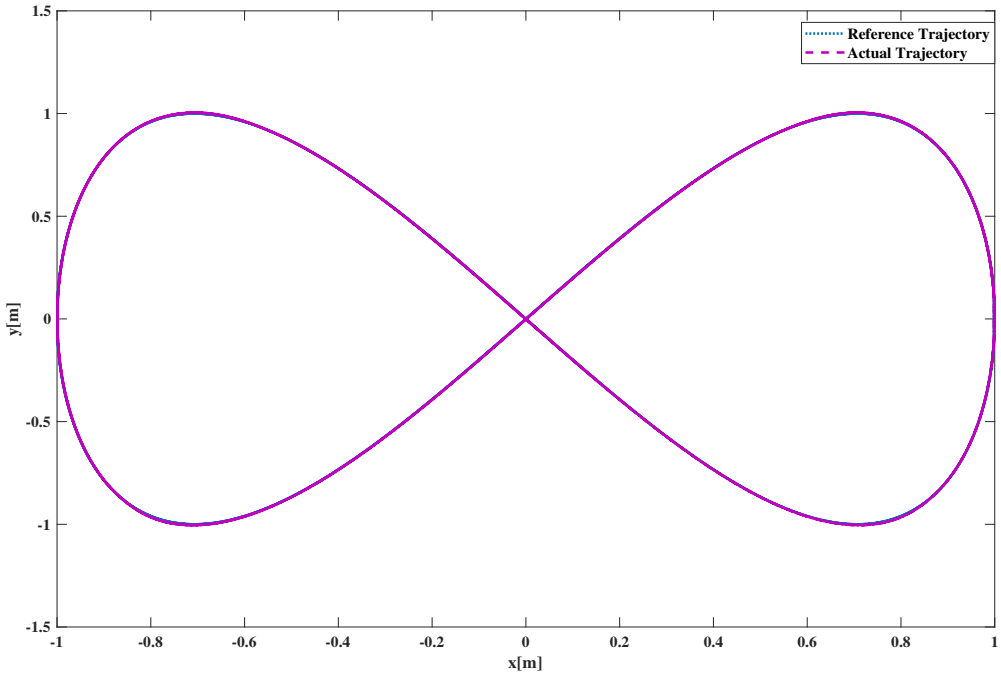


Figure 5.9: X-Y Plot of Infinity Trajectory using BSMC

5.2.2.2 Infinity Trajectory Tracking using BFSMC

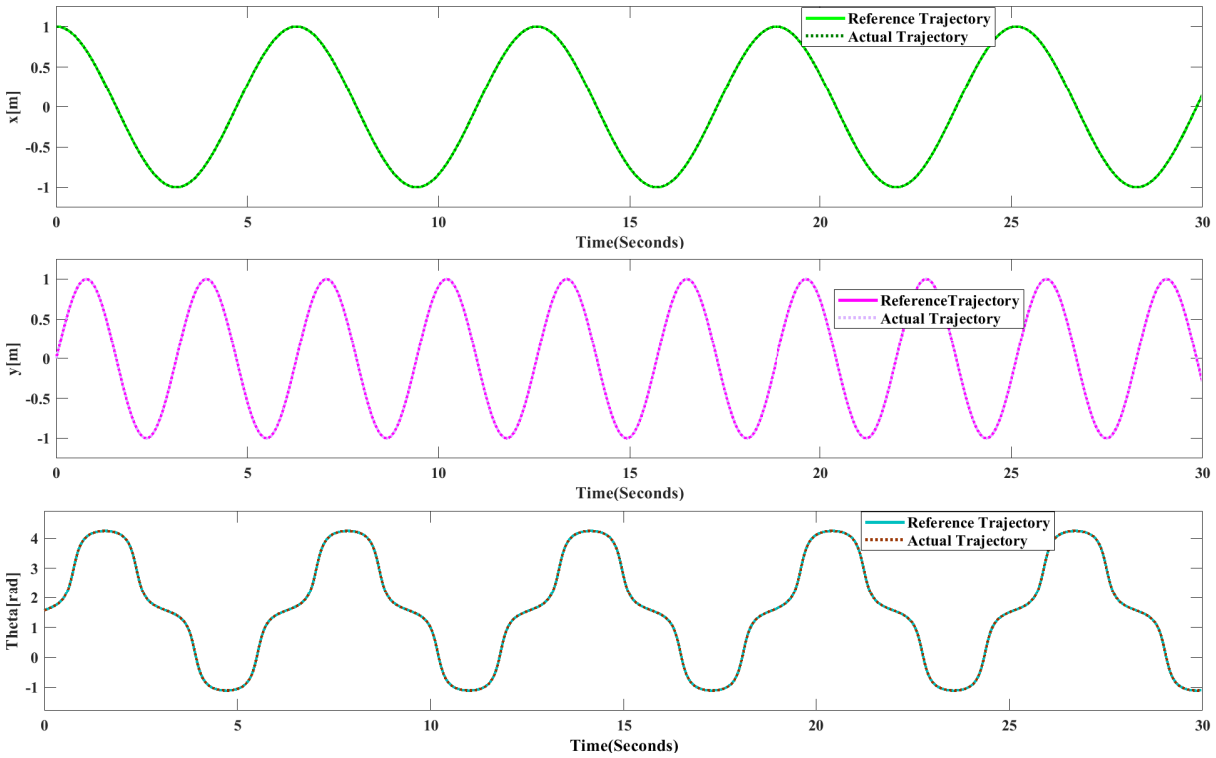


Figure 5.10: Pose for Infinity Trajectory Tracking Result using BFSMC

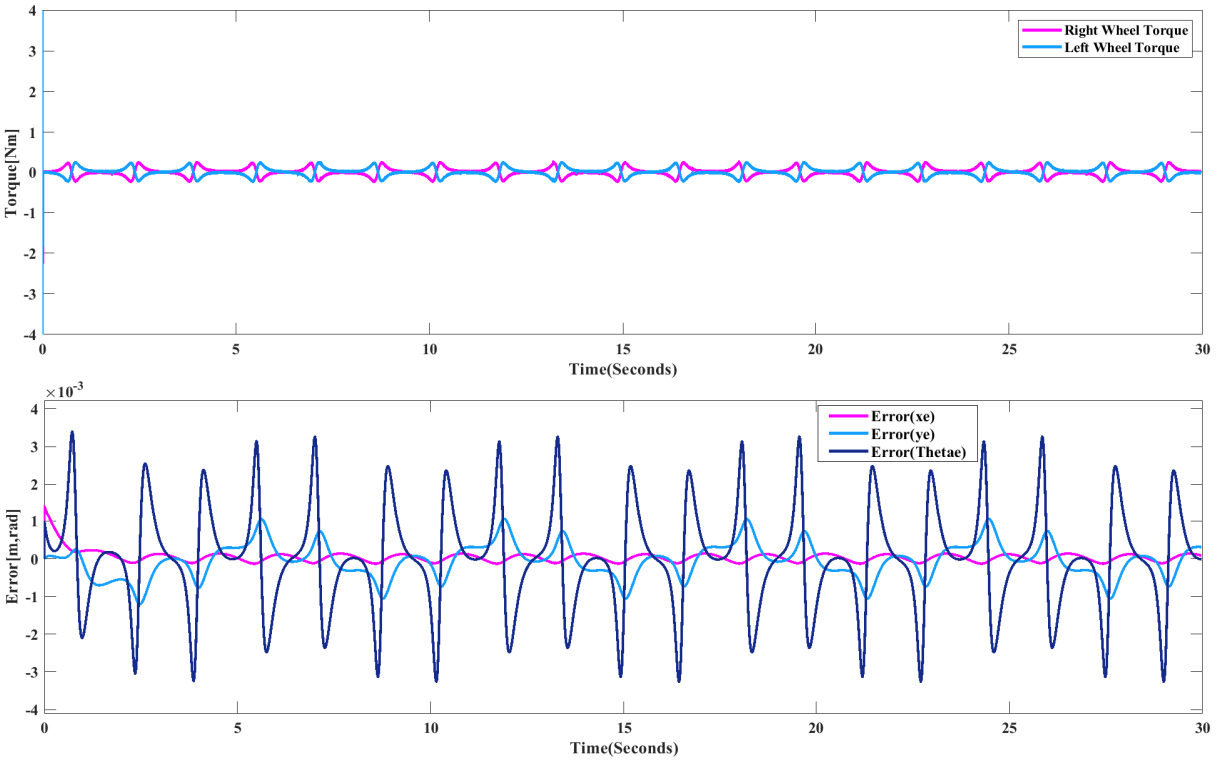


Figure 5.11: Trajectory Tracking Error and Torque for Infinity Trajectory using BFSMC

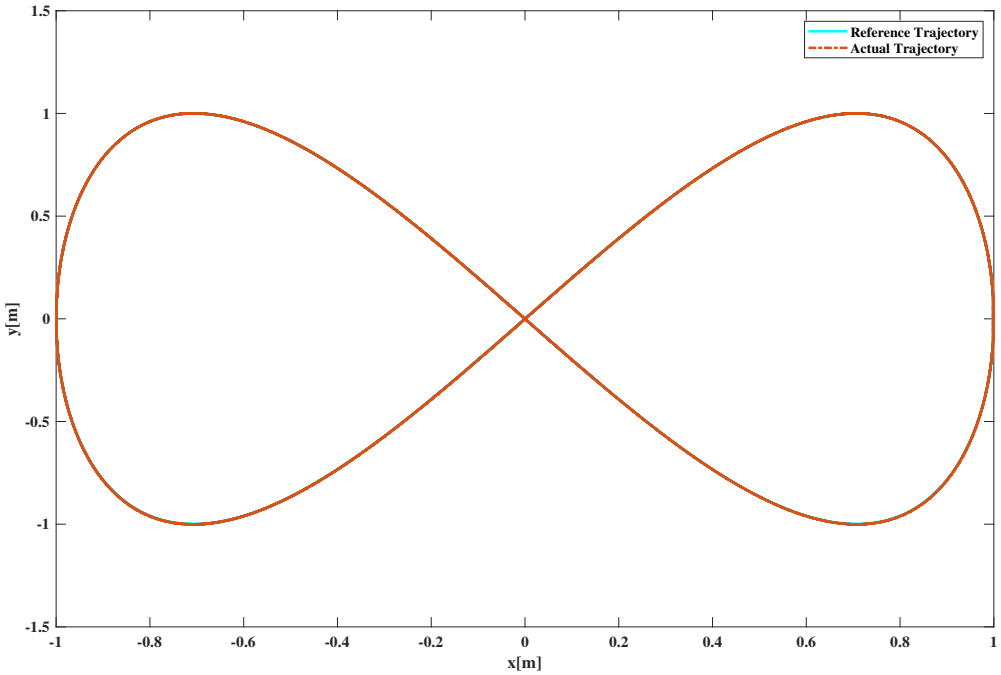


Figure 5.12: X-Y Plot for Infinity Trajectory Tracking Result using BFSMC

5.2.3 Stair Trajectory

5.2.3.1 Stair Trajectory Tracking using BSMC

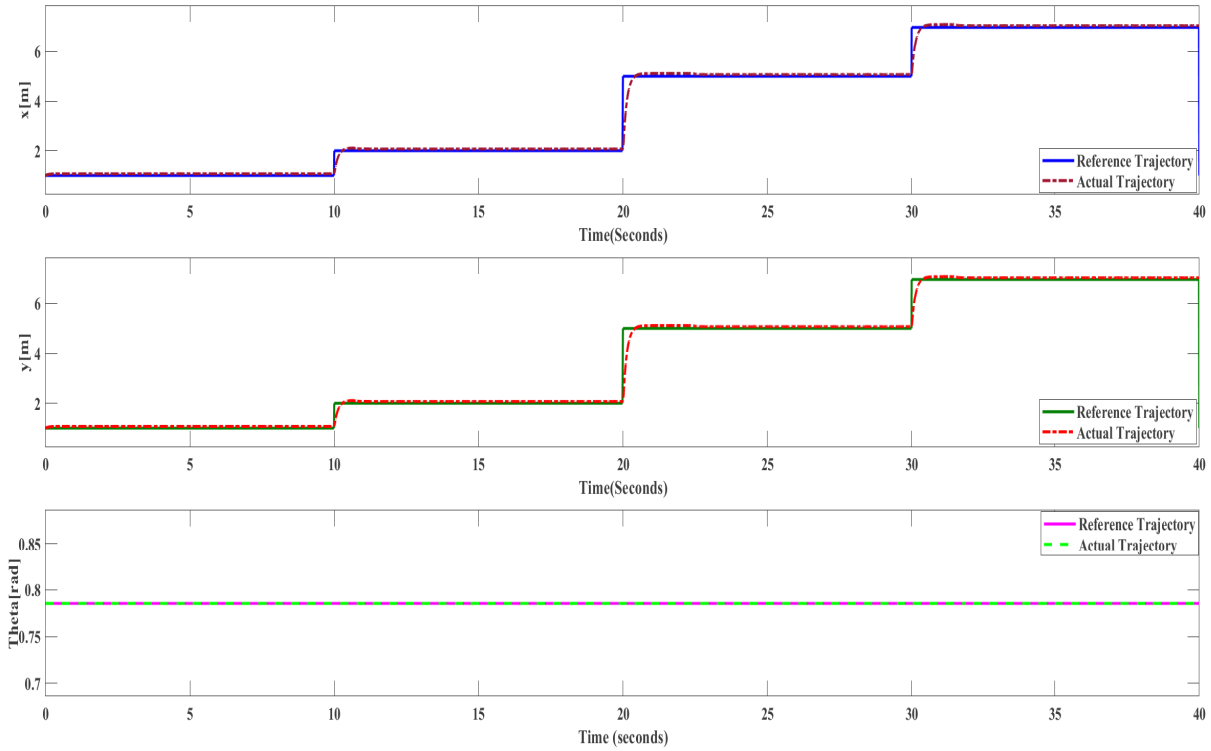


Figure 5.13: Simulation Result for Pose for Stair Trajectory using BSMC

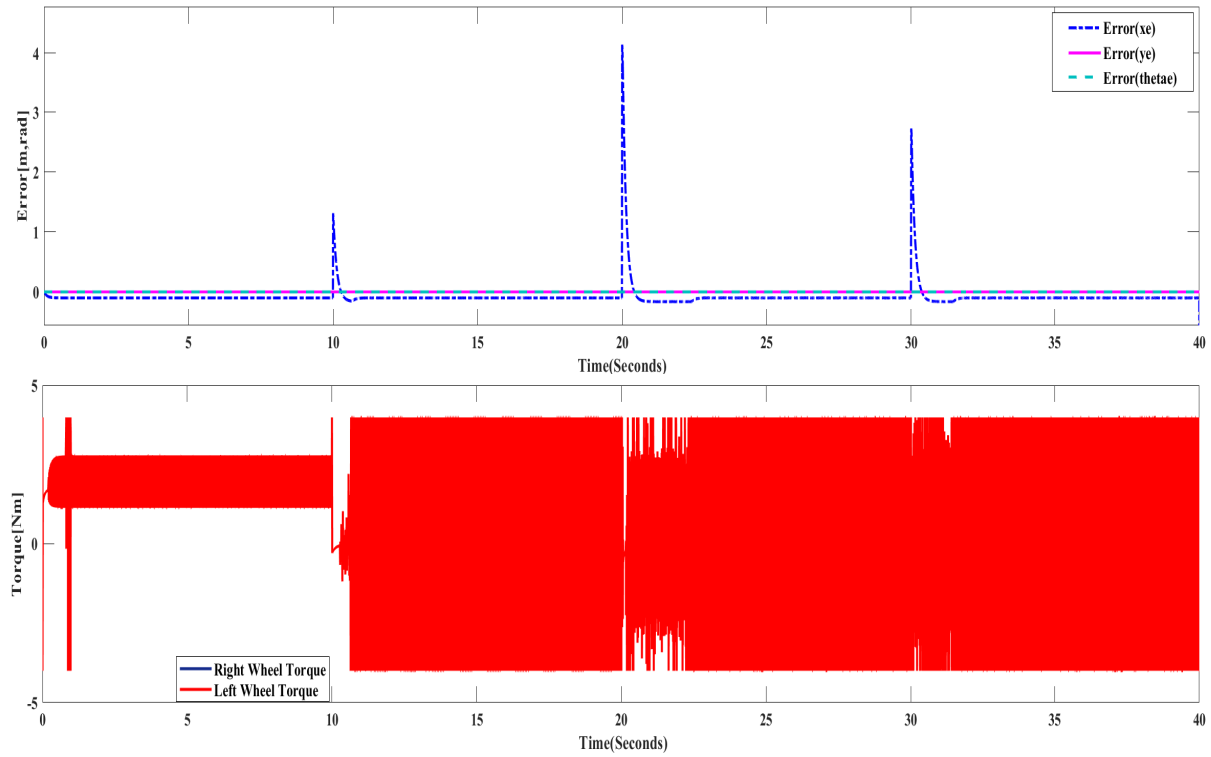


Figure 5.14: Pose Error and Torque for Stair Trajectory using BSMC

5.2.3.2 Stair Trajectory Tracking using BFSMC

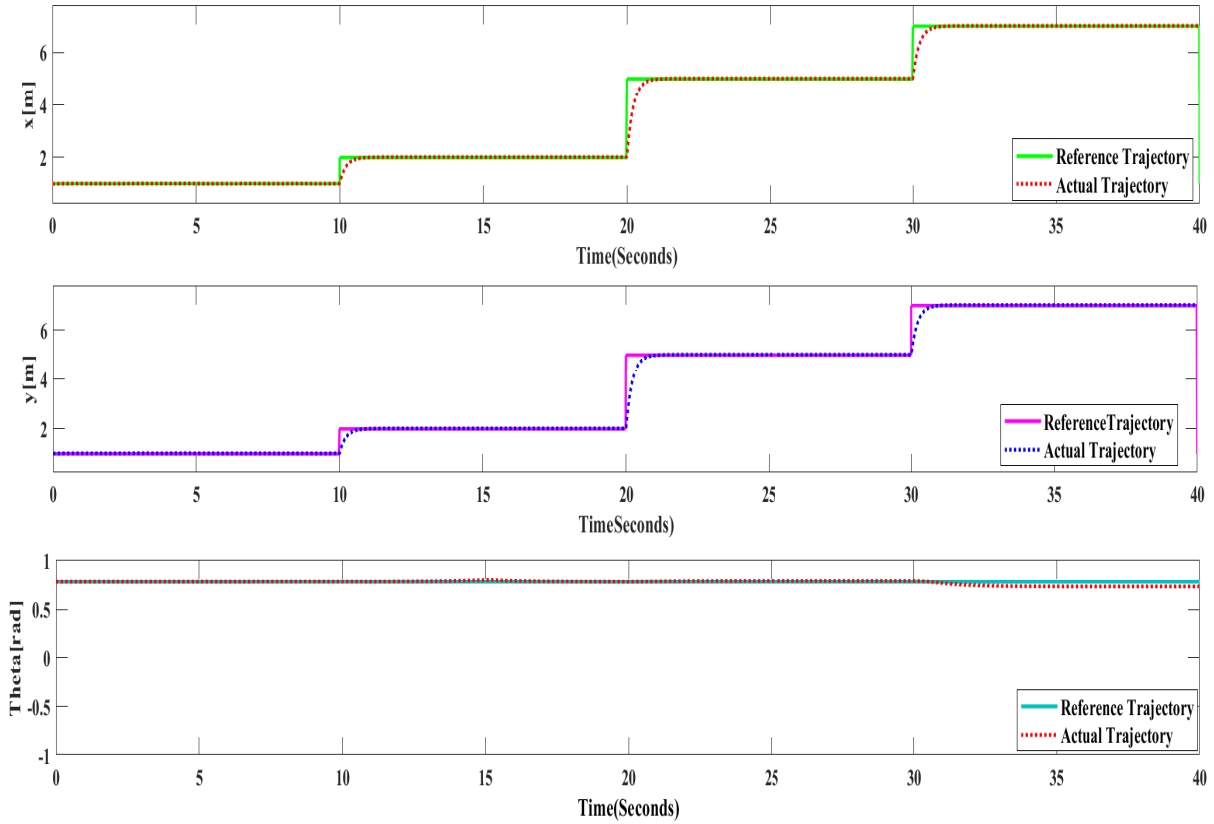


Figure 5.15: Pose for Stair Trajectory Tracking Result of WMR using BFSMC

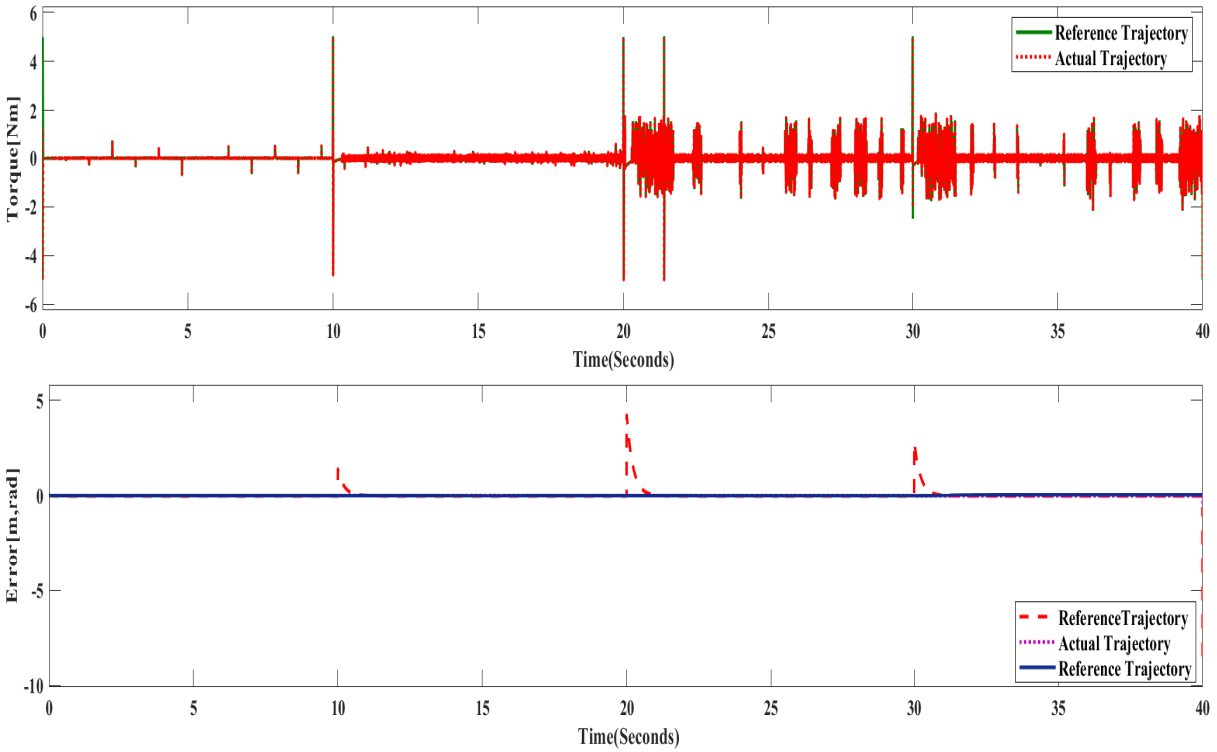


Figure 5.16: Trajectory Tracking Error for Pose and Torque for WMR using BFSMC for Stair Trajectory

5.2.4 Spiral Trajectory

5.2.4.1 Spiral Trajectory Tracking using BSMC

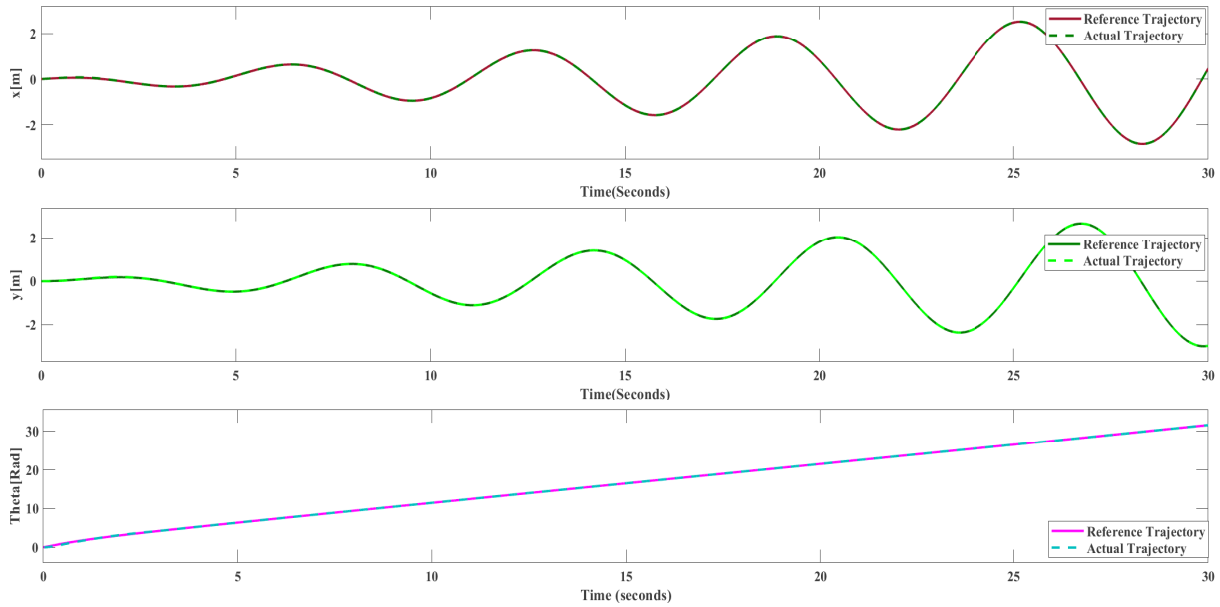


Figure 5.17: Simulation Result for Pose of Spiral Trajectory using BSMC

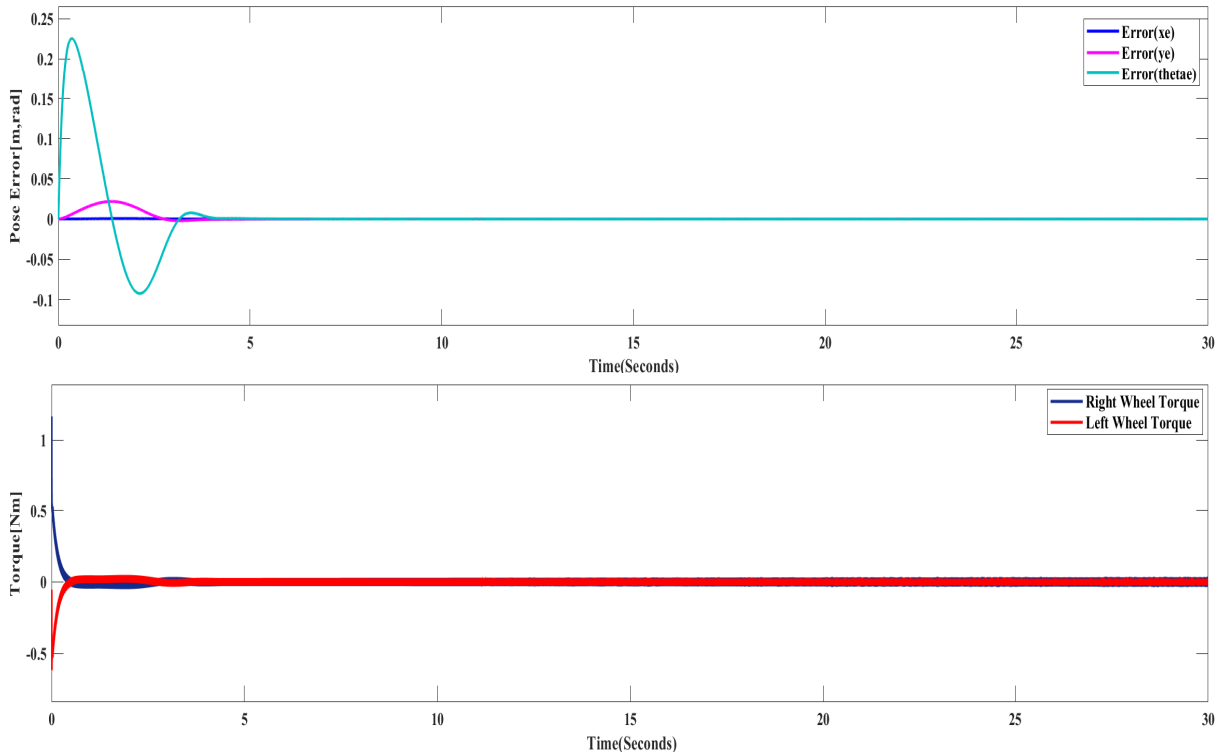


Figure 5.18: Pose Error and Torque for Spiral Trajectory using BSMC

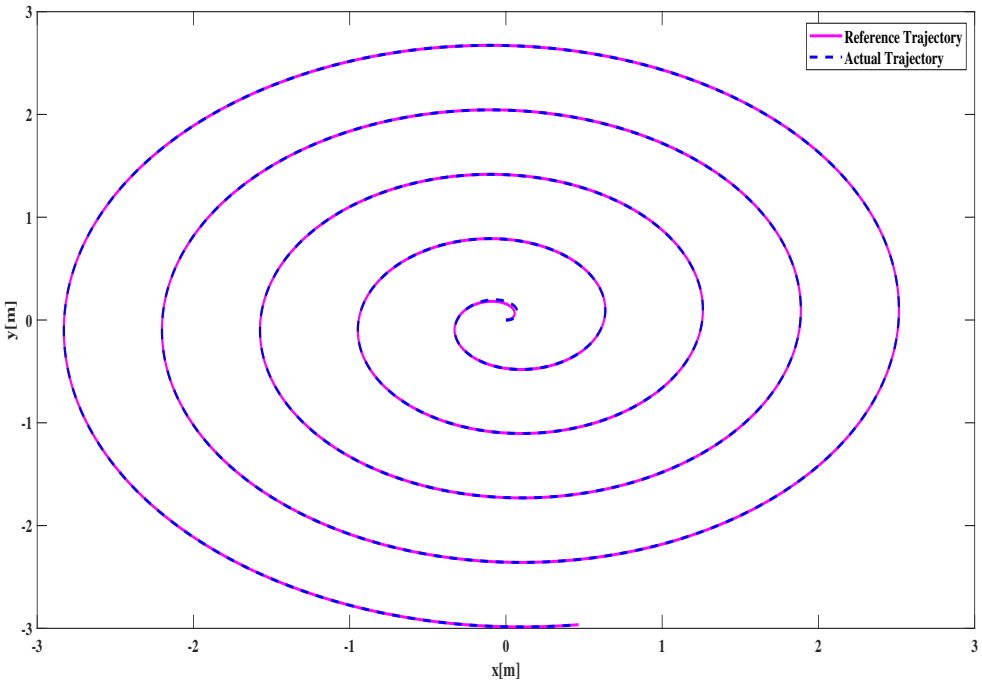


Figure 5.19: X-Y Plot of Spiral Trajectory using BSMC

5.2.4.2 Spiral Trajectory Tracking using BFSMC

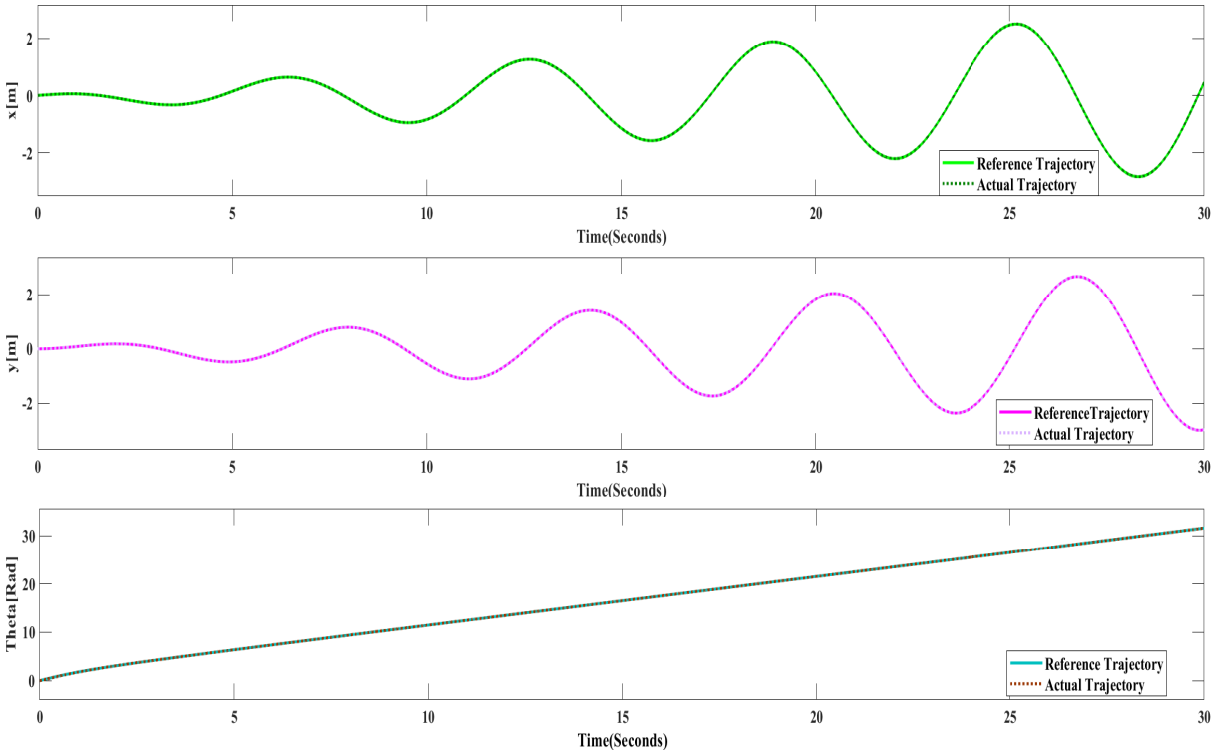


Figure 5.20: Pose for Spiral Trajectory Tracking Result of WMR using BFSMC

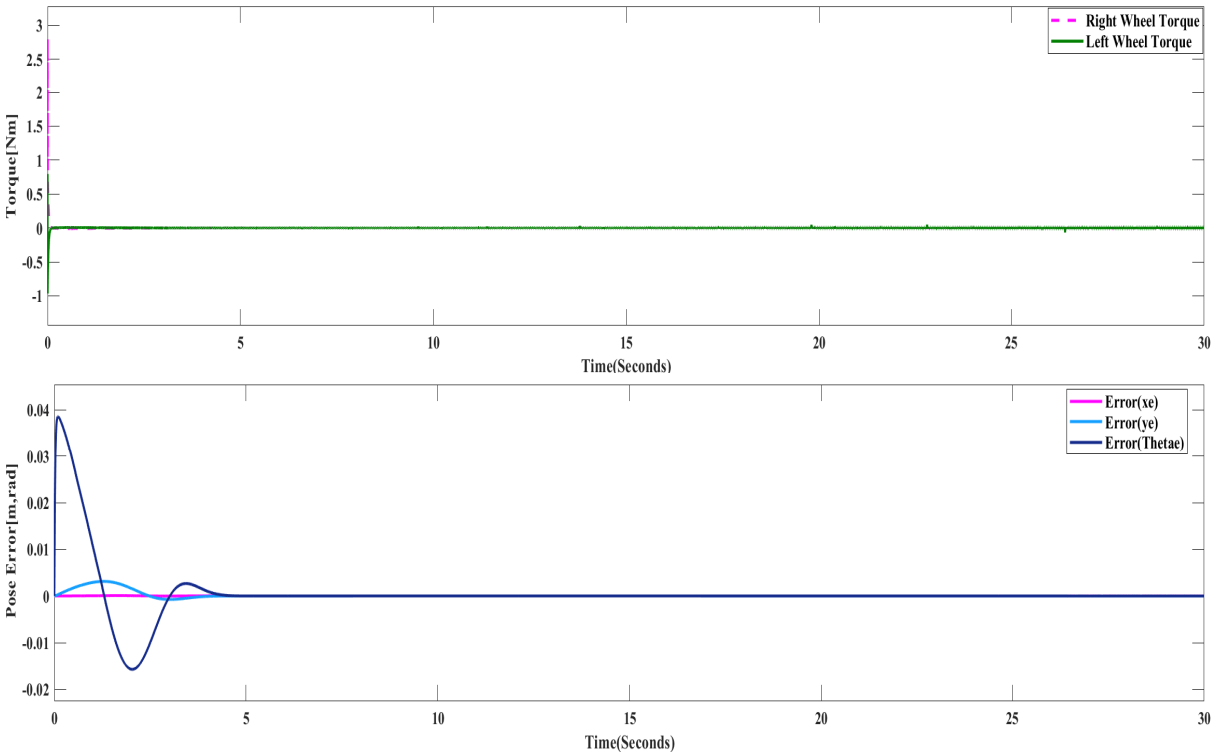


Figure 5.21: Trajectory Tracking Error for Pose and Control Effort for WMR using BFSMC for Spiral Trajectory

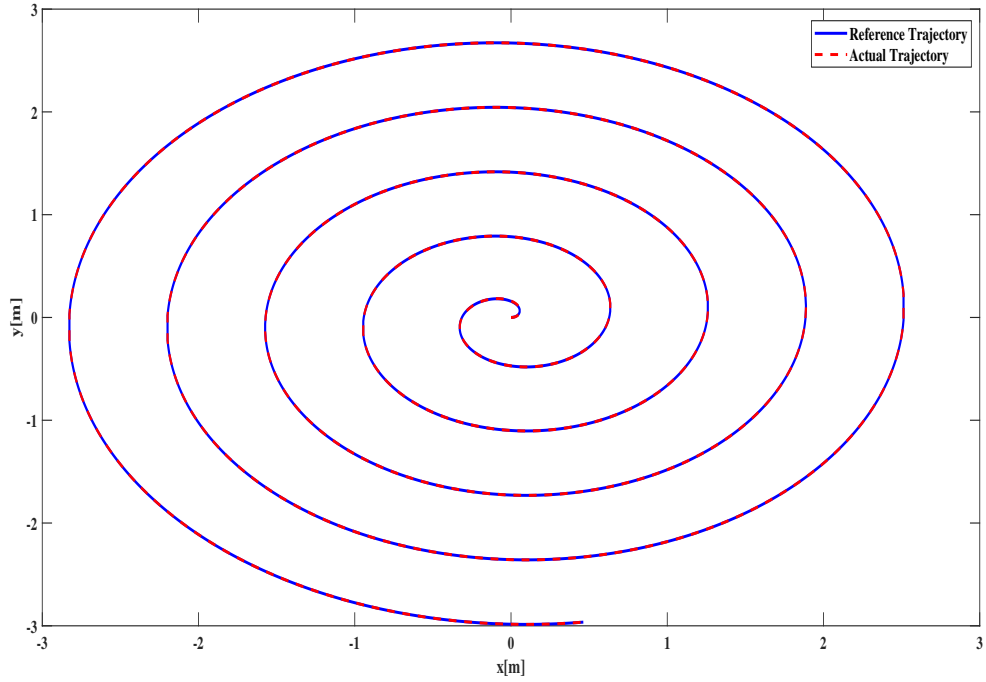


Figure 5.22: X-Y Plot for Spiral Trajectory Tracking Result using BFSMC

5.2.5 Square Trajectory

5.2.5.1 Square Trajectory Tracking Using BSMC

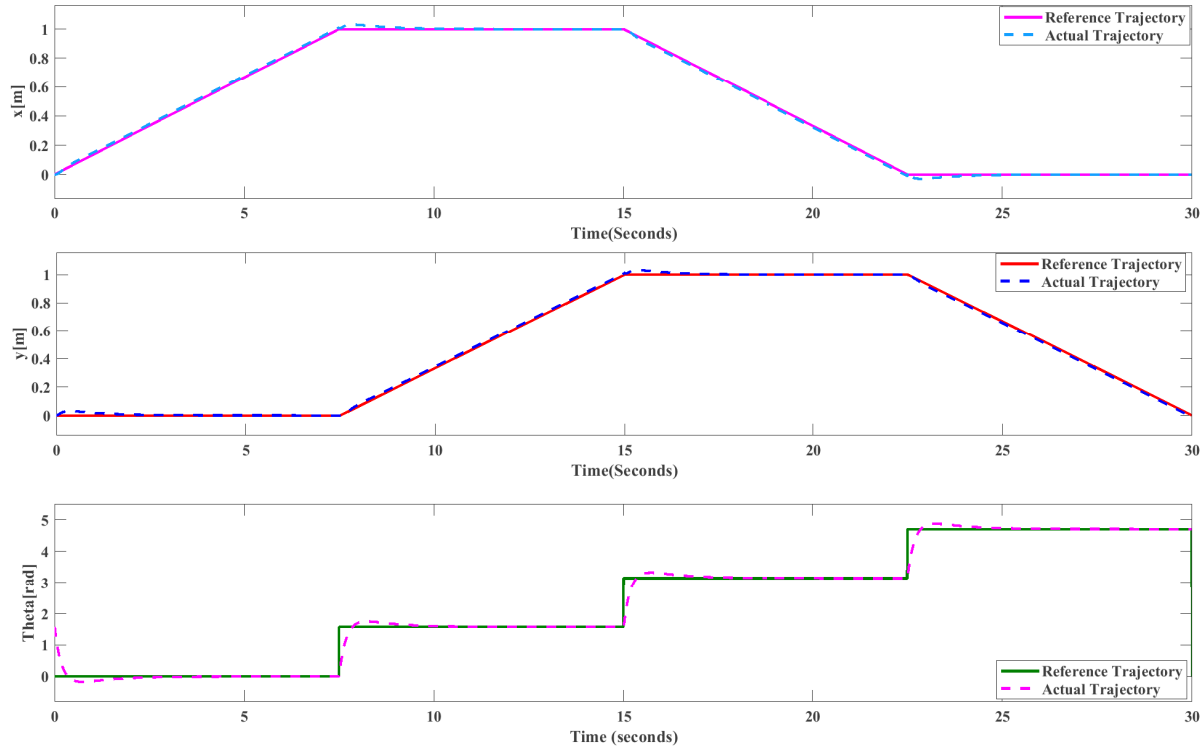


Figure 5.23: Simulation Result for Pose of Square Trajectory using BSMC

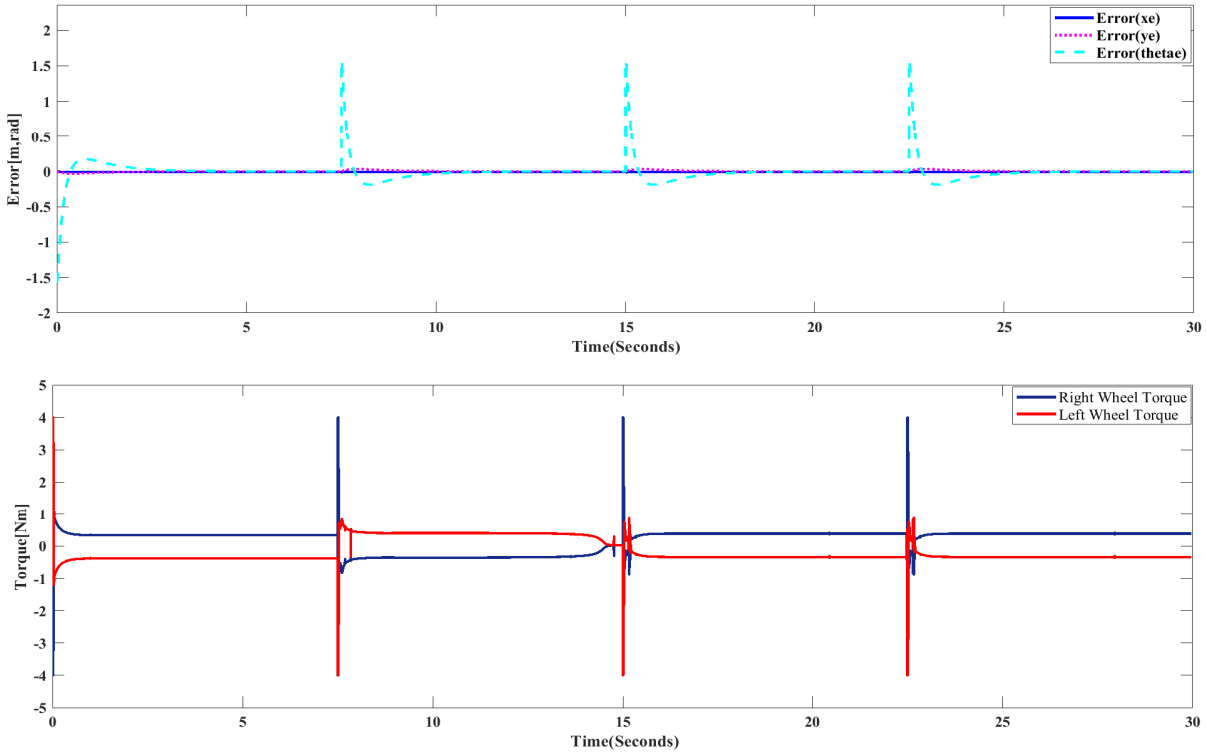


Figure 5.24: Pose Error and Torque using BSMC for Square Trajectory

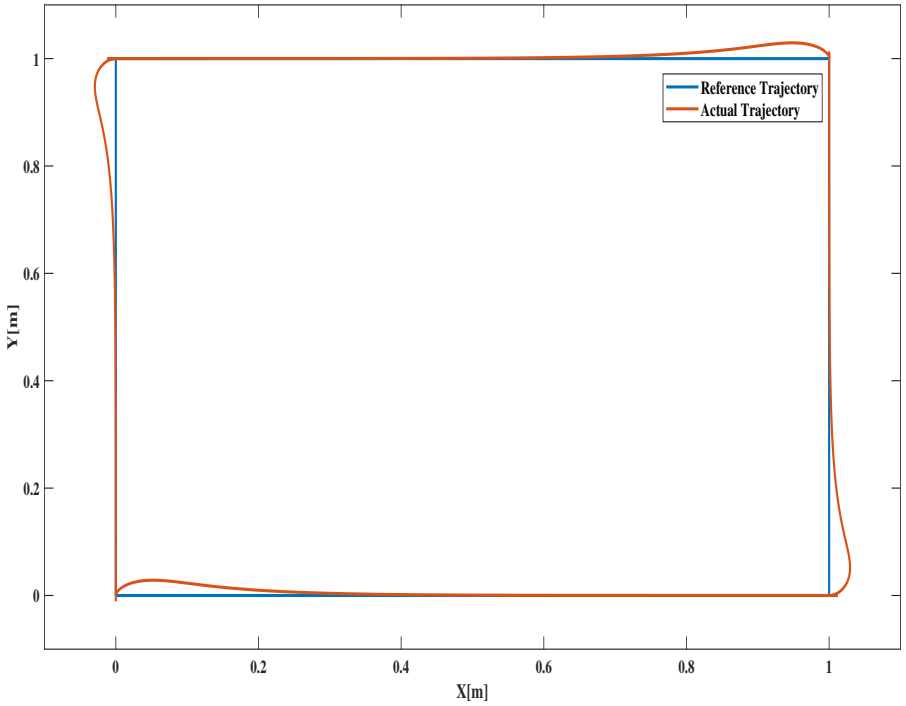


Figure 5.25: X-Y Plot of Square Trajectory using BSMC

5.2.5.2 Square Trajectory Tracking Using BFSMC

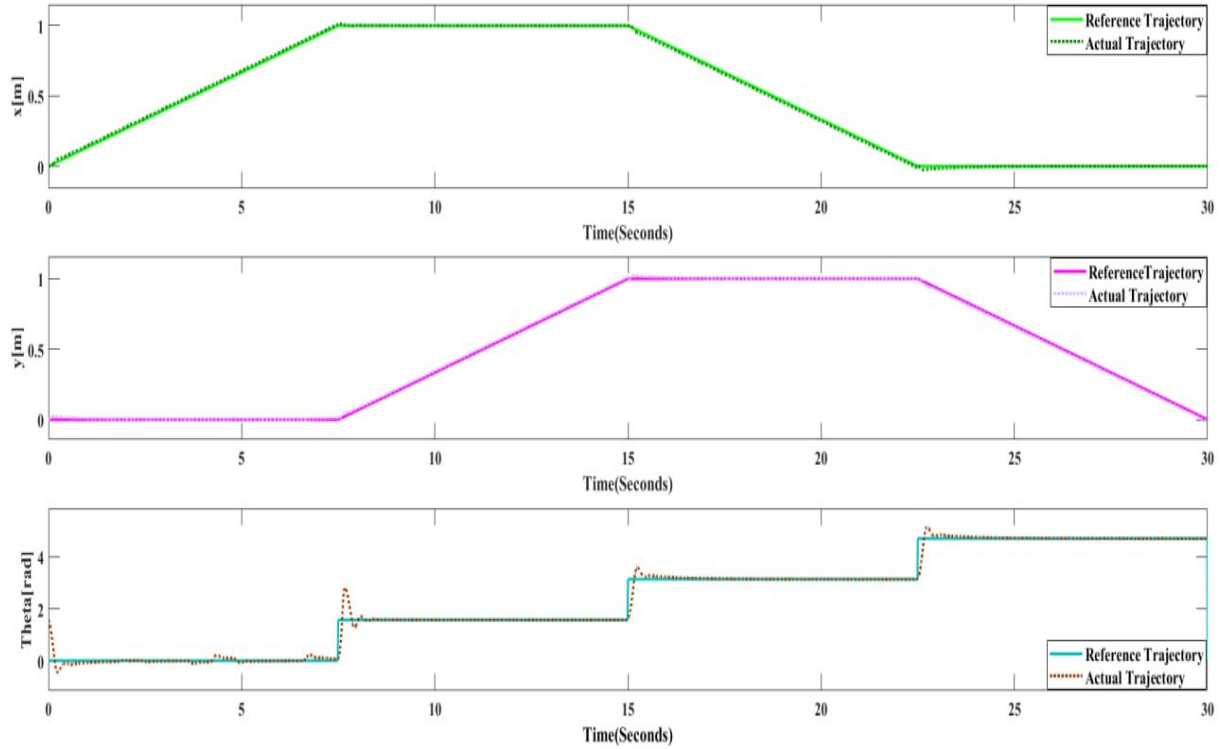


Figure 5.26: Pose for Square Trajectory Tracking Result using BFSMC

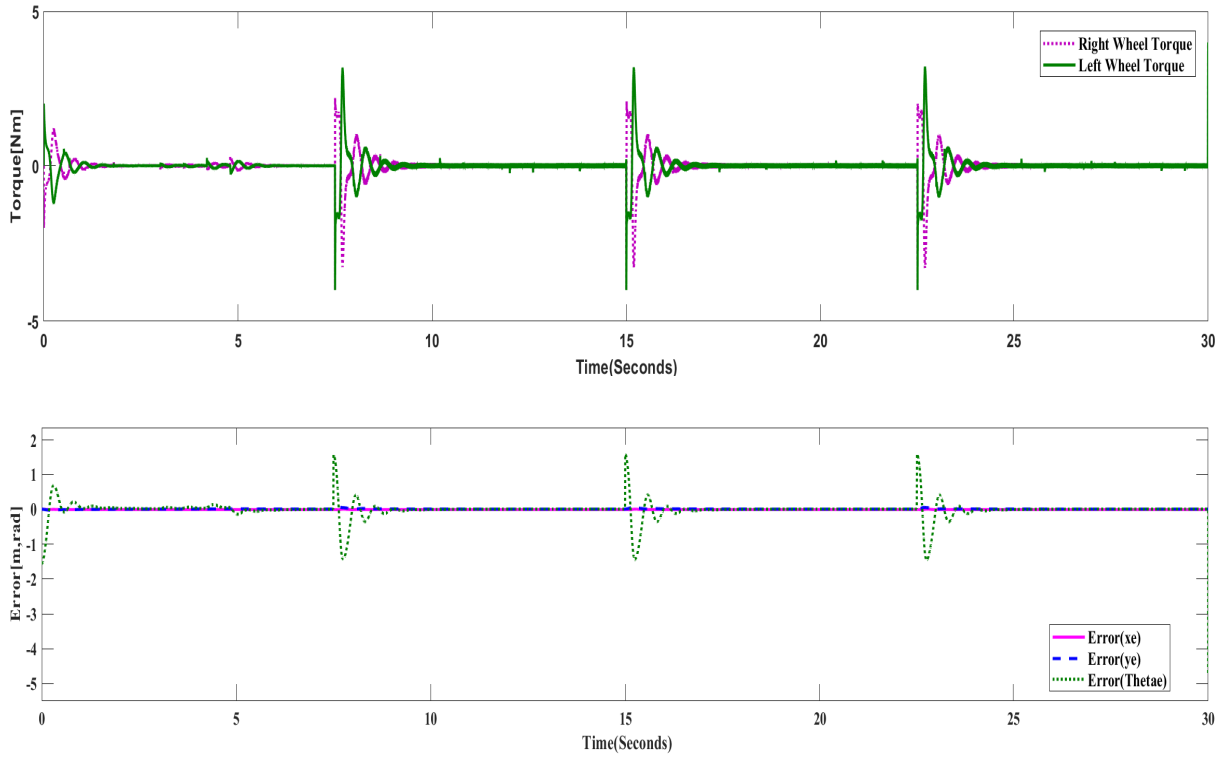


Figure 5.27: Trajectory Tracking Error and Torque for Square Trajectory using BFSMC

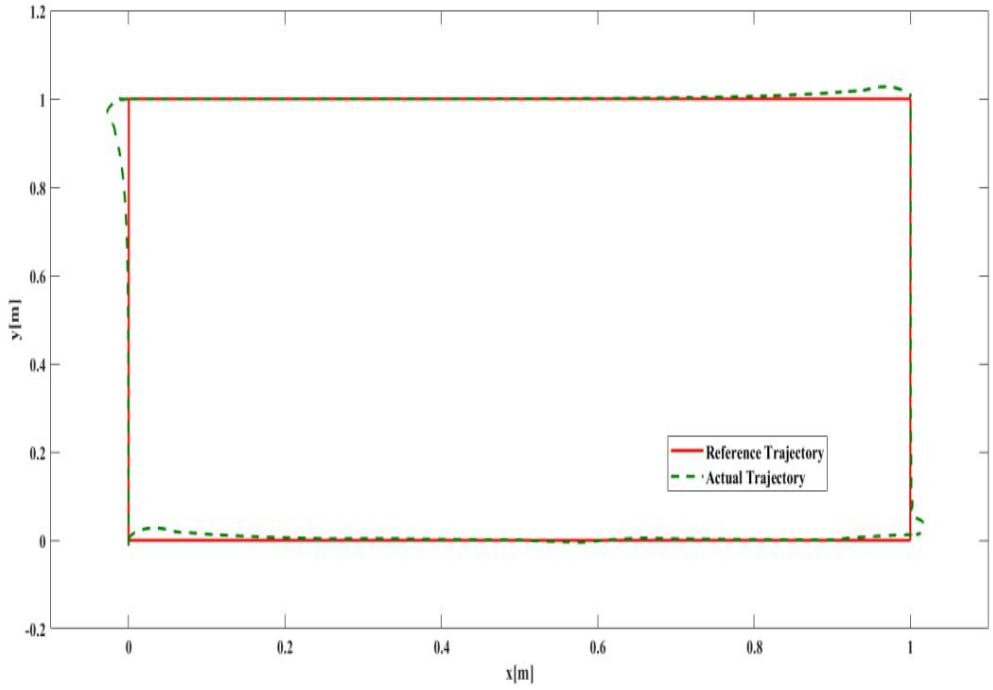


Figure 5.28: X-Y Plot of Square Trajectory Tracking Using BFSMC

5.3 Extended Kalman Filter Simulation Result

In this section, an analysis is undertaken to examine the performance of state estimation for three wheeled mobile robots (TWMRs) using the extended kalman filter (EKF). The analysis covers two scenarios: one with the presence of noise and another without noise. The primary goal of employing the EKF within this system is the mitigation of sensor noise. The investigation seeks to ascertain whether the introduced EKF successfully alleviates the impact of noise within the system. To this end, white noise is introduced into the actual trajectory for the purpose of evaluation.

white noise added to the system implies that a random and constant-intensity noise is introduced to the system's measurements or sensor readings. This addition of white noise helps simulate the effects of real-world measurement uncertainties and disturbances, enabling the evaluation of the EKF's performance in noisy conditions.

5.3.1 State Estimations for Circular Trajectory using EKF

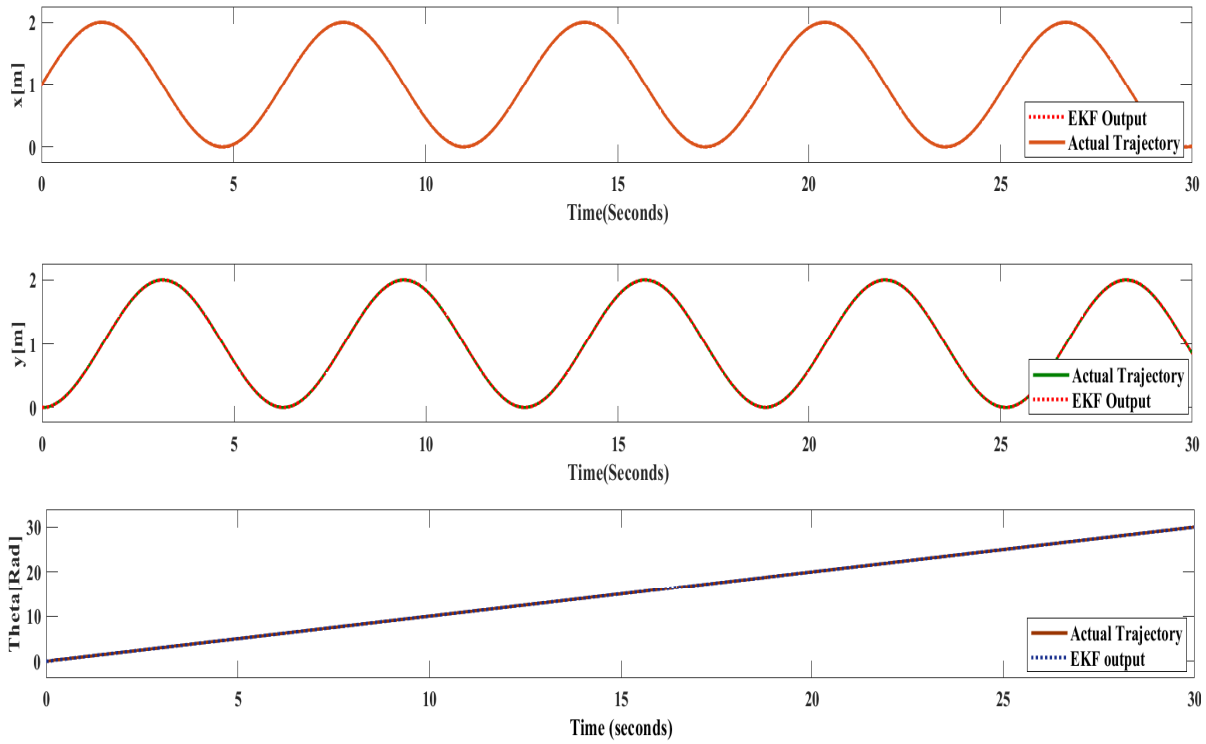


Figure 5.29: Pose Estimation of TWMR Without Addition of Noise

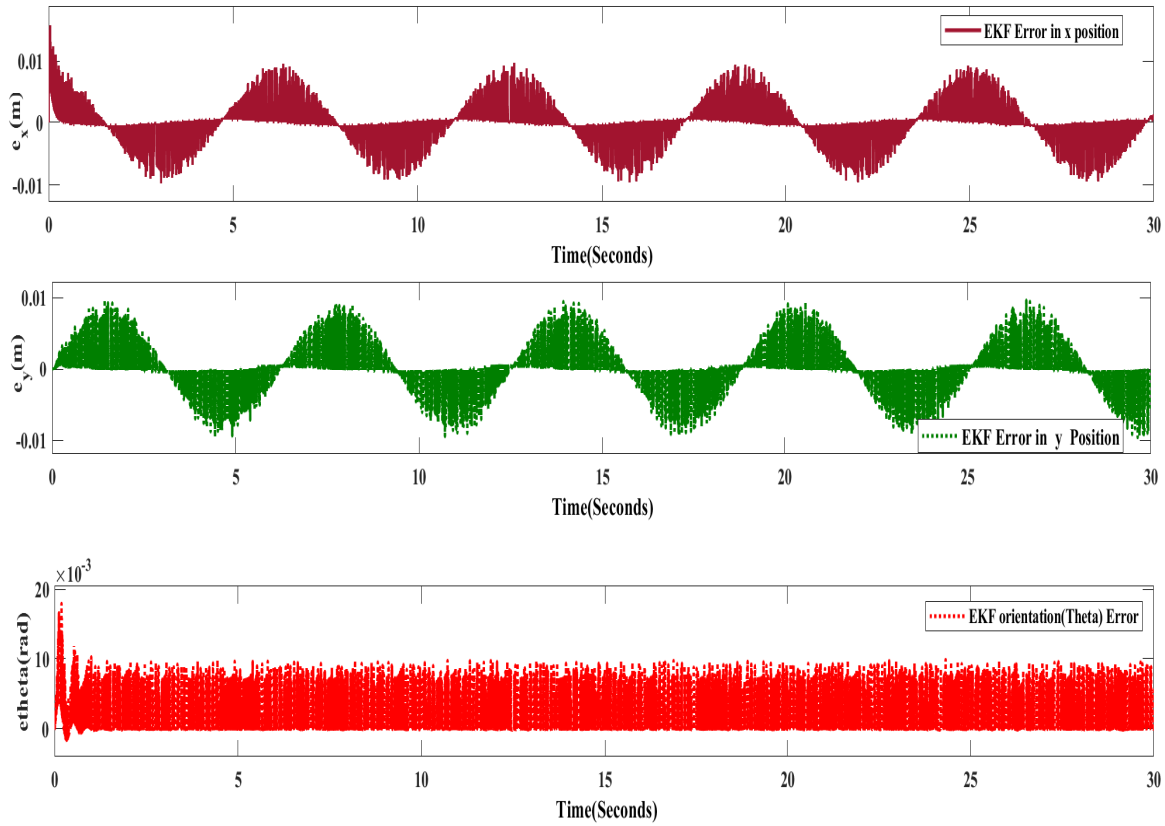


Figure 5.30: Pose Estimation Error of TWMR Without addition of Noise

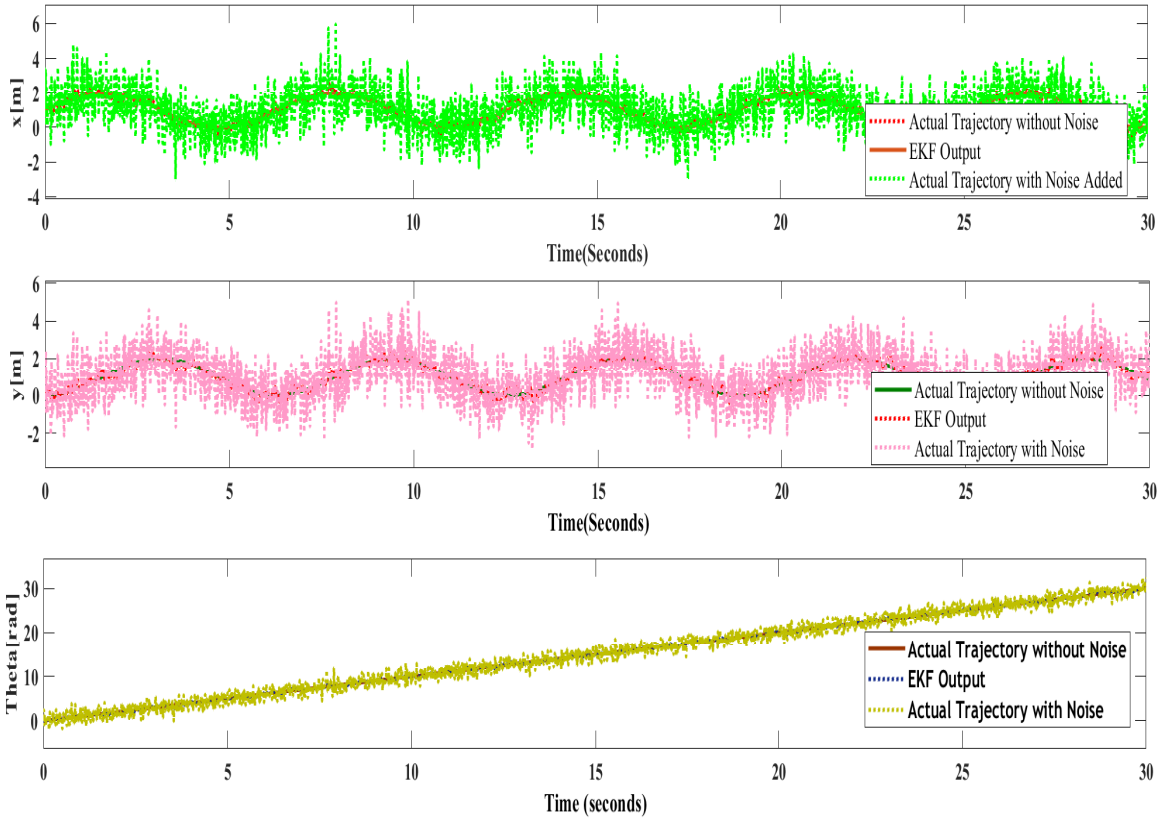


Figure 5.31: Pose Estimation TWMR after Addition of Noise

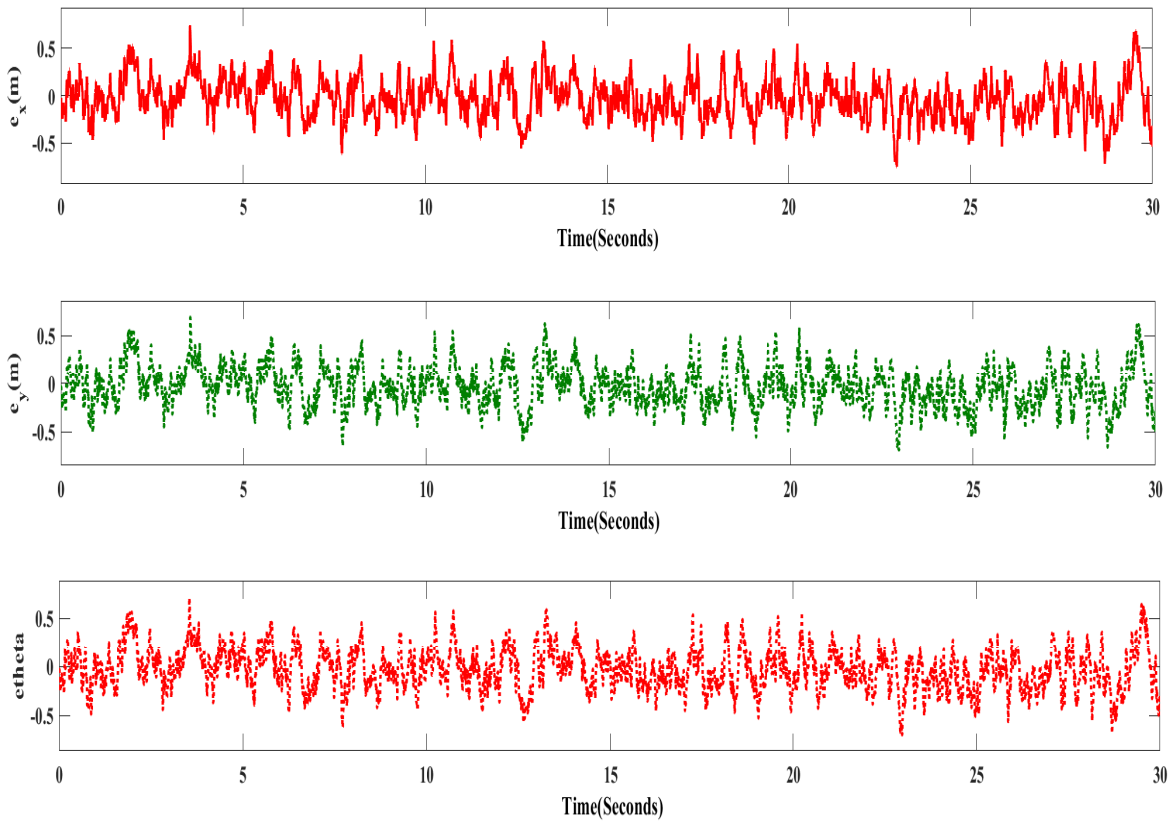


Figure 5.32: Pose Estimation Error of TWMR after Addition of Noise

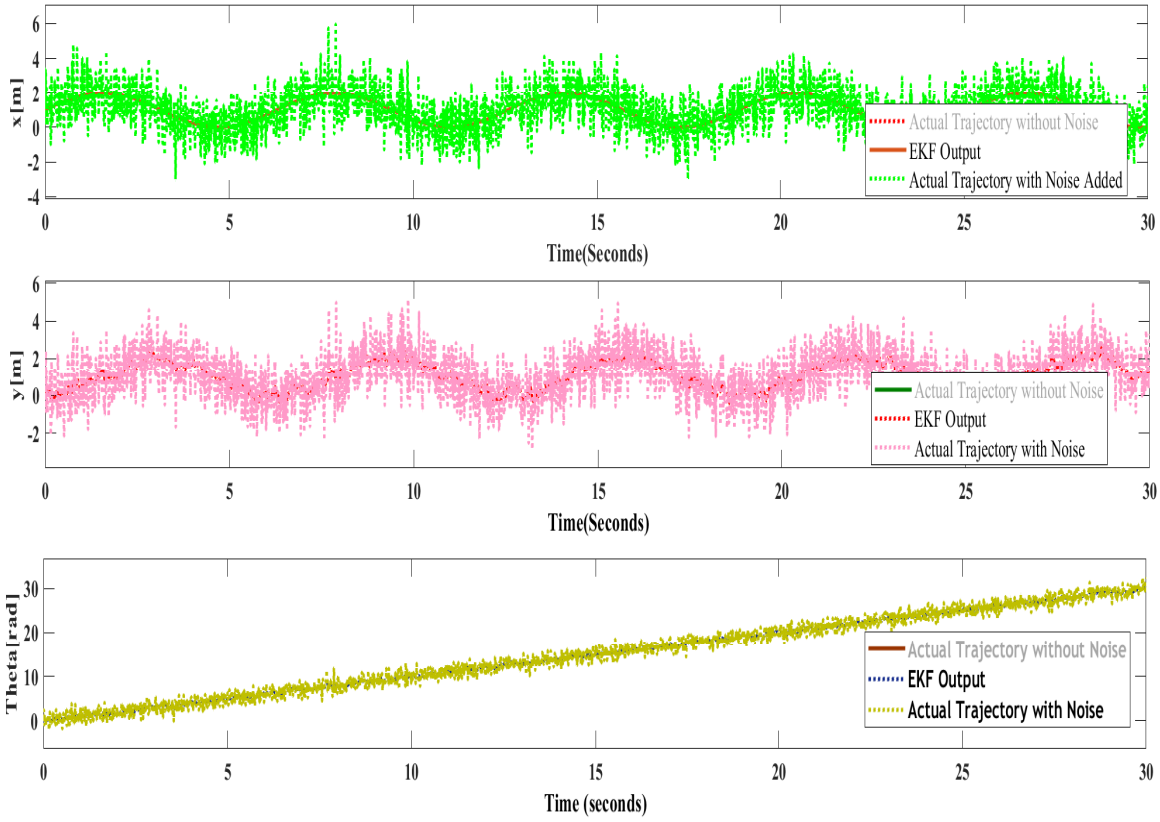


Figure 5.33: Pose Actual Trajectory versus EKF Output Trajectory after Noise Addition

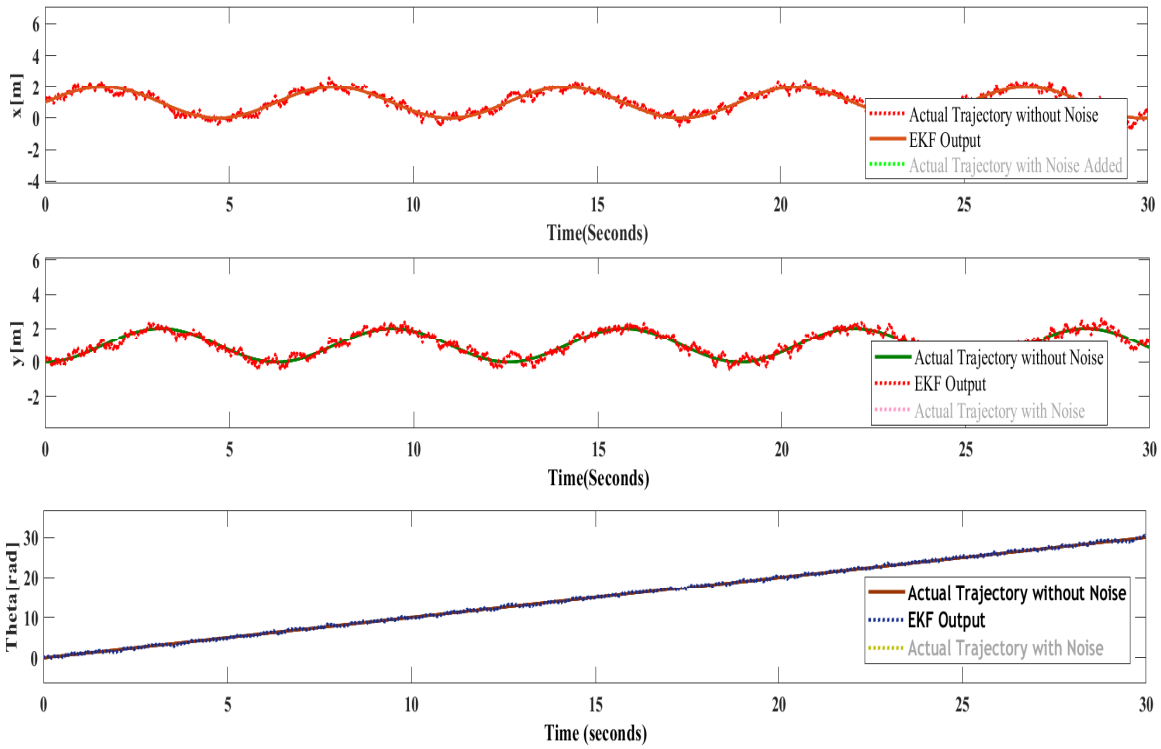


Figure 5.34: Pose Actual Trajectory versus EKF Output Trajectory Without Noise Addition

In the case of a circular trajectory and in the absence of any added noise, it is observed that the EKF yields estimates closely resembling the actual trajectory of the system, as depicted in figure 5.29. This outcome is accompanied by minimal estimation error, as depicted in figure 5.30. Figure 5.31 portrays the system's state estimation, the original trajectory prior to the introduction of noise, and the trajectory after the inclusion of noise. Figure 5.32 demonstrates the error in estimation subsequent to the introduction of white noise. This error remains within the range of -0.5 to 0.5. Furthermore, figure 5.33 illustrates the impact of noise on the actual trajectory when white noise is introduced to the output. Notably, the presence of noise significantly affects the actual trajectory, whereas the EKF effectively mitigates the noise's impact on the system. Figure 5.34 validates the similarity between the actual trajectory before noise addition and the EKF's output subsequent to noise introduction. This validation process aims to ascertain the degree of resemblance between the EKF's estimation and the initial, noise-free trajectory.

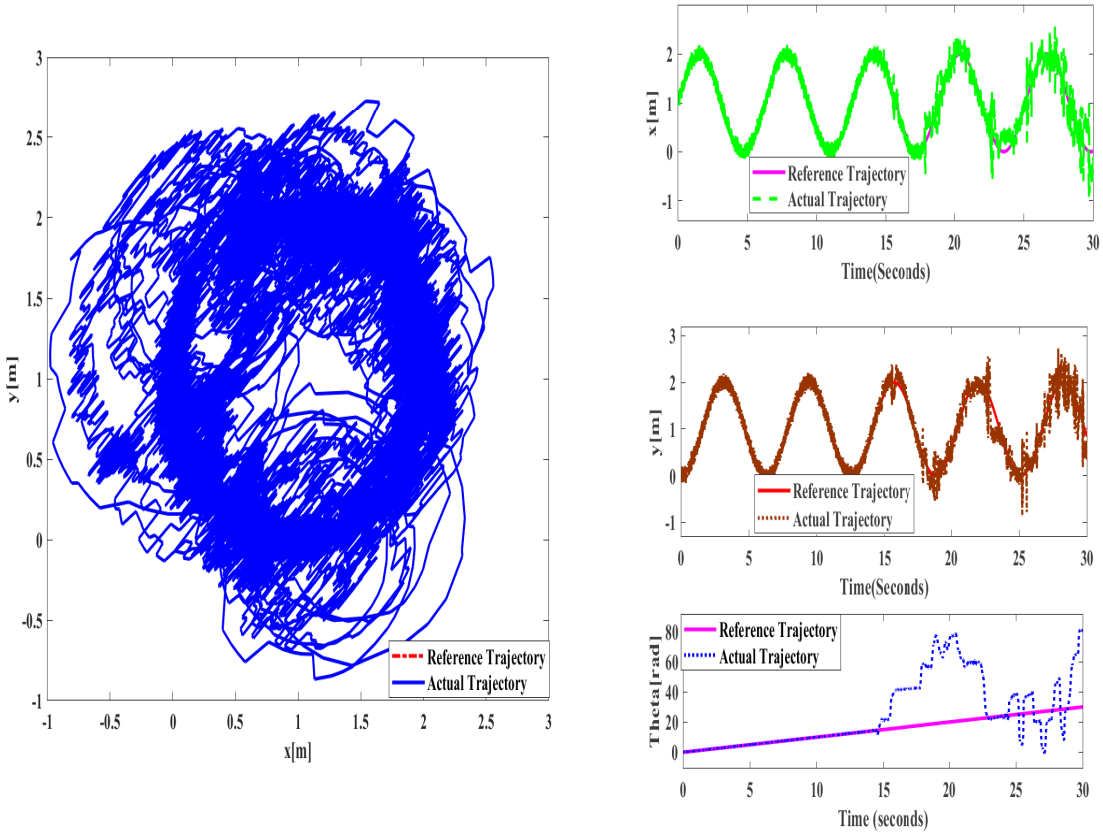


Figure 5.35: Circular Trajectory Tracking with Presence of Noise using BSMC without EKF

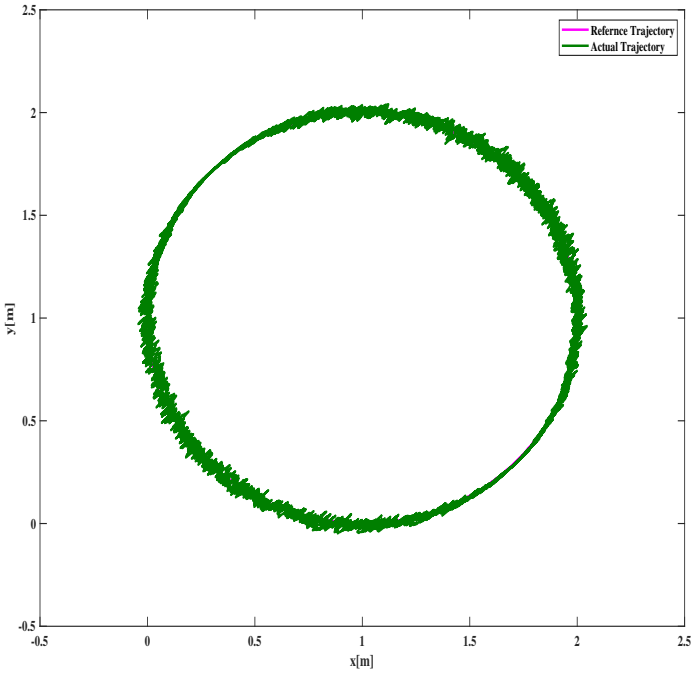


Figure 5.36: Circular Trajectory Tracking with Presence of Noise using BSMC with EKF

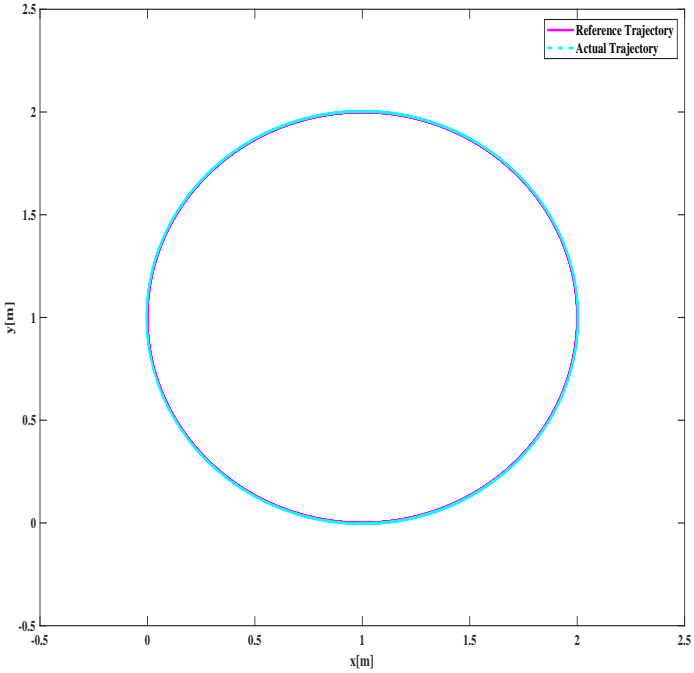


Figure 5.37: Circular Trajectory Tracking without Noise using BSMC with EKF

Figure 5.35 exhibits the impact of introducing white noise to the system's output from start of the simulation for 30 seconds without using EKF only using BSMC . The noise power is specified as 4e-6, and the sample time is set to 0.001. It is evident from the figure that the performance of trajectory tracking is not worse till 13 seconds , but after 14 seconds performance of trajectory tracking deteriorates. This degradation is clearly visualized in figure 5.35. Moving to figure 5.36, the simulation results depict the scenario where the same level of noise is added to the system's output, but an extended kalman filter (EKF) is introduced with BSMC. As illustrated in the figure, the implementation of the EKF mitigates the effects of the noise, enabling the system to resist its influence more effectively. Furthermore, figure 5.37 illustrates the simulation results when no noise is present, but the system still incorporates the BSMC with EKF. From this figure, it is evident that even in the absence of noise, the EKF performs well and exhibits good system performance.

5.4 Simulation Result Comparison BSMC and BFSMC

Performance Indices

When evaluating trajectory tracking performance for a wheeled mobile robot, several performance indices can be used, including ITAE , ISE , and IAE . Each index emphasizes different aspects of the tracking performance.

Integral of Time multiplied by Absolute Error(ITAE): ITAE places more weight on reducing the tracking error during the initial phase of the trajectory. It aims to minimize the integral of the product of the absolute error and the time. This index is suitable when the response time is critical, and reducing the error in the early stages is more important.

Integral of Squared Error(ISE): ISE focuses on minimizing the integral of the squared error over the entire trajectory. It provides a measure of the overall deviation from the desired trajectory. This index is commonly used when the error magnitude needs to be controlled, and the emphasis is on reducing the error throughout the entire trajectory.

Integral of Absolute Error(IAE): IAE is similar to ISE but uses the absolute error instead of squared error. It calculates the integral of the absolute error over the trajectory. IAE provides a measure of the cumulative absolute deviation from the desired trajectory and is suitable when the error magnitude needs to be controlled without the emphasis on squaring the error terms.

MSE(Mean Square Error): Better comparison between the tracking performances of the controllers can be achieved by calculating the trajectory error based on the following equation

$$\text{MSE} = \frac{1}{T}\int_0^T e^2 dt \tag{5.2}$$

5.4.1 Chattering Comparison

The analysis of chattering phenomena in the context of backstepping sliding mode control (BSMC) and backstepping fuzzy sliding mode control (BFSMC) was conducted. In the case of circular trajectories, simulation results are presented in figure 5.2 for BSMC and figure 5.5 for BFSMC, a notable difference in control effort and chattering becomes evident as indicated in figure 5.38,with BSMC the control effort is between (0.04, -0.04) Nm, whereas BFSMC significantly reduces this to between (0.0025, -0.0025) Nm. This observation underscores the effectiveness of BFSMC in reducing control effort and chattering compared to BSMC. Similarly, for square trajectory simulations, figure 5.24 and figure 5.27 depict the results for BSMC and BFSMC, respectively. These figures clearly illustrate a reduction in chattering

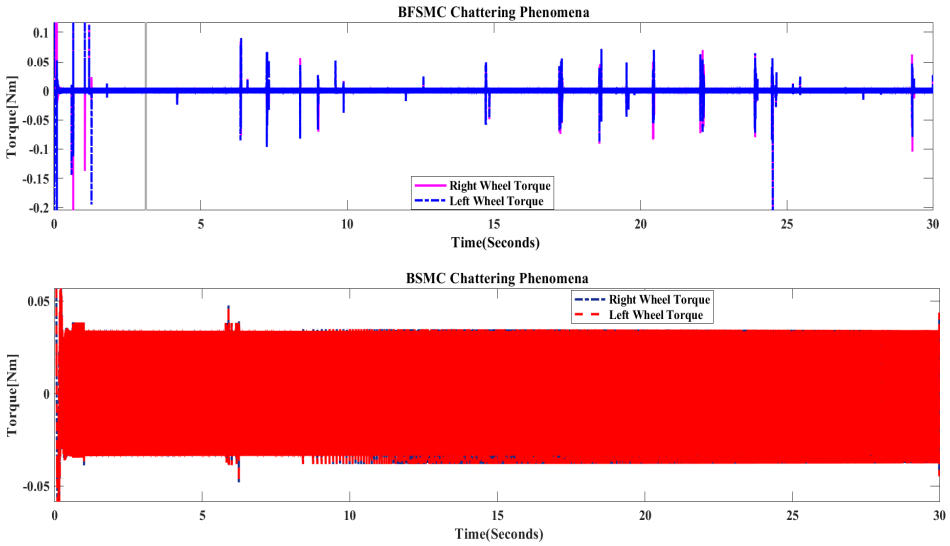


Figure 5.38: Circular Trajectory Chattering Comparison between BSMC and BFSMC

when BFSMC is employed in comparison to BSMC. This reduction in chattering highlights the superior performance of BFSMC in managing complex trajectory scenarios, and also for stair trajectory simulations, figure 5.14 and figure 5.16 depict the results for BSMC and BFSMC, respectively. These figures clearly illustrate a reduction not only magnitude difference but also reduced chattering when BFSMC is employed in comparison to BSMC. This reduction in chattering highlights the superior performance of BFSMC.

5.4.2 Under Nominal Conditions Trajectories Comparison

Infinity Trajectory

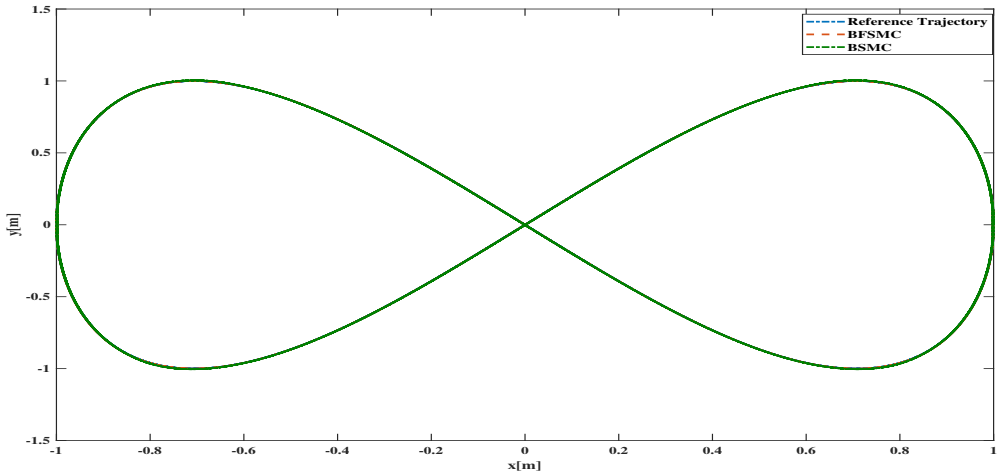


Figure 5.39: Infinity Trajectory Tracking Performance Comparison BSMC and BFSMC

Spiral Trajectory

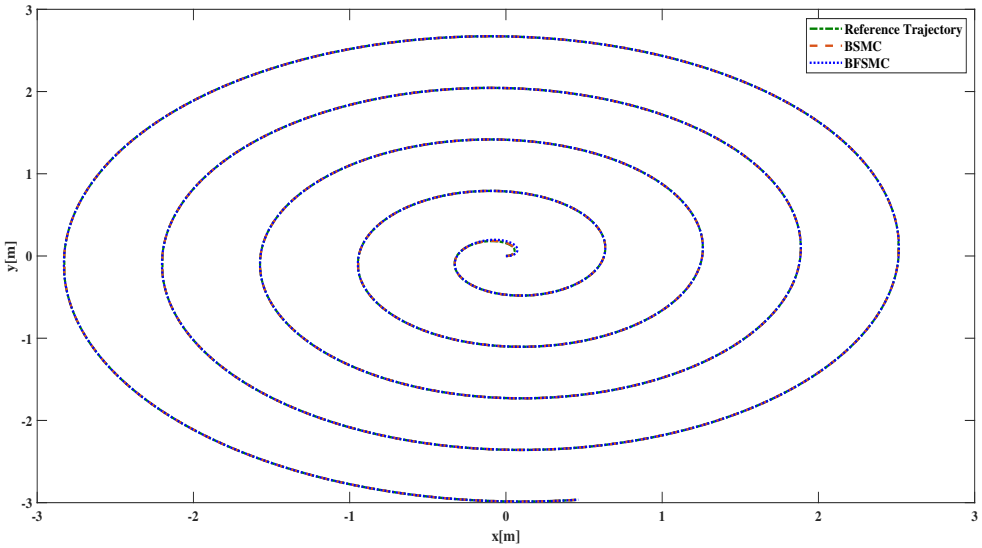


Figure 5.40: Spiral Trajectory Tracking Performance Comparison BSMC and BFSMC

Figure 5.39 and figure 5.40 shows infinity trajectory and spiral trajectory plots respectively under nominal conditions for BSMC and BFSMC

5.4.2.1 Performance Indices Comparison between BFSMC and BSMC for Various Trajectories

Table 5.1: Performance Comparison of BSMC and BFSMC Under Nominal Conditions.

Trajectories	BSMC			BFSMC		
	ITAE	IAE	ISE	ITAE	IAE	ISE
Circular Trajectory	1.2815×10^{-2}	1.19165×10^{-2}	1.396×10^{-4}	7.057×10^{-3}	5.7238×10^{-3}	1.289×10^{-5}
Infinity Trajectory	3.513	0.235	4.589×10^{-3}	0.6206	4.208×10^{-2}	6.14×10^{-5}
Spiral Trajectory	3.98805×10^{-1}	3.30423×10^{-1}	4.05274×10^{-2}	6.8624×10^{-2}	5.2625×10^{-2}	1.0087×10^{-3}

The table 5.1 presents a comparative analysis of the performance of two controllers, BSMC and BFSMC, for various trajectory patterns under nominal conditions. It employs performance metrics including ITAE, IAE, and ISE. In all trajectory types, namely Circular, Infinity, and Spiral, BFSMC demonstrates superior performance with lower ITAE, IAE, and ISE values compared to BSMC. This indicates that BFSMC excels in trajectory tracking, providing more accurate tracking results and reduced tracking errors under nominal conditions.

5.4.2.2 Trajectory Tracking Improvement Comparison of BSMC and BFSMC

$$\text{Percentage Trajectory Tracking Improvement} = \frac{P_{BSMC} - P_{BFSMC}}{P_{BSMC}} \times 100 \tag{5.3}$$

where as:

- P_{BFSMC} is the performance metric(ITAE , IAE , ISE) obtained with BFSMC.
- P_{BSMC} is the performance metric(ITAE , IAE , ISE) obtained with BSMC.

Table 5.2: Percentage of performance Improvement by BFSMC over BSMC

Trajectories	ITAE(%)	IAE(%)	ISE(%)
Circular Trajectory	44.93	51.97	90.76
Infinity Trajectory	82.3	82.09	98.66
Spiral Trajectory	82.7	84.073	97.5

Table 5.2 provides a comparative analysis of the performance improvement achieved by BFSMC over BSMC in various trajectory tracking scenarios. Notably, BFSMC exhibits substantial enhancements in performance compared to BSMC. For instance, in the context of circular trajectory tracking, BFSMC demonstrates a remarkable 44.93% improvement in ITAE, a 51.97% improvement in IAE, and an impressive 90.76% improvement in ISE when compared to BSMC. This trend continues in more complex trajectory shapes like infinity and spiral, with BFSMC consistently delivering improvements exceeding 82% across all evaluated performance indices. These results underscore the efficacy of BFSMC in significantly enhancing trajectory tracking precision and stability when compared to BSMC.

5.4.3 Disturbance and Uncertainty of Unmodeled Dynamics

The disturbance analysis of the BFSMC for wheeled mobile robots involves evaluating the controller's ability to reject disturbances and maintain the desired performance under various scenarios. from control point of view modeling inaccuracies can be classified into two major kinds.

1. Structured Parametric Uncertainly
2. Unstructured parametric uncertainty or unmodeled dynamics:- E.g this is due to friction between the wheel of mobile robot and the ground.

Parameter Variation

Wheeled mobile robots operate in dynamic and uncertain environments where variations in parameters, such as mass, friction coefficients, and inertia, are inevitable. Testing the robot's performance under parameter variations helps ensure its robustness and ability to adapt to different operating conditions. It enables the identification of critical parameters that significantly affect the system's behavior and allows for the development of control strategies that can handle such variations.

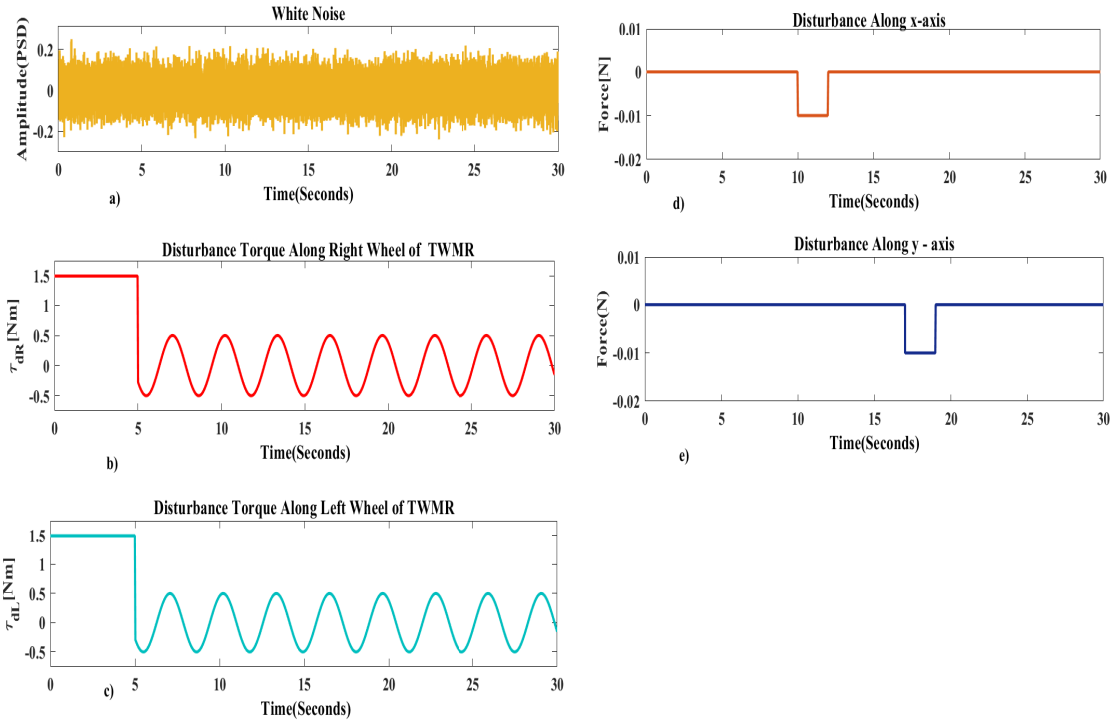


Figure 5.41: White Noise , External disturbance , and Disturbance Torque

5.4.4 Robustness Analysis Between BSMC and BFSMC

5.4.4.1 Infinity Trajectory

To assess the robustness of both BFSMC and BSMC for TWMRs, simultaneously disturbance torques and parameter variations related to mass and inertia introduced into the designed system. Equation (5.4) indicates that the disturbance torque and parameter variation introduced into the system. Figure 5.41b and figure 5.41c indicates that disturbance added to the system.

$$\begin{cases} m & 10kg \rightarrow 15kg \\ I & 5kgm^2 \rightarrow 7.5kgm^2 \end{cases} \qquad \tau_d = [\tau_{dR} \quad \tau_{dL}]^T = \begin{cases} 1.5 & \text{for } t < 5 \\ 0.5\sin(2t) & \text{for } t >= 5 \end{cases} \tag{5.4}$$

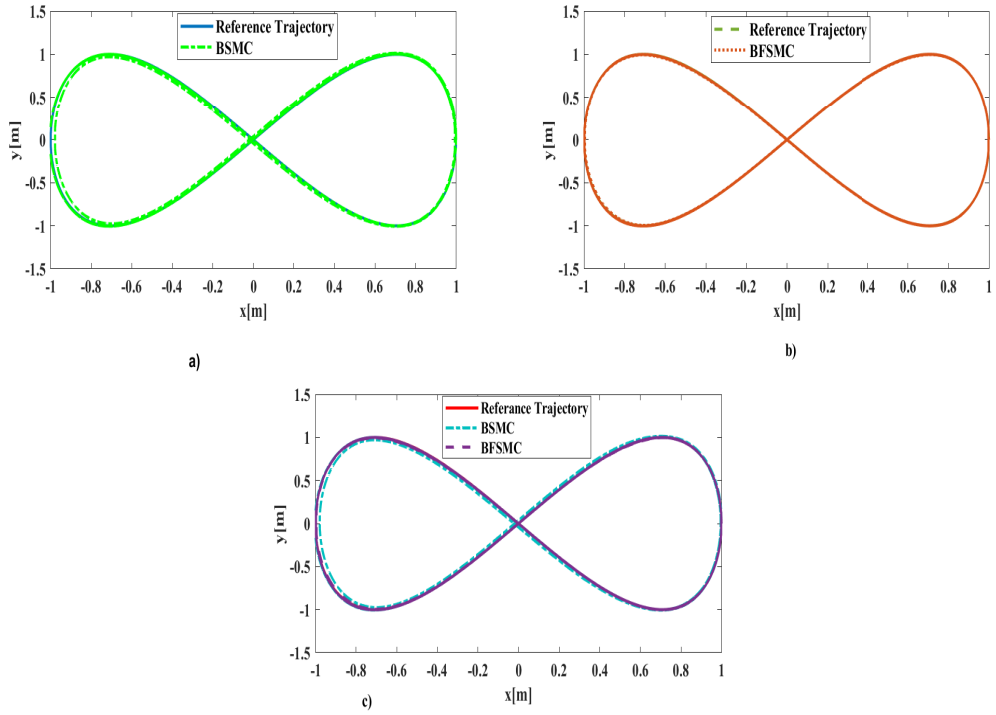


Figure 5.42: Infinity Trajectory Tracking Comparison between BSMC and BFSMC with Parameter Variation and Disturbance

The results depicted in figure 5.42a represent the outcomes when BSMC is utilized, figure 5.42b illustrates the results obtained with BFSMC, and figure 5.42c shows the combined trajectories plots reference trajectory , BFSMC and BSMC. It is evident from the results that both BFSMC and BSMC are influenced by the introduction of disturbances and parameter variations. However, BFSMC demonstrates a relatively lower impact compared to BSMC.

Table 5.3: Performance Comparison of BSMC and BFSMC for Infinity Trajectory Tracking

Controller	Under nominal conditions			With Disturbance and parameter Variation		
	ITAE	IAE	ISE	ITAE	IAE	ISE
BSMC	3.513	0.235	4.589×10^{-3}	7.129	0.7606	2.861×10^{-2}
BFSMC	0.6206	4.208×10^{-2}	6.14×10^{-5}	1.439	0.1353	5.685×10^{-4}

Table 5.3 in nominal scenarios, BFSMC better than BSMC with notably lower values for critical performance metrics such as ITAE, IAE, and ISE, indicating superior trajectory

tracking precision. When subject to disturbances and parameter variations, both controllers experience performance degradation; however, BFSMC maintains its supremacy with consistently lower ITAE, IAE, and ISE values compared to BSMC. These findings underline BFSMC's robustness and effectiveness in enhancing trajectory tracking in dynamic and uncertain environments relative to BSMC.

5.4.4.2 Square Trajectory

To assess the ability of both BFSMC and BSMC to reject external disturbances, external disturbances were introduced along the position coordinates (x, y) of the square trajectories, as described in equation (5.5). Figure 5.41d and figure 5.41e indicates that disturbance added to the system.

$$\begin{aligned} f_{xd} &= -0.01, \quad \text{for } t \geq 10 \text{ and for } t \leq 12 \\ f_{yd} &= -0.01, \quad \text{for } t \geq 17 \text{ and for } t \leq 19 \end{aligned} \tag{5.5}$$

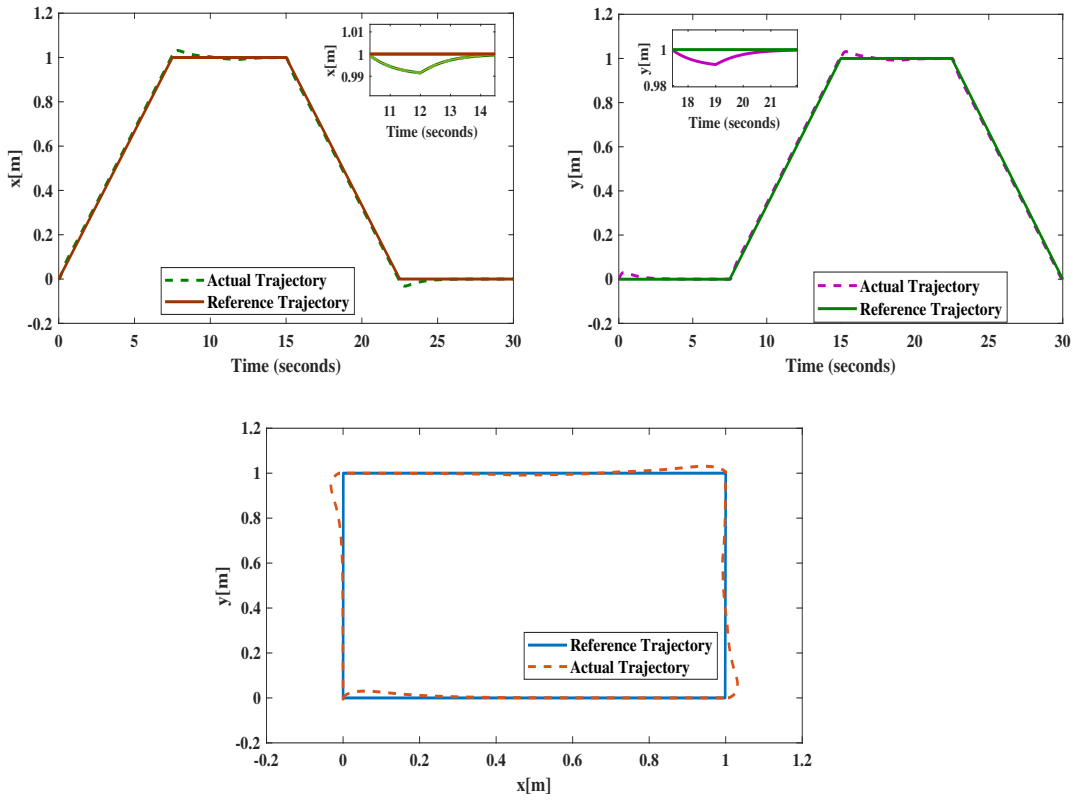


Figure 5.43: Square Trajectory Tracking after Disturbance Added to the System using BSMC

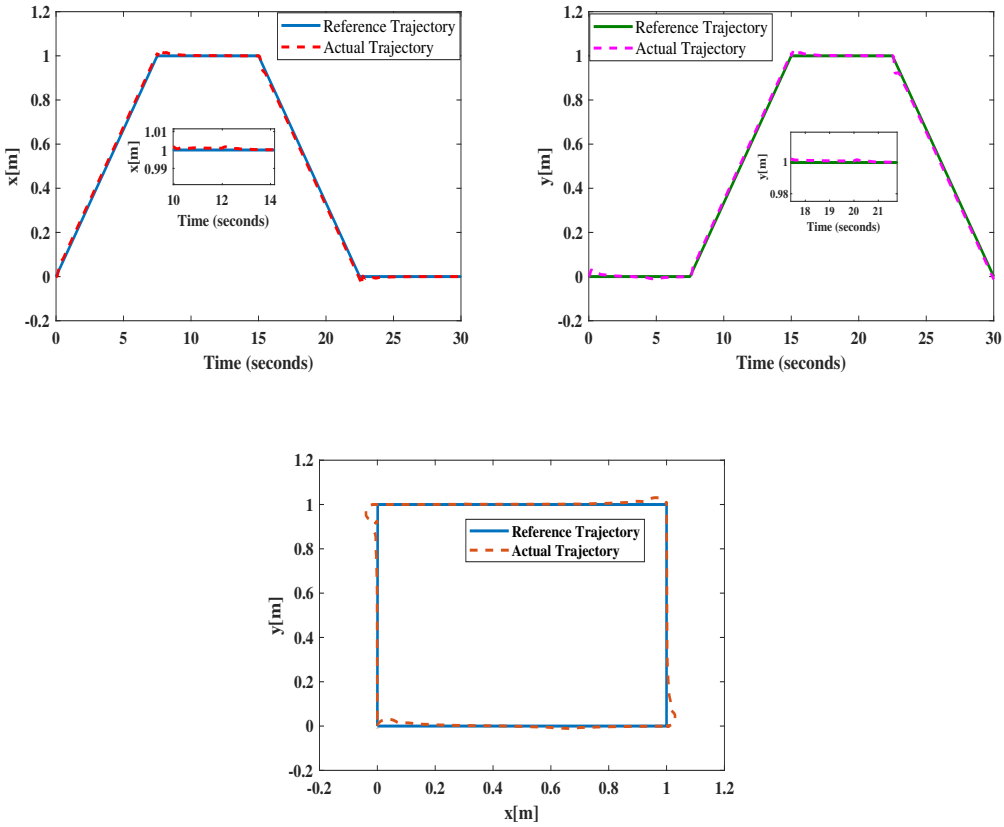


Figure 5.44: Square Trajectory Tracking After Disturbance Added to the System using BFSMC

As shown in figure 5.43, simulation results for the square trajectory employing BSMC Presented, considering the introduction of disturbances into the system dynamics. In contrast, figure 5.44 displays the simulation outcomes for the same square trajectory using BFSMC, also with disturbances in the system dynamics.

The simulation findings clearly indicate that both BFSMC and BSMC effectively ensure system stability, leading to the convergence of the robot's position to the desired values. However, it is noteworthy that BFSMC exhibits superior disturbance rejection capabilities, resulting in errors converging to zero more effectively when compared to BSMC

5.4.4.3 Circular Trajectory

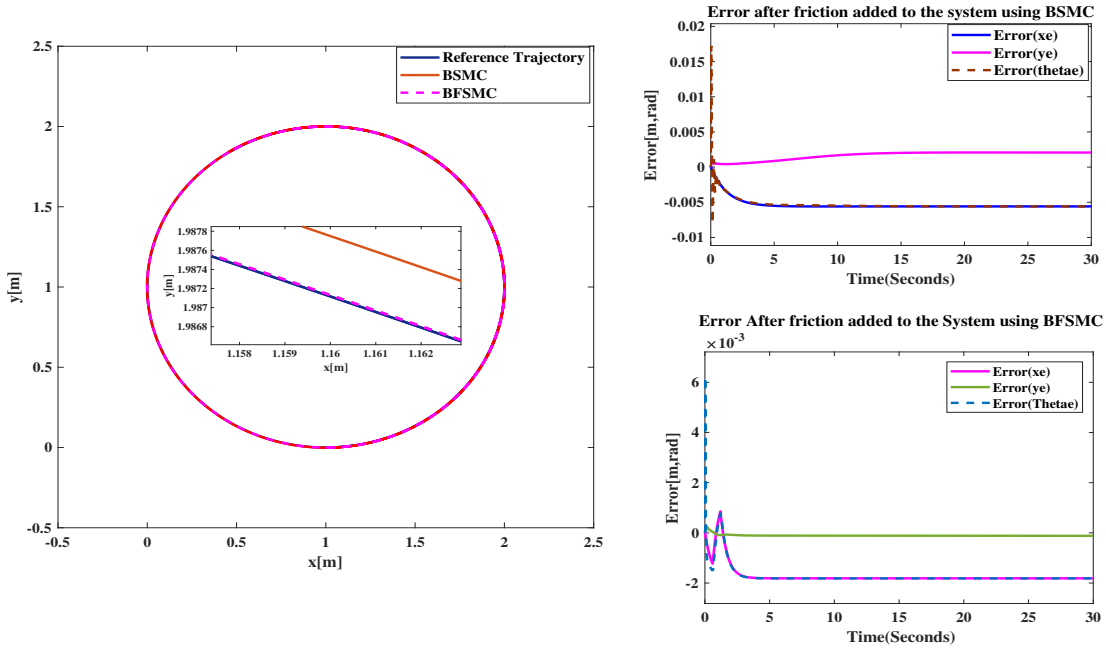


Figure 5.45: Circle Trajectory Tracking after Friction Introduced to the System using BFSMC and BSMC Comparison

In control systems, friction is frequently regarded as an unmodeled disturbance, introducing nonlinearities into system dynamics and potentially causing errors in position and velocity control. To assess the impact of friction, friction forces, specifically 0.1 Nm on the right wheel and 0.09 Nm on the left wheel, were added.

Figure 5.45 depicts a comparison between BFSMC and BSMC in terms of trajectory tracking and trajectory error when friction is introduced to the systems. The figures illustrate that BFSMC is more resilient to the effects of friction on the system compared to BSMC. Additionally, the trajectory tracking error of BFSMC is lower than that of BSMC.

Table 5.4: Performance Comparison of BSMC and BFSMC for Circular Trajectory Tracking with the Presence of Friction Forces

Controller	Under nominal conditions			With Friction		
	ITAE	IAE	ISE	ITAE	IAE	ISE
BSMC	1.2815×10^{-2}	1.19165×10^{-2}	1.396×10^{-4}	5.879	0.3714	1.853×10^{-3}
BFSMC	7.057×10^{-3}	5.7238×10^{-3}	1.289×10^{-5}	1.676	0.1073	1.861×10^{-4}

The presented results in table 5.4 provide a comprehensive evaluation of the (BSMC) and (BFSMC) using the following performance indices: ITAE , IAE , and ISE . The evaluation

covers two scenarios: one Under nominal conditions and the other with friction forces (With Friction). In the absence of friction, BFSMC better than BSMC, exhibiting significantly lower ITAE, IAE, and ISE values. This underscores BFSMC's superior trajectory tracking capabilities and robustness in nominal conditions. When friction forces are introduced, both controllers experience performance degradation, but BFSMC maintains its lead over BSMC with notably lower ITAE, IAE, and ISE values, emphasizing BFSMC's resilience and disturbance-rejection capabilities.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In conclusion, this thesis has undertaken a comprehensive exploration of trajectory tracking enhancement for three-wheeled mobile robots (TWMRs) by utilizing the backstepping fuzzy sliding mode controller (BFSMC) within a framework encompassing thorough modeling, control design, and performance evaluation. The research journey initiated with the derivation and careful verification of the kinematic, dynamic, and actuator models for TWMRs. These foundational models were subjected to strict testing and validation to ensure their accuracy and suitability for subsequent control strategy development. The core contributions of this research include the design and implementation of two distinct control methodologies: the backstepping fuzzy sliding mode controller (BFSMC) and the backstepping sliding mode controller (BSMC). These controllers were diligently designed to facilitate precise trajectory tracking, a challenging task due to the non-holonomic nature of TWMRs. To further enhance the system's performance, an extended kalman filter (EKF) was implemented. The EKF served a crucial role in mitigating sensor noise, significantly enhancing the accuracy of position estimation. Fine-tuning the controllers to optimize their performance was achieved through the application of particle swarm optimization (PSO). This optimization process ensured that both BFSMC and BSMC were finely tuned to extract the maximum potential of trajectory tracking accuracy. Performance evaluation was conducted using a set of robust performance indices, including ITAE, IAE, and ISE. MATLAB, a powerful computational tool, was employed for simulation of designed systems. The comparative analysis between BFSMC and BSMC encompassed diverse trajectory tracking scenarios, including the introduction of real-world challenges such as sensor noise, disturbances, parameter vari-

ations, and even frictional forces. The outcomes consistently demonstrated superiority for BFSMC, which exhibited superior trajectory tracking enhancements across various conditions. Notably, BFSMC demonstrated its effectiveness in reducing chattering and control effort. In summary, the findings of this research underscore the superior trajectory tracking capabilities of BFSMC, particularly when coupled with the EKF for noise reduction.

6.2 Future Work

The research has underscored the remarkable trajectory tracking performance accomplished through the integration of the backstepping fuzzy sliding mode controller (BFSMC) and the extended kalman filter (EKF). Nevertheless, there are limitations associated with relying on fixed gains, which have been optimized using particle swarm optimization (PSO). These limitations become particularly apparent when the robots operate in unfamiliar environments. Additionally, to enhance trajectory tracking performance, it is crucial to integrate advanced sensor technologies such as inertial measurement units (IMUs), light detection and ranging (LIDAR) systems, and cameras. These sensors should be fused together to strengthen the system's resilience. Furthermore, implementing these advancements in hardware is a vital step towards practical application. In light of these conclusions, the following concise recommendations are proposed:

- Sensor fusion: Integrate IMUs, LIDAR, and cameras, and implement advanced sensor fusion techniques to boost navigation accuracy and reliability.
- Dynamic gain optimization.
- Machine learning integration: To facilitate adaptive control and decision-making in complex and unstructured environments.
- Integration of actuator model with designed system and also hardware implementation.

Bibliography

[1] Nikolaus Correll, Bradley Hayes, Christoffer Heckman, and Alessandro Roncone. *Introduction to Autonomous Robots: Mechanisms, Sensors, Actuators, and Algorithms*. MIT Press, Cambridge, MA, 1st edition, 2022.

[2] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[3] George C Devol and Joseph F Engelberger. Development of the first robot in the world. *IEEE Transactions on Robotics*, 1(1):4–8, 1985.

[4] Nils J Nilsson. *Principles of Artificial Intelligence*. Springer Science & Business Media, 2014.

[5] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. Springer, 2008.

[6] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[7] Johann Borenstein, H. R. Everett, and Liqiang Feng. *Navigating Mobile Robots: Systems and Techniques*. A K Peters/CRC Press, 1996.

[8] Roohollah Barzamini and A. Afshar. Dynamic adaptive tracking control for wheeled mobile robot. 2005.

[9] E. F. Wijn. Unstable behavior of a unicycle mobile robot with tracking control. 2004.

[10] Gregor Klančar and Igor Škrjanc. Tracking-error model-based predictive control for mobile robots in real time. *Robotics and Autonomous Systems*, 55(6):460–469, 2007.

[11] Prabin Kumar Panigrahi and Sukant Kishoro Bisoy. Localization strategies for autonomous mobile robots: A review. *Journal of King Saud University - Computer and Information Sciences*, 34(8, Part B):6019–6039, 2022.

[12] Pioneer 3-Dx. https://robots.ros.org/pioneer-3-dx/.

[13] Roland Siegwart and Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press, Cambridge, MA, 1st edition, 2004.

[14] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[15] J. S. Jang. Anfis: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993.

[16] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.

[17] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(1):28–44, 1973.

[18] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995.

[19] Valery I Utkin. Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, 22(2):212–222, 1977.

[20] Jean-Jacques E Slotine and Wei Li. *Applied Nonlinear Control*. Prentice-Hall, 1991.

[21] V. I Utkin. Sliding mode control design principles and applications to electric drives. *IEEE Transactions on Industrial Electronics*, 40(1):23–36, 1992.

[22] Kalman filter tutorial. https://www.kalmanfilter.net/default.aspx.

[23] Miroslav Krstić, Ioannis Kanellakopoulos, and Petar V Kokotović. *Nonlinear and Adaptive Control Design*. John Wiley & Sons, 1995.

[24] Petros A Ioannou and Jing Sun. *Robust Adaptive Control*. Courier Corporation, 2012.

[25] Jaime A Moreno, Marcos Osorio, Romeo Ortega, and Jean-Pierre Barbot. High-order backstepping control of autonomous underwater vehicles. *IEEE Transactions on Control Systems Technology*, 26(4):1324–1335, 2018.

[26] Yuri Shtessel, Christopher Edwards, Leonid Fridman, and Arie Levant. *Sliding Mode Control and Observation*. Springer Science & Business Media, 2014.

[27] Jean-Jacques E Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice-Hall, 1991.

[28] Jing Sun, Shuzhi Sam Ge, and Tong Heng Lee. Adaptive backstepping control of uncertain systems: Nonsmooth nonlinearities, interactions or time-varying parameters. *IEEE Transactions on Automatic Control*, 48(9):1658–1664, 2003.

[29] James Kennedy and Russell C Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, pages 1942–1948. IEEE, 1995.

[30] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization: An overview. *Swarm Intelligence*, 1(1):33–57, 2007.

[31] Maurice Clerc and James Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[32] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 384–389 vol.1, 1990.

[33] R. Fierro and F.L. Lewis. Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, 9(4):589–600, 1998.

[34] Marvin K. Bugeja and Simon G. Fabri. Dual adaptive control for trajectory tracking of mobile robots. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2215–2220, 2007.

[35] Lianzheng Ge, Ruifeng Li, Dianyong Yu, and Lijun Zhao. A nonlinear adaptive variable structure trajectory tracking algorithm for mobile robots. In Caihua Xiong, Yongan Huang, Youlun Xiong, and Honghai Liu, editors, *Intelligent Robotics and Applications*, pages 892–900, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[36] Zhen-yu Liu, Rui-huan Jing, Xiang-qian Ding, and Jian-hua Li. Trajectory tracking control of wheeled mobile robots based on the artificial potential field. In *2008 Fourth International Conference on Natural Computation*, volume 7, pages 382–387, 2008.

[37] Farzad Pourboghrat and Mattias P. Karlsson. Adaptive control of dynamic mobile robots with nonholonomic constraints. *Computers Electrical Engineering*, 28(4):241–253, 2002.

[38] Turki Younis Abdalla and A Abdulkareem.A. Pso-based optimum design of pid controller for mobile robot trajectory tracking. *International Journal of Computer Applications*, 47:30–35, 2012.

[39] Sang won Kim and Chongkug Park. Optimal tracking controller for an autonomous wheeled mobile robot using fuzzy genetic algorithm. In *ICMIT: Mechatronics and Information Technology*, 2005.

[40] K. Faress, M. hagry, and Ahmed Mahfouz. Trajectory tracking control for a wheeled mobile robot using fuzzy logic controller. 4:9, 07 2005.

[41] Oscar Castillo, Luis Aguilar, and Selene Cardenas-Maciel. Fuzzy logic tracking control for unicycle mobile robots. *Engineering Letters*, 13, 08 2006.

[42] X. Jiang, Y. Motai, and X. Zhu. Predictive fuzzy logic controller for trajectory tracking of a mobile robot. In *Proceedings of the 2005 IEEE Midnight-Summer Workshop on Soft Computing in Industrial Applications, 2005. SMCia/05.*, pages 29–32, 2005.

[43] B Kosko. *Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence*. Prentice-Hall, 1992.

[44] M Krstic, I Kanellakopoulos, and P. V Kokotovic. *Nonlinear and adaptive control design*. Wiley, 1995.

[45] J. J. Slotine and W Li. *Applied nonlinear control*. Prentice-Hall, 1991.

[46] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 18(4):22–34, 1998.

[47] J. D. Wijn. Unstable behavior of a three-wheeled mobile robot. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[48] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning, and Control*. Springer, 2010.

[49] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, 2005.

[50] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2006.

[51] G. Campion, G. Bastin, and B. Dandrea-Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1):47–62, 1996.

[52] Iman Anvari. Non-holonomic differential drive mobile robot control & design: Critical dynamics and coupling constraints. Technical report, Arizona State University, 2013.

[53] Yasmine Koubaa, Mohamed Boukattaya, and Tarak Damak. Adaptive sliding-mode control of nonholonomic wheeled mobile robot. In *2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pages 336–342, 2014.

[54] Omid Mohareri. Mobile robot trajectory tracking using neural networks. Master's thesis, American University of Sharjah, Sharjah, U.A.E., 2009.

[55] Pascal Morin and Claude Samson. *Motion Control of Wheeled Mobile Robots*, pages 799–826. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[56] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 384–389 vol.1, 1990.

[57] Jean-Jacques E Slotine and Shankar S Sastry. Tracking control of nonlinear systems using sliding surfaces, with application to robot manipulators. *International Journal of Control*, 38(2):465–492, 1983.

[58] Amitava Chatterjee and Fumitoshi Matsuno. A neuro-fuzzy assisted extended kalman filter-based approach for simultaneous localization and mapping (slam) problems. *IEEE Transactions on Fuzzy Systems*, 15(5):984–997, 2007.

[59] Eugenio Bargiacchi, Diederik M. Roijers, and Ann Nowé. Ai-toolbox: A c++ library for reinforcement learning and planning (with python bindings). *J. Mach. Learn. Res.*, 21(1), jan 2020.

[60] Mohd Nasir, Nabil Zhafri, Zakaria, Muhammad Aizzat, Razali, Saifudin, and bin Abu, Mohd Yazid. Autonomous mobile robot localization using kalman filter. *MATEC Web Conf.*, 90:01069, 2017.

Appendix A

MATLAB Code

A.1 PSO Code

clc clear

lb = [55 35 0.0001 0.2 0.13 0.12 0.11 0.1 0.4]; ub = [70 60 20 5 10 8 20 3 8]; obfun = @fosmc;

Np = 6; T = 3; w = 0.9; c1 = 2; c2 = 2;

f = NaN(Np, 1); bestfititer = NaN(T + 1, 1); D = length(lb); P = repmat(lb, Np, 1) + repmat((ub - lb), Np, 1) .* rand(Np, D); v = repmat(lb, Np, 1) + repmat((ub - lb), Np, 1) .* rand(Np, D);

for p = 1:Np f(p) = obfun(P(p, :)); end

pbest = P; fpbest = f;

[fgbest, ind] = min(fpbest); gbest = pbest(ind, :); bestfititer(1) = fgbest;

for t = 1:T for p = 1:Np v(p, :) = w * v(p, :) + c1 * rand(1, D) .* (pbest(p, :) - P(p, :)) + c2 * rand(1, D) .* (gbest - P(p, :)); P(p, :) = P(p, :) + v(p, :); P(p, :) = max(P(p, :), lb); P(p, :) = min(P(p, :), ub); f(p) = obfun(P(p, :));

if f(p) ¡ fpbest(p) fpbest(p) = f(p); pbest(p, :) = P(p, :);

if fpbest(p) ¡ fgbest fgbest = fpbest(p); gbest = pbest(p, :); end end end

bestfititer(t + 1) = fgbest; disp(['Iteration ', num2str(t), ': Best fitness = ', num2str(bestfititer(t + 1))]); end

bestFitness = fgbest; bestSolution = gbest;

disp("Best Fitness: " + num2str(bestFitness)); disp("Best Solution: " + num2str(bestSolution));

function cost = fosmc(k) assignin('base', 'k', k); a = sim('YBF', 'SimulationMode', 'normal'); b = a.ITAE; cost = b(end); end

A.2 Rectangular Trajectory Generation

```
function [x_ref, y_ref, theta_ref, v_ref, w_ref] = square_trajectory(t)
    % Time period of the square wave (in seconds)
    period = 30; phase = mod(t, period) / period;
    % Define the length of one side of the square
    square_side_length = 1;
    % Calculate the reference position and velocities for x, y, theta, v, and w
    if phase < 0.25
        x_ref = square_side_length * phase / 0.25;
        y_ref = 0;
        theta_ref = 0;
        v_ref = 1;   % Default linear velocity
        w_ref = 0;   % Default angular velocity
    elseif phase >= 0.25 && phase < 0.5
        x_ref = square_side_length;
        y_ref = square_side_length * (phase - 0.25) / 0.25;
        theta_ref = pi / 2;
        v_ref = 1;   % Default linear velocity
        w_ref = 0;   % Default angular velocity
    elseif phase >= 0.5 && phase < 0.75
        x_ref = square_side_length * (1 - (phase - 0.5) / 0.25);
        y_ref = square_side_length;
        theta_ref = pi;
        v_ref = 1;   % Default linear velocity
        w_ref = 0;   % Default angular velocity
    else
        x_ref = 0;
        y_ref = square_side_length * (1 - (phase - 0.75) / 0.25);
        theta_ref = 3 * pi / 2;
        v_ref = 1;   % Default linear velocity
        w_ref = 0;    end end
```

Appendix B

MATLAB® Script

Figure B.1: General System Design in MATLAB/SIMULINK Software

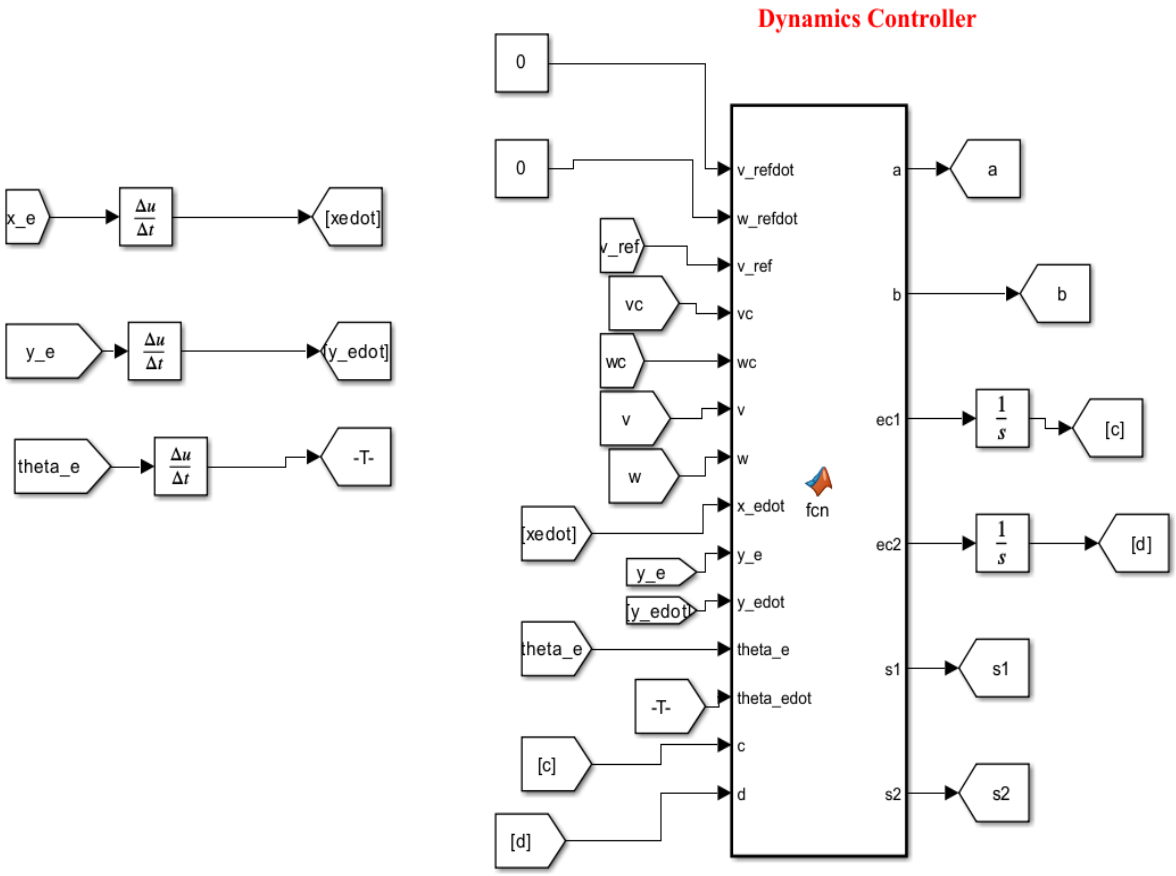


Figure B.2: Dynamics Controller System Design in MATLAB/SIMULINK Software

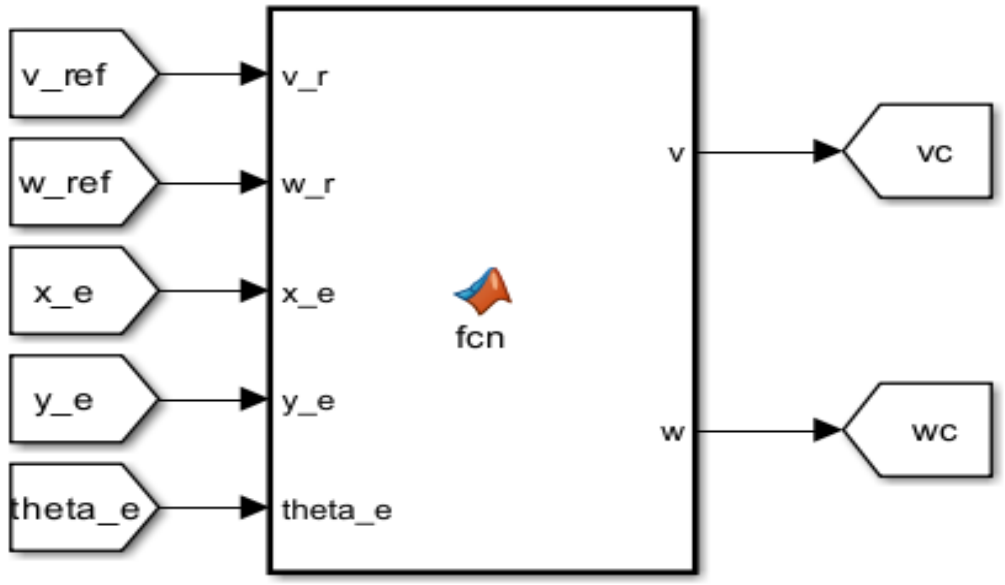


Figure B.3: Backstepping Controller and Kinematics Controller Design in MATLAB/SIMULINK Software

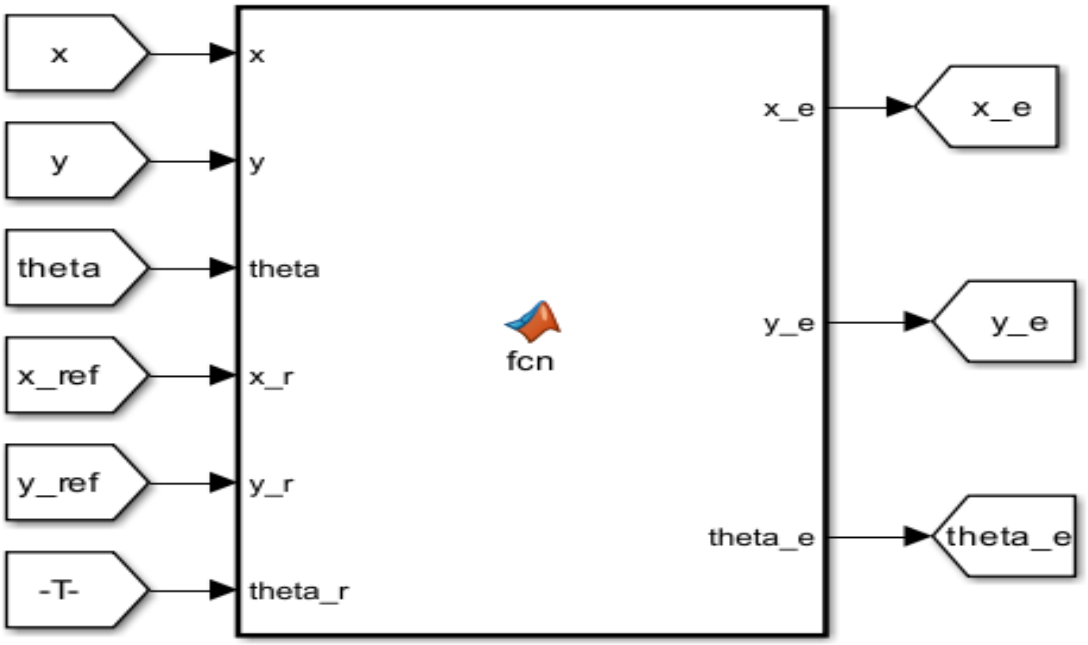


Figure B.4: Rotational Matrix Design of the System in MATLAB/SIMULINK Software

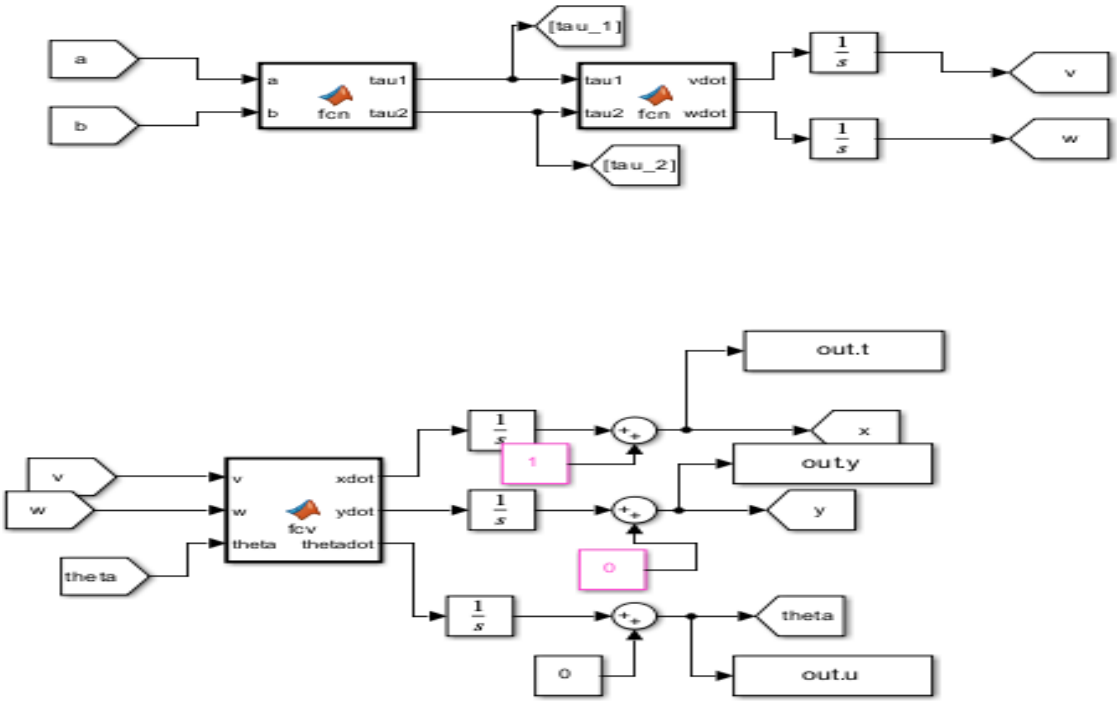


Figure B.5: Kinematics and Dynamics Model of TWMR Design in MATLAB/SIMULINK Software

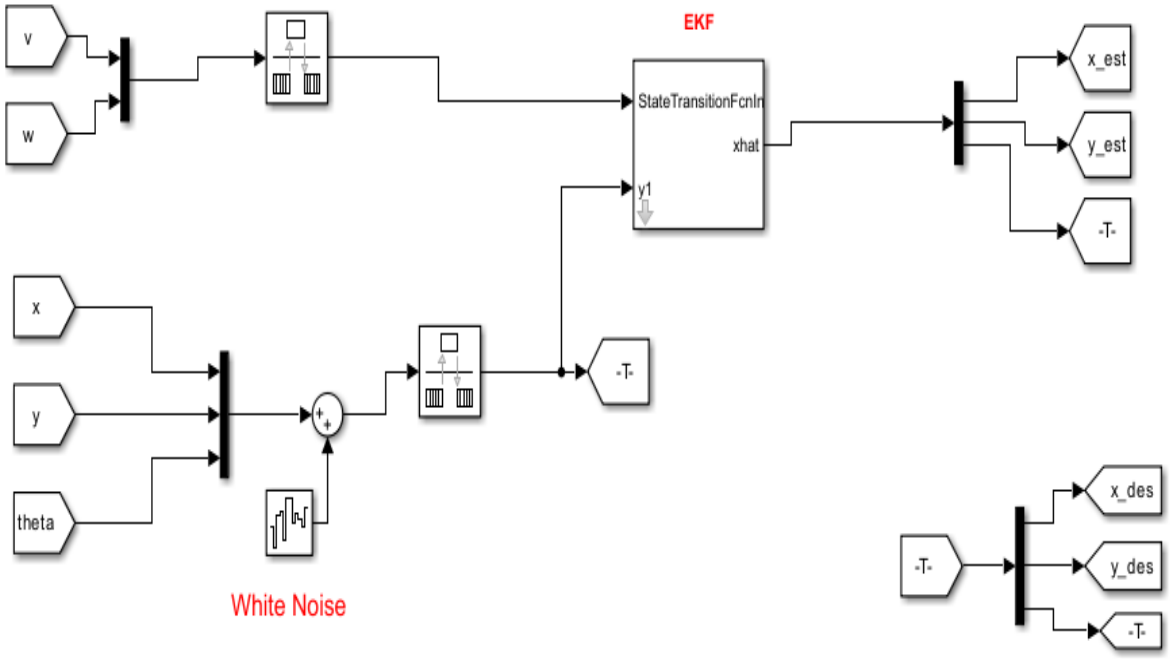


Figure B.6: Extended Kalman Filter Design in MATLAB/SIMULINK Software

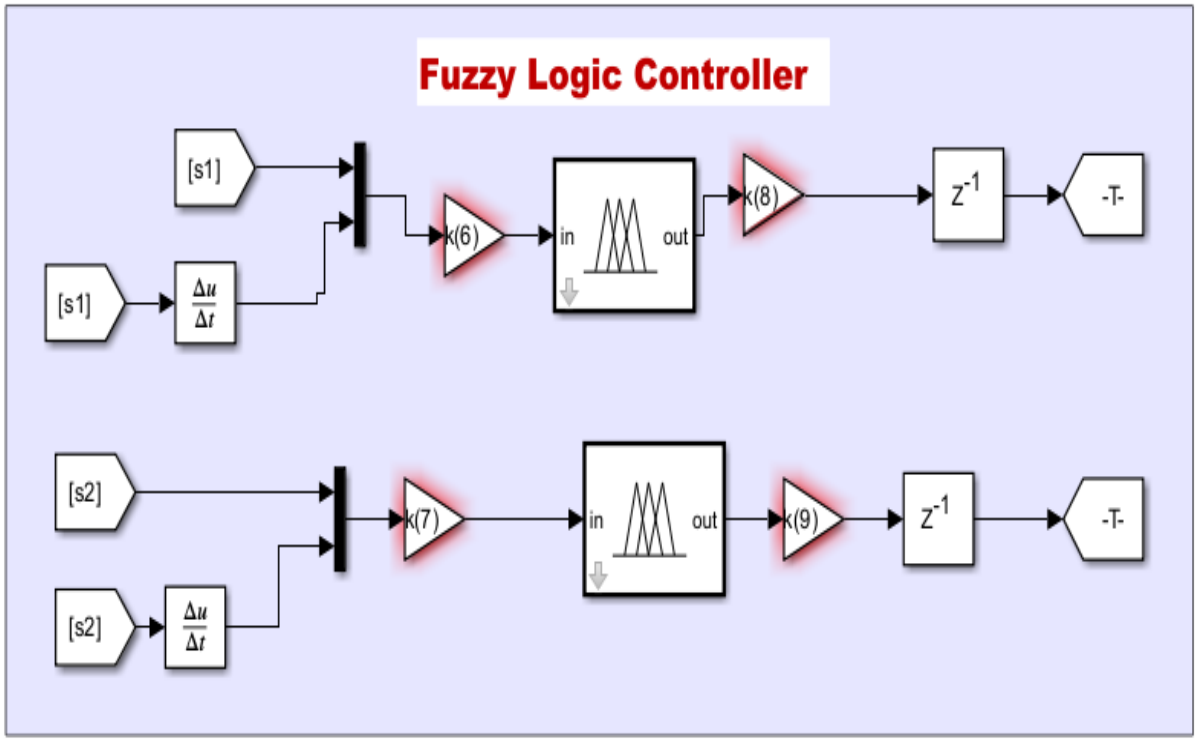


Figure B.7: Fuzzy Logic Controller Design in MATLAB/SIMULINK Software

Appendix C

Appendix® Formula

Integral Error Integral Time Weighted Absolute Error $ITAE = \int_0^t t \mid e \mid edt$

Integral Absolute Value of Error. $IAE = \int_0^t \mid e \mid dt$

Integral Square Error. $ISE = \int_0^t e^2 dt$

Reference Velocity(V_r) $V_r = \sqrt{\dot{x}^2 + \dot{y}^2}$

Reference Angular Speed(w_r) $w_r = \frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{\dot{x}^2 + \dot{y}^2}$

Orientation Reference(θ_r) $\theta_r = \int w_r$

$Error = \sqrt{((x - x_r)^2 + (y - y_r)^2)}$

Tracking Improvement Percentage(TIP)=$\frac{MSTE_{BSMC} - MSTE_{BFSMC}}{MSTE_{BSMC}}$

Table C.1: Parameters used for Stair Trajectories Simulation

Time(Sec)	0	10	20	30	40
Amplitude(m)	1	2	5	7	1