
Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

# Backstepping Fuzzy Sliding Mode Controller for Trajectory Tracking of Mobile Manipulator

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

**Geta Meneyechel**

Advisor

**Dereje Shiferaw (Ph.D)**

April , 2024

Addis Ababa, Ethiopia


Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

# Backstepping Fuzzy Sliding Mode Controller for Trajectory Tracking of Mobile Manipulator

By: Geta Meneyechel

APPROVED BY BOARD OF EXAMINERS

| **Name** | **Date** | **Signature** |
|---|---|---|
| _____ | _____ | _____ |
| (Dean, School of Graduate Committee) | | |
| _____ | _____ | _____ |
| (Advisor) | | |
| _____ | _____ | _____ |
| (Internal Examiner) | | |
| _____ | _____ | _____ |
| (External Examiner) | | |

# Declaration

I, the undersigned, declare that this MSc thesis entitled "Backstepping Fuzzy Sliding Mode Controller for Trajectory Tracking of Mobile Manipulator" is the result of my own independent research work and has not been presented for fulfilment of any degree in this or any other University and all sources and materials used for the thesis are properly acknowledged.

| | | |
|---|---|---|
| Name of the student | Date | Signature |

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

| | | |
|---|---|---|
| Name of the Advisor | Date | Signature |

# Acknowledgement

First and for most praise and thanks to God who guided me to the right path, gave me knowledge and patience to accomplish this work and made me successful throughout this journey.

I extend my deepest appreciation to my advisor, Dr.Dereje Shiferaw for their unwavering guidance, insightful feedback, and constant encouragement in accomplishing this work. Their expertise and commitment to academic excellence have played a key role in guiding the course of this research.

I owe deep gratitude to my family for their constant support and understanding during the ups and downs of this academic endeavor. Their encouragement has been a driving force, and I am truly fortunate to have them by my side. Last but not least, I want to express my gratitude to all my friends who have provided emotional support and shared in the excitement and challenges of this academic pursuit.

This thesis would not have been possible without the collective support of all these individuals, and for that, I am sincerely thankful.

*Geta Meneyechel*

# Abstract

A Mobile Manipulator (MM) is essentially a robotic arm attached to a mobile platform, which could be designed for space, ground, aerial, or underwater environments. The mobile platform expands the reach of the manipulator, allowing it to access a larger workspace. This increased mobility enhances the ability to position the manipulator in various configurations, leading to more efficient task execution.

Mobile Manipulators has complex system structure, highly coupling dynamics between mobile base and mounted manipulator arm, holonomic and nonholonomic kinematics constraints and highly nonlinear characters substantially increase the difficulty in designing a controller for the wheeled mobile manipulator.

Designing a robust controller for mobile manipulator with the aim of simultaneous control of the velocity of the mobile platform and the motion of the end-effector is the aim of this thesis work. By employing the concepts of kinematic backstepping control and fuzzy sliding mode torque control, a two-step control approach is introduced for the nonholonomic mobile manipulator. In the first step, the kinematic velocity control is designed to ensure that all desired trajectories are achieved. In the second step, a fuzzy sliding mode torque controller, based on the dynamics of the mobile manipulator, is designed to ensure that the mobile platform's velocity and the end-effector's position converge to the reference trajectories generated in the first step.

The proposed method stability is proved using Lyapunov theory, and its convergence is mathematically guaranteed. Comparision between BSMC and the proposed BFSMC is conducted in terms of tracking performance in the face of both disturbance and parameter variation and the proposed BFSMC has shown better performance in tracking the given trajectory by rejecting the external disturbances and tolerating the parametric uncertainties results in performance improvement of 31.6%. The effectiveness of the suggested control approach is confirmed through the creation of simulation outcomes using MATLAB/SIMULINK software.

**Keywords:** *Mobile Manipulator, Sliding Mode Control,Fuzzy Sliding Mode Control,Velocity Controller, Torque Control,Tracking Performance, External disturbance*

# Contents

## List of Figures

# List of Tables

## List of Acronyms

| | |
|---|---|
| **MM** | Mobile Manipulator |
| **MR** | Mobile Robot |
| **BSM** | Backstepping Sliding Mode |
| **BFSM** | Backstepping Fuzzy Sliding Mode |
| **DOF** | Degree of Freedom |
| **PSO** | Particle Swarm Optimization |
| **COG** | Center of Gravity |
| **UGV** | Unmanned Ground vehicle |

# Chapter 1

# Introduction

## 1.1 Background

Traditionally, robotic manipulators are usually bolted on the fixed base. The workspace of such a fixed base manipulator is a limited volume of the operation space that can be reached by the end-effector of the manipulator. The workspace is limited and tasks must be carefully structured so that the manipulator can reach parts to be assembled. Integrating a mobile robot to those robotic manipulators will give the capability of moving enabling them to reach the areas which was impossible to reach for a fixed base robotic arms. A mobile manipulator is a robotic system comprising a robotic manipulator mounted on a mobile robot. In this configuration, the DOF of the mobile robot is combined with those of the manipulator arm robot, thereby extending the manipulator's workspace. Mobile manipulators offer significant potential for performing a wide range of tasks that stationary manipulators cannot accomplish. However, leveraging the advantages of such a system also introduces several challenges.

1. The mobile robot (MR) often faces nonholonomic constraints, which restrict its movement capabilities.
2. The mobile platform usually has lower accuracy and a dynamic response which is slower than a robot arm.
3. Combining mobility with manipulation typically results in kinematic redundancy.
4. Typically, the task has to be divided into small movements carried out with the manipulator and large movements executed by the platform.[1]

Mobile manipulator systems, as illustrated in Figure 1.1, have been proposed for numerous applications, including tasks in environments which involves hazards, explosive handling, waste management, space missions and outdoor exploration. Due to their large workspace, dexterous manipulation, and the ability to coordinate between base locomotion and arm manipulation, mobile manipulators represent a key focus area in robotics research [3]. In recent years, MMs have been recognized for their great potential in addressing numerous tasks. These characteristics have been utilized in industries for assembling, welding, painting, and object transportation. Additionally, manipulation has remained one of the most important tasks in

the history of robotics. Leveraging the above mentioned tasks, the intricate physical structure, the tightly interconnected dynamics between the mobile base and the attached manipulator arm, and the mobility of the mobile base are some of the features that substantially increase the difficulty of system design and control[4].

## 1.2 Mobile Manipulator Classification

The classification of mobile manipulators can be based on various criteria, including mobility type, robot arm configuration, application, and the environment in which the mobile manipulators are deployed.

Mobile manipulators can be deployed in various environments, including Aerial, Underwater, Ground enviroments.

### 1.2.1 Aerial Manipulators

Aerial manipulators are robotic systems that combine aerial mobility with manipulation capabilities. These robots are designed to fly in the air, and they typically feature an attached robotic manipulator for performing various tasks. Aerial manipulators have applications in fields such as search and rescue, inspection, maintenance, agriculture and surveillance.[5]


Figure 1.1: Quad copter with robotic arm

### 1.2.2 Underwater Manipulator

Underwater manipulators are robotic systems designed for manipulation tasks in underwater environments. Underwater manipulators are designed to perform tasks such as pipe inspection, opening and cutting ropes, laying cables and repairs, closing valves, drilling and archaeological works.[6]

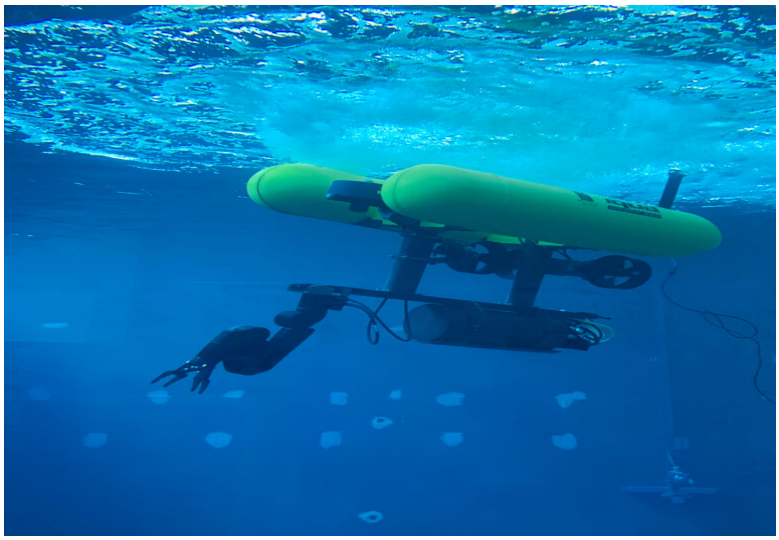


Figure 1.2: Underwater Mobile Manipulator

### 1.2.3 Ground Mobile Manipulators

Ground mobile manipulators are type of mobile manipulator that navigates on the ground, such as, legged mobile manipulators, wheeled mobile manipulators, tracked mobile manipulators.


Figure 1.3: Ground Mobile Manipulator, Robotic Lab Assistant

Mobile manipulators can be used in different areas that can be classified in four domains[7] professional/service(home and health care), space exploration, military and industry. Figure 1.4 shows these four domains.


Figure 1.4: Domains of Mobile Manipulation based on Application [1]

## 1.3 Statement of the Problem

Mobile Manipulators, composed of a mobile robot and a robotic manipulator, offer significant potential for various applications, ranging from manufacturing, agriculture, healthcare and beyond. Mobile Manipulators can provide a solution to highly laborious and extremely dangerous works of manufacturing and mining and missions that require high precision in medical area. However, controlling these complex systems presents several challenges that need to be addressed for optimal performance and efficiency.

Applying Mobile Manipulators for the above mentioned tasks required effective trajectory tracking mechanisms which involves considering the impacts of highly nonlinear and coupled dynamics of the plant and highly variable conditions that the mobile manipulator going to act. Designing a robust controller for mobile manipulator for efficient trajectory tracking with the aim of reducing the effect of chattering on the control input when applying conventional sliding mode making it robust and smooth on the face of unknown disturbance.

By addressing these challenges, this research aims to contribute to the advancement of trajectory tracking control for mobile manipulators, specifically leveraging the benefits of Backstepping Fuzzy Sliding Mode Control. The goal is to enhance the robustness, and precision of mobile manipulators in real-world applications, making them more capable of executing

complex tasks in dynamic and uncertain environments.

## 1.4 Objective of the thesis

### 1.4.1 General Objective

To design and simulate Backstepping Fuzzy Sliding Mode Control for Trajectory Tracking of Mobile Manipulator.

### 1.4.2 Specific Objective

The specific objectives of this thesis are

- To design the kinematics and dynamics that takes in to account the highly coupled nature of Mobile Manipulator.
- Generating smooth trajectory that is going to be feed to the mobile manipulator without resulting in abrupt changes in the plant output.
- Validate the model of Mobile Manipulator.(Model Verification)
- Design a BSMC for Mobile Manipulator.
- Design a BFSMC for mobile manipulator.
- To test and validate the proposed control topology on MATLAB®/SIMULINK® .

## 1.5 Scope of the Thesis

The scope of this thesis is understanding the unified dynamic and kinematic model of the Mobile Manipulator that is composed of two wheel differential drive mobile robot and 3 dof robot manipulator, and designing a controller for trajectory tracking using Backstepping Fuzzy Sliding Mode controller that will enable effective trajectory tracking of MMs in the face of external perturbation and check the stability of the designed controller using lyapunov method.

The designed plant model and controller will be inserted in to the MATLAB®/Simulink® and simulation analysis is conducted.

## 1.6 Contribution of the Thesis

In recent times, in Ethiopia, the trend of using robots or more specifically Mobile Manipulators in many applications is minimal. This is mainly due to the fact that such vehicles are just thought as luxury and of no use in the country. However, applications like agricultural, managing huge ware houses, welding and painting tasks, environments where safety issue is highly on risk e.g places that filled with radioactive materials and mine in case of war, and for tasks that necessitate extreme precision and efficiency are of paramount significance demand smart machines to automate pertaining processes. Therefore, mobile manipulators are needed to steps in to bridge the gap, fulfilling requirements for safety, significant manpower, efficiency, and accuracy. The developed robust Backstepping fuzzy sliding mode controller contributes for efficient trajectory tracking Mobile Manipulators making mobile manipulators stable in disturbed scenarios and avoiding the chattering problem of conventional SM.

## 1.7 Thesis Organization

The overall thesis is organized in to seven distinct chapters including this introductory chapter. Further details of the later chapters are herein presented.

**Chapter 2:** Provides a review of several technical and theoretical research papers that are deemed significant and serve as foundational pillars for the work presented in this paper.

**Chapter 3:** Describes the modelling of the mobile manipulator. Both the dynamic and kinematic equations are subsequently developed, taking into account all significant physical effects on the mobile manipulator.

**Chapter 4:** Presents the overall control scheme design where the Fuzzy-SMC is designed to control the dynamics of MM and Backstepping controller is designed as kinematic controller with their respective stability analysis and convergence properties.

**Chapter 5:** Discusses the simulation approach, trajectory tracking performances, the results found and their analysis. And analyzes the results of both the conventional sliding mode (SM) controller and the fuzzy SM controller under unfavorable scenarios simulated by applying disturbances and parametric uncertainties.

**Chapter 6:** Conclusions are drawn based on the comprehensive analysis conducted, and the potential avenues for further research are intricately discussed.

# Chapter 2

# Literature Review

The following papers are reviewed regarding the modeling and trajectory tracking control methods for mobile manipulators.

## 2.1 Conventional Control Schemes

The litrature in [7][8][9]designed LQR controller with the help of matlab assuming full states are available which is captured from mathematical equations of two wheeled manipulator. The derived dynamics of MM is nonlinear. Hence the mobile manipulator system get linearized by approximating the its generalized coordinate values to some linear and constant values.The proposed Linear Quadratic Regulator is developed by recursively solving the Riccati equation to obtain the optimal value of the control law. Although the LQR has proven to be an efficient controller for linear and ideal models, it falls short when applied to nonlinear system models. The proposed LQR controller provides high steady-state accuracy and low energy consumption but is sensitive to coupling effects and exhibits a poorer transient response.

PID controller is proposed in [10] for trajectory tracking control of a mobile manipulator consisting of four degrees of freedom (4-DOF) robotic arm installed on the top of a mobile platform. This study focus on the modeling of mobile manipulator and applying partial feedback linearization method. The core concept is to mathematically convert the dynamics of nonlinear systems into a partial linear form. This approach allows control techniques which are linear to be used effectively to manage trajectory tracking on MMs, enabling tracking of various trajectories such as ellipses and circles while respecting non-holonomic constraints. In the proposed method, if the disturbances are non-vanishing, the controller can only attenuate their effects and maintain the output close to the refrance path. The results from the simulation of this study reveals that the designed PID controller exhibits strong performance characteristics and stability. However, the research focuses solely on regulating the trajectory of the MR, assuming manipulator arm on the mobile robot follows the position of the MR itself.

## 2.2 Modern Control Schemes

The paper in [11] introduces a decentralized SMC approach for a redundant wheeled mobile manipulator. Initially, the mobile MM is partitioned into two interconnected subsystems: the mobile robot subsystem and the robotic arm subsystem. In the subsequent step, dynamic and kinematic controllers are formulated for mobile platform subsystem using backstepping and sliding mode control technique respectively. The kinematic control produces the desired velocity, subsequently employed in crafting the dynamic controller. This process ensures the mobile platform subsystem effectively tracks the desired trajectories. In the third stage, the dynamic control law for the manipulator subsystem is devised by considering both the desired and actual joint angle values of the robotic arm subsystem, along with the desired velocity of the MR subsystem. The developed control law for manipulator was implementing state feedback law using lyapunov technique which suffers from instablization from external disturbances and nonlinear uncertainities. According to the simulation results, the tracking error does not smoothly converge to zero; instead, it exhibits an oscillatory behavior. This oscillatory nature is not desirable behavior in a real-world physical system.

The author in [12] proposed trajectory tracking control of mobile manipulator composed of 2 dof robotic arm and two wheel differential drive mobile robot. Modeling of the two subsystems is done independently and coupling is done between mobile platform and robotic arm dynamics by using dynamic coupling term. While independent modeling offers modularity, it may not capture certain dynamic interactions between the mobile platform and robot arm accurately. This approach is suitable when the interactions between mobile robot and manipulator arm are relatively straightforward or when a modular design is a priority. Actuation space control for manipulators subsystem in specified workspace is used which involves specifying control inputs directly in terms of joint angles or velocities. This representation may not be as intuitive for users or designers, as it requires thinking in terms of joint configurations rather than the mobile manipulator end-effector's required behavior in task space.

The literature in [2] proposed adaptive robust controlling mechanism implementing sliding mode CMAC for a mobile manipulator composed of 3 dof robot arm and two wheel differential drive mobile robot. Dynamic model is constructed in the way that includes the dynamic interactions between the mobile robot and the manipulator arm which is the best representative of the real plant. Coordinate transformation and end effector positioning is also included in kinematic modeling which helps in determining how the movement of the mobile robot affects the

position and orientation of the robotic arm end-effector. It has also shown the effective cooperation of the MR and manipulator arm to accomplish tasks which are beyond the workspace of the robot arm. Although the approach presented in[2] is good representative of the real system it lacks a detailed inverse kinematics modeling and corresponding velocity controller, due to this the control mechanism was unable to compensate the errors in Cartesian space for the improvement the overall accuracy of the robots movement and this inefficiencies are clearly observed on the given simulation results that conducted on complex curved trajectories.

Backstepping-Adaptive control proposed in [13] for mobile manipulator which has 2 dof robot arm and two wheel deferential drive robot. A brief description of dynamic and kinematic modeling for the plant is included. The derivation of inverse kinematics model for the plant using pseudo inverse mechanism also included. And two stage controlling mechanism is designed for the plant *i.e.* inner loop adaptive controller for torque controlling and outer loop backstepping controller for velocity controlling which helps in reducing the errors in both joint angle level and Cartesian position level. But having highly varying nature of the plant and the working environment the proposed adaptive controller seen struggling to adapt quickly enough to the optimal performance thus taking longer convergence time to the desired trajectory.

The paper in [14] employed backstepping control and technique called filtered-error method to design nonlinear tracking controller for MM. The main advantage of the BCS can be mentioned as keeping the robustness nature despite of the fact that facing high uncertainties. Although the principles of this method were derived from Lyapunov theory to ensure the stability of the system which is being controlled, this paper lacks taking into consideration external disturbances and unmodeled dynamics. Techniques which are intelligent for instance the fuzzy logic can be considered as a good candidate for improving the ability and eliminating the drawbacks of the backstepping control design methodology.

On the other hand, several control techniques involving feedback linearization control (FLC), computed torque control (CTC) [15], have been employed to overcome the problem of trajectory tracking control for one-arm mobile manipulators. However, the main flaws of FLC and CTC is that demanding precise system modeling as as necessary precondition in designing the controller as in order to eliminate the nonlinear dynamic effects; this requirement is impractical since it is not possible to precisely estimate the system's physical parameters and to get values which stay away from system uncertainties and external disturbances.

In summary from all the literature's described above it can be conclude that mobile manipulators present a big problem from finding a best model representatives which fully represent the system and its behaviour due to complex nature of Mobile Manipulator plant. This difficulty also imposes its negative impact in the finding and integrating of controllers for the application of trajectory tracking specially for conventional controller type ones and make it compulsory to use robust and intelligent controllers to get the required amount of accurate results.

# Chapter 3

# Modeling of Mobile Manipulator

## 3.1 Introduction

The control and design of MM robot necessitate a thorough understanding of the entire system and an appropriate mathematical representation. Here in this section, kinematic and dynamic modeling of a mobile manipulator robot are presented, specifically a two-wheel differential drive with an additional caster wheel and a 3-degree-of-freedom robotic arm. Kinematic modeling of the MM robot, excluding the effects of forces and torques, is presented in the following pages, with dynamic aspects addressed in the next section. First, the mathematical model mobile robot and its nonholonomic constraints are described. On the sections after that, the kinematics of manipulator is added, and all necessary kinematic relationships between the wheeled mobile robot and manipulator arm are formulated. Following this the dynamic model of MM is formulated by calculating the kinetic and potential energy of the components of mobile manipulator to be used as an input for Lagrangian method derivation of dynamic model.

## 3.2 Mobile Robot Kinematic Modeling

Wheeled mobile robots are a type of non-holonomic system prized for their capability to autonomously navigate and perform designated tasks. A distinctive feature of these systems is their inability to move sideways, attributed to the perfect rolling without slipping constraints of their wheels.[16]

**Coordinate Systems:** To describe the location of the MR in its environment, two distinct coordinate systems needs to be defined.

**1. Inertial Coordinate System:** defined as global frame anchored in the environment or plane where the mobile robot operates. Additionally, this frame serves as frame of reference. and is denoted as $X_I, Y_I$.

**2. Robot Coordinate System:** This coordinate system is a a frame on the body of mobile

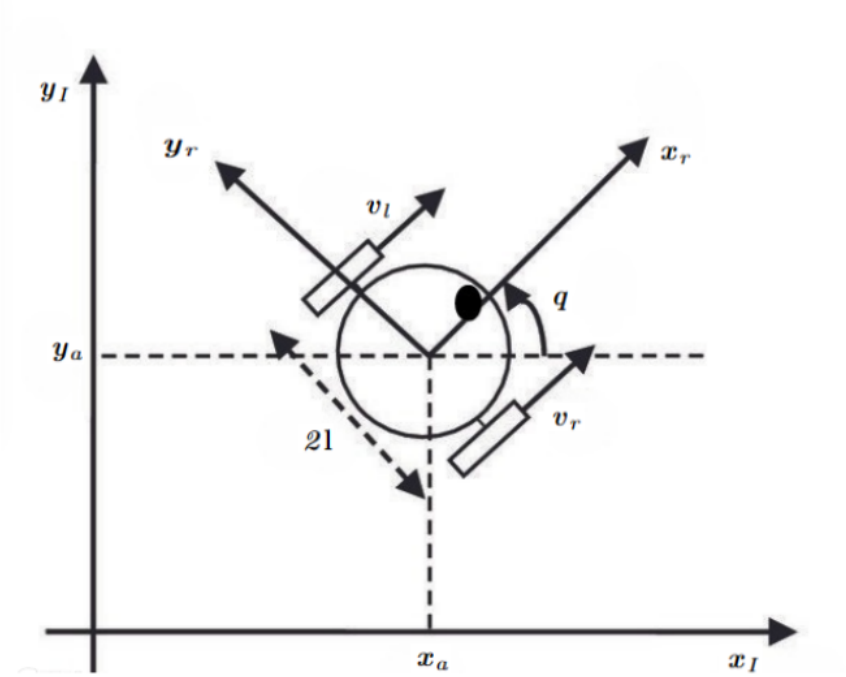



Figure 3.1: Mobile Robots Main Coordinate Frames.

robot, moving with it as it navigates its environment. It is represented as $X_R, Y_R$. The above mentioned two frames are depicted in Figure 3.1. The robot frame's origin is set at the midpoint of the axis between the wheels.

The position of any point on the robot can be defined in the robot frame and the inertial frame as follows.

$$X^R = \begin{bmatrix} x_r \\ y_r \\ q_r \end{bmatrix}, \text{ and } \quad X^I = \begin{bmatrix} x_I \\ y_I \\ q_I \end{bmatrix}$$

$X^R$ and $X^I$ be the coordinates of the given point in the robot frame and inertial frame, respectively.

Then, the relationship between these two coordinates can be expressed by the following transformation:

$$X^I = R(q)X^R \tag{3.1}$$

Where $R(q)$ is the orthogonal rotation matrix

$$R(q) = \begin{bmatrix} C_q & -S_q & 0 \\ S_q & C_q & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

where $C_q$ and $S_q$ are $Cosq$ and $Sinq$ The mobile platform depicted in Fig. 2.1 is a well known differential-driven mobile platform. It is made up of a vehicle with two drive wheels aligned on the same axis and an additional one passive front wheel. The movement and orientation are controlled by separate actuators, for instance two dc motors supplying the required torques to the rear wheels. Mechanical system is subjected to two constraints holonomic and Nonholonomic constraints.[12]

**Holonomic constraints:** Constraints on the position (configuration) of a particle system are referred to as holonomic constraints. These constraints can be integrated into constants of integration.

**Non holonomic constraints:** Constraints that limit the velocities of particles without affecting their positions and is subjected to at least one nonintegrable (i.e. non-holonomic) constraint.

The nonholonomic constraint indicates that the robot can only move perpendicular to the axis of the driving wheels, ensuring that the mobile base adheres to pure rolling and nonslipping conditions. To represent this aspect, initially define a holonomic constraint as any restriction that can be expressed mathematically as:

$$f(q,t) = 0 \tag{3.3}$$

where $q = [q_1, q_2, q_3, ...q_n]$ is the array of the robot's position in robot frame of reference. To mathematically verify the non-holonomic constraint, considering that the non-holonomic constraint of the mobile robot takes the following form:

$$f(q,\dot{q},t) = 0 \tag{3.4}$$

If Eq. (3.4) can't integrated back to give Eq. (3.3) this implies that the mobile robot has non-holonomic constraints, which impose certain limitations on its mobility.

In terms of control, if a mobile robot has fewer actuators than its total degrees of freedom , it will exhibit non-holonomic constraints. A two-wheel differential drive mobile robot has only two actuators, while the system has a total of three DOF. Consequently, this MR system has two non-holonomic constraints and one holonomic constraint. Therefore, the mobile robot's motion is characterized as a non-holonomic mechanical system with three kinematic constraints.

The motion of a MP is defined by two nonholonomic constraint equations:

**No lateral slip motion:** Mobile robot can only move only in a forward and backward direc-

tion, following the plane of rotation of its wheels, and cannot move laterally or sideways.

$$\dot{y_r} = 0 \tag{3.5}$$

Employing the orthogonal rotation matrix $R(\theta)$, the velocity the mobile robot the inertial frame is givesn as

$$cos(y_r) - sin(x_r) = 0 \tag{3.6}$$

Eq. (3.6) is a nonholonomic constraint which can't be integrated analytically to establish a constraint between the MR's configuration variables, namely $x_r, y_r$ and $q$ .

**Pure rolling constraint:** The wheel does not slip along its longitudinal axis($x_r$).
The Contact point velocities in the robot frame are connected to the wheel velocities by::

$$v_r = r * \dot{q_r} \tag{3.7}$$

$$v_l = r * \dot{q_l} \tag{3.8}$$

This system has 3 DOF, but the robot can only move instantly in two directions (forward and backward, left and right rotational ) beacuse of the velocity-level constraint. This means there is a non-holonomic constraint (i.e., lateral motion is not possible). This constraint prevents the mobile robot to have any lateral motion thus making it impossible to have a velocity parallel to the direction of the tracks' axles.[2]

The following assumptions are applied during the mathematical modeling:

- The maximum velocity that can be achieved by MR is less than 2 m/s, which allows to neglect high-speed navigation and thereby prevent high inertia.
- Longitudinal sliding of the MR is disregarded, as it only occurs if the MR slides and slips, which is not accounted for in this study.
- The wheel's contact with the horizontal plane (ground) is modeled as a single point, with no consideration given to distributed track-surface contact.[2]

- Each wheel's rotational plane remains perpendicular to the ground, allowing the robot to pitch and roll without impacting its kinematics.

  Consider a differentially-driven nonholonomic MP moving in a plane where a body stable frame $X^r$, is attached to it at the center of the axle. Let the forward direction of travel

Table 3.1: Kinematic wheeled mobile robot parameters.

| Parameter | Description |
|---|---|
| $r$ | Wheel radius |
| $2l$ | The distance separating the left and right wheels |
| $l_b$ | Distance between wheel axis and MR center of mass |
| $v_l, v_r$ | Right and Left wheel linear velocity |
| $\dot{q}_l, \dot{q}_r$ | Right and Left wheel angular velocity |
| $q$ | MR heading angle |
| $x_a, y_a$ | MR center of mass location |

be as shown in Figure 3.1, along the robot's $x_r$ axis. Let be the $q$ orientation of frame $X^r$ w.r.t the world coordinate frame $X^I$. The following table (Table 3.1) provides the robot parameters for the mobile platform used in the kinematic modeling:

As widely recognized, the system's configuration space is three-dimensional (fully unrestricted), whereas the velocity space is two-dimensional. This constraint arises

$$\dot{x}_a C_q + \dot{y}_a S_q + l\dot{q} = r\dot{q}_l \tag{3.9}$$

$$\dot{x}_a C_q + \dot{y}_a S_q - l\dot{q} = r\dot{q}_l \tag{3.10}$$

where $q_r$ and $q_l$ are the diving wheel angular positions respectively, and 2l is the MR width

Let $q = (x_a, y_a, q, q_r, q_l)$ represent the generalized coordinates of mobile robot. The above discussed 3 constraints results matrix A(q)

where:$A(q) \in R^{m \times n}$ is a full rank matrix.

$$A(q) = \begin{bmatrix} -S_q & C_q & 0 & 0 & 0 \\ C_q & S_q & l & -r & 0 \\ C_q & S_q & l & -r & 0 \end{bmatrix} \tag{3.11}$$

Let $m$ rank matrix $S(q) \in R^{n \times m}$ created using a set of smooth and linearly independent vector fields covering the null space of $A(q)$ , i.e.,

$$S^T(q)A^T(q) = 0 \tag{3.12}$$

Using (3.7) and (3.9), it is can be obtained an auxiliary vector time function $v(t) \in R^{n-m}$ such

that, for all $t$.

$$\dot{q} = S(q)v(t) \tag{3.13}$$

where: $v(t) = [\dot{q_r}, \dot{q_l}]$

For the two wheels mobile platform as depicted in Fig. 3.1, $S(q) \in R^{5\times 2}$ can be formulated as

$$S(q) = [s_1(q)s_2(q)] = \begin{bmatrix} \frac{r}{2}\cos q & \frac{r}{2}\cos q \\ \frac{r}{2}\sin q & \frac{r}{2}\sin q \\ \frac{r}{2l} & -\frac{r}{2l} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.14}$$

$S(q)$ matrix is in the null space of matrix $A(q)$, i.e $S^T(q)A^T(q) = 0$. To get the holonomic constraint, subtract Equation (3.9) from Equation (3.10)

$$2l\dot{q} = r(\dot{q_r} - \dot{q_l}) \tag{3.15}$$

which is a holonomic constraint. Thus $q$ can get eliminated from generalized coordinates. Nonholonomic constraints are

$$\dot{x}_a \sin q - \dot{y}_a \cos q = 0 \tag{3.16}$$

$$\dot{x}_a \cos q + \dot{y}_a \sin q = \frac{r}{2}(\dot{q_r} + \dot{q_l}) \tag{3.17}$$

The other nonholonomic constraint equation which is mentioned above is derived by adding Eq. (3.9) from Eq. (3.10). It can be observed that $q$ is now a shorthand notation for $c(q_r - q_l)$ rather than an independent variable. It is possible to formulate these two constraint equations the following matrix form

$$A(q)\dot{q} = 0 \tag{3.18}$$

where$q$ (the generalized coordinate vector) $q$ is defined as

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} x_a \\ y_a \\ q_r \\ q_l \end{bmatrix}$$

and $A(q)$ is given by

$$A = \begin{bmatrix} -\sin q & \cos q & 0 & 0 \\ -\cos q & -\sin q & \frac{r}{2} & \frac{r}{2} \end{bmatrix}$$

Since the mobile platform lacks a steering wheel, its rotational motion is controlled differentially. This means that every movement sequence of the mobile platform must be carefully managed to adhere to the nonholonomic slip constraint, ensuring no violations occur. MP linear velocity of wheels (both right and left wheels respectively ) are obtained using Eq. (3.19) while the wheels (both MR total linear velocity is found using Eq. (3.21)

$$v_l = r * \dot{q}_l \tag{3.19}$$

$$v_r = r * \dot{q}_r \tag{3.20}$$

$$v = \frac{v_r + v_l}{2} = \frac{r(\dot{q}_r + \dot{q}_l)}{2} \tag{3.21}$$

and the angular velocity of the MR is

$$\omega = \frac{v_r - v_l}{2l} = r\frac{(\dot{q}_r - \dot{q}_l)}{2} \tag{3.22}$$

In the kinematics the MR's $v$ (linear velocity) and $\omega$ (angular velocity) can be represented in a matrix form as

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2l} & -\frac{r}{2l} \end{bmatrix} \begin{bmatrix} \dot{q}_r \\ \dot{q}_l \end{bmatrix} \tag{3.23}$$

The MRs velocities in the robot frame can now be represented in terms of the center-point $A$ velocities as follows:

$$\begin{cases} \dot{x}_a = r\frac{(\dot{q}_r+\dot{q}_l)}{2} \\ \dot{y}_a = 0 \\ \dot{\theta} = \omega = r\frac{(\dot{q}_r+\dot{q}_l)}{2l} \end{cases} \tag{3.24}$$

Thus

$$\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{2l} & -\frac{r}{2l} \end{bmatrix} \begin{bmatrix} \dot{q}_r \\ \dot{q}_l \end{bmatrix} \tag{3.25}$$

The MR velocities on inertial frame can also be obtained as follows:

$$\dot{q}^I = \begin{bmatrix} \dot{x_I} \\ \dot{y_I} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos q & \frac{r}{2}\cos q \\ \frac{r}{2}\sin q & \frac{r}{2}\sin q \\ \frac{r}{2l} & -\frac{r}{2l} \end{bmatrix} \begin{bmatrix} \dot{q}_r \\ \dot{q}_l \end{bmatrix} \tag{3.26}$$

$$\dot{q}^I = \begin{bmatrix} \dot{x_I} \\ \dot{y_I} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \cos q & 0 \\ \sin q & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.27}$$

If the center of mass of the mobile platform does not overlap on the center of its axle and distance between center of the mass of the robot to its center of wheel axle is defined to be $l_b$, the above kinematic equation will have to modified. The modified holonomic and nonholonomic constriant equation will have the following forms

$$\dot{y_a}cosq - \dot{x_a}sinq - l_b\dot{q} = 0 \tag{3.28}$$

$$\dot{x_a}cosq + \dot{y_a}sinq + l\dot{q} = r\dot{q_r} \tag{3.29}$$

$$\dot{x_a}cosq + \dot{y_a}sinq - l\dot{q} = r\dot{q_l} \tag{3.30}$$

The above mentioned equation Eq(3.18) will also modified as

$$A(q) = \begin{bmatrix} -\sin q & \cos q & -l_b & 0 & 0 \\ \cos q & \sin q & l & -r & 0 \\ \cos q & \sin q & l & -r & 0 \end{bmatrix} \tag{3.31}$$

The kinematic model of the platform is modified as,

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{q} \\ \dot{q}_r \\ \dot{q}_l \end{bmatrix} = \begin{bmatrix} \frac{r}{2l}(l\cos q - l_b \sin q) & \frac{r}{2l}(l\cos q + l_b \sin q) \\ \frac{r}{2l}(l\sin q + l_b \cos q) & \frac{r}{2l}(l\sin q - l_b \cos q) \\ \frac{r}{2l} & -\frac{r}{2l} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_r \\ \dot{q}_l \end{bmatrix}$$

$$\dot{q}^I = \begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \cos q & -l_b \sin q \\ \sin q & l_b \cos q \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.32}$$

In this kinematic equation wheel angular velocities of the mobile platform are independently controlled. Given the mobile platforms heading velocity $w$ and the mobile platform linear velocity $v$ then it's possible to calculate $\dot{q}_r$ and $\dot{q}_l$ (right and left wheel angular velocities ) as

$$\dot{q}_r = \frac{v + lw}{r} \tag{3.33}$$

$$\dot{q}_l = \frac{v - lw}{r} \tag{3.34}$$

The previous set of equations (Eqs. 3.7 ,3.9, 3.17 and 3.18) represents the mathematical kinematic model of MR and are used in the control development.

## 3.3 Kinematics of Robotic Manipulators

In robotic manipulators, each pair of joint and link represents one DOF. For an $n$ DOF robot, there are $n$ joints while link 0 attached to a supporting mobile base. The numbering of the joints and the link are done progressively from the base outwards toward the arm; and the arms $i$-th joint is the point where the $i$-th link and the $(i-1)$ th link gets connected. To accurately describe the relative relation between two adjacent links, the Denavit-Hartenberg (D-H) convention is commonly used. This convention involves selecting frames of reference in robotic systems, where each homogeneous transformation matrix represents the coordinate system of each link at the joint relative to the coordinate system of the previous link. Figure 3.2 will be used for understanding the definition of Denavit-Hartenberg frame.

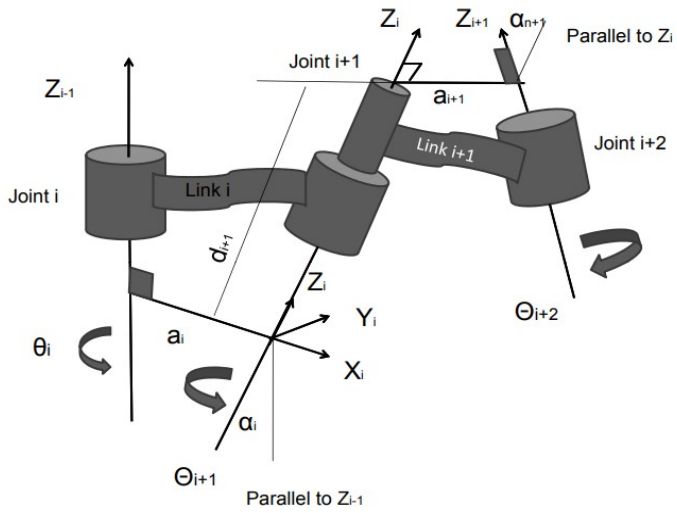



Figure 3.2: D-H frame assignment

The transformation matrix between two adjacent manipulator links is typically derived in a generalized form, as illustrated in Eq. (3.35) using the parameters in Table 3.2:

$$T_i^{i+1} = \begin{bmatrix} Cq_i & -Sq_iC\alpha_i & Sq_iS\alpha_i & a_iCq_i \\ Sq_i & Cq_iC\alpha_i & -Cq_iS\alpha_i & a_iSq_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.35}$$

Table 3.2: ANA-Bot manipulator D-H parameters

| $i$ | $\alpha$ | $a$ | $d$ | $q$ |
|---|---|---|---|---|
| 1 | $\pi$ | 0 | $a_1$ | $q_1$ |
| 2 | 0 | $a_2$ | 0 | $q_2$ |
| 3 | 0 | $a_3$ | 0 | $q_3$ |

In case $i$ th joint is prismatic, the homogeneous transformation matrix is determined as follows:

$$T_i^{i-1}(q_i) = \begin{bmatrix} \cos q_i & -\cos\alpha_i \sin q_i & \sin\alpha_i \sin q_i & 0 \\ \sin q_i & \cos\alpha_i \cos q_i & -\sin\alpha_i \cos q_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.36}$$

where $q_i$, $\alpha_i$ and $a_i$ are constants and $d_i$ is the variable distance.

- $a_i$ is the length of the common normal, which is equal to shortest distance between the $z_{i-1}$ axis and the $z_i$ axis.
- $d_i$ is the offset, the distance from the origin of the $i-1$ coordinate frame to the intersection point of the $z_{i-1}$ axis.
- $\alpha_i$ is the twist, the angle between the $z_{i-1}$ axis and the $z_i$ axis about the $x_i$ axis in the right-hand sense.
- $q_i$ is the angle between the $x_{i-1}$ axis and the $x_i$ about the $z_{i-1}$ axis in the right-hand sense.

Therefore the last link transformation matrix with respect to the base of the manipulator is achieved as:

$$T_0^3 = T_0^1 * T_1^2 * T_2^3 \tag{3.37}$$

$$T_0^1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.38}$$

$$T_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.39}$$

$$T_2^3 = \begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.40}$$

Therefore, the forward kinematics equation of the manipulator is derived as follows:

$$T_0^3 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.41}$$

Thus, the forward kinematics for the robot arm is:

$$T_0^3 = T_0^1T_1^2T_2^3 = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & S_1 & C_1\left(a_2C_2 + a_3C_{23}\right) \\ S_1C_{23} & -S_1S_{23} & -C_1 & S_1\left(a_2C_2 + a_3C_{23}\right) \\ S_{23} & C_{23} & 0 & a_1 + a_2S_2 + a_3S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.42}$$

where: $C_i = \cos(q_i)$ $S_i = \sin(q_i)$ $C_{23} = \cos(q_2 + q_3)$ $S_{23} = \sin(q_2 + q_3)$ for $i = 1, 2, 3$

The location of the end effector of the robot arm w.r.t the base frame of the manipulator arm base is found in Equation (3.28):

$$X_b^e = C_1 \left(a_2 C_2 + a_3 C_{23}\right), \tag{3.43}$$

$$Y_b^e = S_1 \left(a_2 C_2 + a_3 C_{23}\right) \tag{3.44}$$

$$Z_b^e = a_1 + a_2 S2 + a_3 S_{23} \tag{3.45}$$

In the manipulator's model, the Jacobian matrix is required consistently during both the modeling and subsequent control development stages. In this context, the Jacobian matrix will be utilized to define the dynamic relationship within the mobile manipulator. Each entry in the Jacobian matrix represents the derivative of a corresponding kinematic equation w.r.r one of the variables. This implies that the first kinematic equation should describe movements along the $x$-axis, which, in this specific case, corresponds to $X$. $X$ represents the $x$ axis movement of the hand frame of reference. In order to achieve this the elements of the end effector of the MM, $X$, $Y$ and $Z$ , needs to be differentiated to get the Jacobian components. The final column of robot arm's forward kinematic equation in Eq. (3.42) representing the end effector position are differentiated w.r.t. $q_1$ - $q_3$. Due to this the following equations are obtained:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = J * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{3.46}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} -s_1 * (a_2 c_2 + a_3 c_{23}) & -c_1 * (a_2 s_2 - a_3 s_{23}) & -a_3 c_1 s_{23} \\ c_1 * (a_2 c_2 + a_3 c_{23}) & -s_1 * (a_2 s_2 - a_3 s_{23}) & -a_3 s_1 s_{23} \\ 0 & a_2 c_2 + a_3 c_{23} & a_3 c_{23} \end{bmatrix} * \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{3.47}$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} -s_1 * (a_2 c_2 + a_3 c_{23}) & -c_1 * (a_2 s_2 - a_3 s_{23}) & -a_3 c_1 s_{23} \\ c_1 * (a_2 c_2 + a_3 c_{23}) & -s_1 * (a_2 s_2 - a_3 s_{23}) & -a_3 s_1 s_{23} \\ 0 & a_2 c_2 + a_3 c_{23} & a_3 c_{23} \end{bmatrix}^{-1} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \tag{3.48}$$

By utilizing the Jacobian, one can calculate the total force needed at the end effector to ac-

complish the task. The total force exerted by the joint actuator must be computed accordingly in Eqn. (3.49):[2]

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \left[J^T\right]^{-1} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{3.49}$$

## 3.4 Kinematic Modeling of the MM (MR + Manipulator Arm)

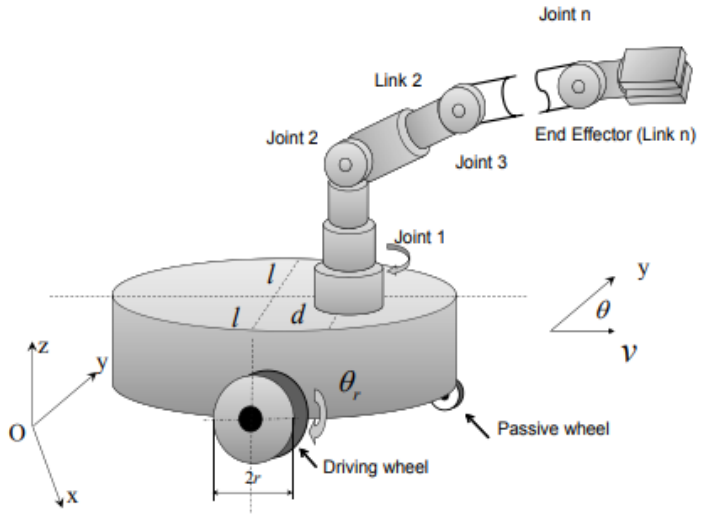



Figure 3.3: Differential-driven mobile manipulator

Using the kinematic equation of the MR (Section 3.2), the manipulator arm (Sections 3.3) and it is possible to proceed to combine them to get a complete MM kinematic equation. Combining the models of the mobile robot (MR) and the manipulator arm will results in a kinematic and dynamic equation of MM which is complex, highly coupled, and exhibits a significant degree of dynamic and kinematic interactions.[17] Although modern mobile manipulators come equipped with various types of wheels and arrangements (such as car-like platform, tracked, tricycle, differential, and Mecanum wheels) to enhance maneuverability, there has been limited effort in using comprehensive models for control purposes. Here, it is tried to employ a relatively complete model of the mobile manipulator for trajectory tracking control. Taking into account the mobile manipulator depicted. In Fig. 3.3, the main four coordinate frames

are depicted: platform frame $O_p$ , world frame $O_w$ , end effector frame $O_e$ and manipulator base frame $O_b$ . After having this, the manipulator's end effector position and orientation with respect to world frame reference is found as:

$$T_e^w = T_p^w T_b^p T_e^b \tag{3.50}$$

where $T_p^w$ is obtained form position of the MR, $T_b^p$ is a constant matrix decided by the base point of manipulator arm and $T_e^b$ is found using joint variables $q = [q_1, q_2, ..., q_m]^T$ with $m$ is DOF of manipulator arm.

$$T_p^w = \begin{bmatrix} \cos q & -\sin q & 0 & 0 \\ \sin q & \cos q & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.51}$$

$$T_b^p = \begin{bmatrix} 1 & 0 & 0 & l_b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.52}$$

$$T_e^b = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & S_1 & C_1\left(a_2C_2 + a_3C_{23}\right) \\ S_1C_{23} & -s_1S_{23} & -C_1 & S_1\left(a_2C_2 + a_3C_{23}\right) \\ S_{23} & C_{23} & 0 & a_1 + a_2S_2 + a_3S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.53}$$

$$T_e^w = T_p^w T_b^p T_e^b$$

$$T_e^w = \begin{bmatrix} C_qC_{q_1}C_{q_2q_3} - S_qS_1S_{q_2q_3} & -C_qC_{q_1}C_{q_2q_3} + S_qS_1S_{q_2q_3} & S_{qq1} & l_bC_q + a_2C_{q_2}C_{qq1} + a_3C_{q_2q_3}C_{qq_1} \\ S_qC_{q_1}C_{q_2q_3} - C_qS_{q_1}S_{q_2q_3} & S_qC_{q1}C_{q_2+q_3} - C_qS_{q_1}S_{q_2q_3} & C_{q+q_1} & l_bS_q + a_2C_{q_2}S_{qq1} + a_3C_{q_2q_3}S_{qq_1} \\ S_{q_2q_3} & C_{q_2q_3} & 0 & a_1 + a_2S_{q_2} + a_3S_{q_2q_3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.54}$$

Now the kinematic equation of MR is successfully combined with the manipulator arm kinematic equation Eq,(3.42)

The result is Eqn. (3.54) which represents mobile manipulatior's end effector position in 3D space $(X_e^w, Y_e^w, Z_e^w)$ as it is moved on top of the MR during its motion on the given trajectory. By taking the last column of $T_e^w$ the value of $(X_e^w, Y_e^w, Z_e^w)$ found as:

$$\begin{aligned} X_e^w &= x_0 + l_bC_q + a_2C_2C_{qq1} + a_3C_{q2q3}C_{qq1} \\ Y_e^w &= y_0 + l_bS_q + a_2C_{q2}S_{qq1} + a_3C_{q2q3}Sqq1 \\ Z_e^w &= z + a_1 + a_2S_{q2} + a_3S_{q2q3} \end{aligned} \tag{3.55}$$

where $z$ is the height of the MR.

Using Equation 3.55 Jacobian matrix can be calculated by derivating the direct kinematics function w.r.t the joint variables.". The main objective of the differential kinematics is finding the relationship between the joint variable velocity and endeffector linear velocity. Having Eq(3.55) the Jacobian of the MM by employing the partial derivative of $X_w^e, Y_w^e$ and $Z_w^e$ with respect to $x_b, y_b, q, q_1, q_2, q_3$ can be calculated. The jacobian will have the following eighteen

entry's:

$$
\begin{aligned}
J_{11} &= 1 \\
J_{12} &= 0 \\
J_{13} &= -l_b S_q - a_2 C_{q2} S_{qq1} - a_3 C_{q2q3} S_{qq1} \\
J_{14} &= -S_{qq1}(a_2 C_{q2} + a_3 C_{q2q3}) \\
J_{15} &= -C_{qq1}(a_2 S_{q2} + a_3 S_{q2q3} \\
J_{16} &= -a_3 C_{qq1} S_{q2q3})
\end{aligned}
\tag{3.56}
$$

$$
\begin{aligned}
J_{21} &= 0 \\
J_{23} &= I_b C_q + a_2 C_{q2} C_{qq1} + a_3 C_{q2q3} C_{qq1} \\
J_{24} &= C_{qq1}(a_2 C_{q2} + a_3 C_{q2q3}) \\
J_{25} &= -S_{qq1}(a_2 S_{q2} + a_3 S_{q2q3}) \\
J_{26} &= -a_3 S_{qq1} S_{q2q3}) \\
J_{31} &= J_{32} = J_{33} = J_{34} = 0 \\
J_{32} &= 0 \\
J_{35} &= C_{q2} + a_3 C_{q2q3} \\
J_{36} &= a_3 C_{q2q3}
\end{aligned}
\tag{3.57}
$$

$$
J = \begin{bmatrix}
J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\
J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \\
J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36}
\end{bmatrix}
\tag{3.58}
$$

$$
\dot{\psi} = J * \dot{q} \tag{3.59}
$$

where

$$
q = \begin{bmatrix} x_b & y_b & q & q_1 & q_2 & q_3 \end{bmatrix}^T \tag{3.60}
$$

$$\psi = \begin{bmatrix} X_{ee} & Y_{ee} & Z_{ee} \end{bmatrix}^T \tag{3.61}$$

substituting (3.32), on to (3.59) the mobile manipulator kinematic model is obtained as

$$\begin{bmatrix} \dot{X}_{ee} \\ \dot{Y}_{ee} \\ \dot{Z}_{ee} \end{bmatrix} = \begin{bmatrix} C_q & -aS_q - S_{qq_1}\left(l_2C_{q_2} + l_3C_{q_2q_3}\right) & -S_{qq_1}\left(l_2C_{q_2} + l_3C_{q_2q_3}\right) \\ S_q & aC_q + C_{qq_1}\left(l_2C_{q_2} + l_3C_{q_2q_3}\right) & C_{qq_1}\left(l_2C_{q_2} + l_3C_{q_2q_3}\right) \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -C_{qq_1}\left(l_2S_{q_2} + l_3S_{q_2q_3}\right) & -l_3C_{qq_1}S_{q_2q_3} \\ -S_{qq_1}\left(l_2S_{q_2} + l_3S_{q_2q_3}\right) & -l_3S_{qq_1}S_{q_2q_3} \\ l_2C_{q_2} + l_3C_{q_2q_3} & l_3C_{q_2q_3} \end{bmatrix} \begin{bmatrix} u \\ \omega \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

## 3.5 Inverse Kinematics

Inverse kinematics represents the procedure of finding the joint angles or velocities that correspond to a given end-effector position and orientation, or velocities in Cartesian space. It involves converting the MM's end-effector coordinates from Cartesian space to joint space. Cartesian space encompasses the position and orientation of the end effector, while joint space includes the variables associated with each joint of the mobile manipulator. There are two analytical methods to solve the inverse kinematics problem. They are geometric solution approach and algebraic solution approach. As mentioned earlier, only the redundant mobile manipulator is considered, resulting in a non-square Jacobian matrix. This means that traditional methods cannot be used to directly invert the Jacobian matrix. Moreover, finding the solution for inverse kinematics problem is difficult due to the following reasons:

- More than one solution may exist.
- Equations are non-linear in most the cases, thus it may not be possible always to find the closed-loop solution all times.
- If the manipulator is kinematically redundant, there may be infinite solutions.

One of the most commonly used methods to invert a non-square matrix in redundant manipulator controls is the pseudo-inverse.

### 3.5.1 Pseudo-Inverse of the Jacobian

If the Jacobian matrix has full rank and is square, the joint velocities needed to get the required end-effector motion will be unique and it is possible to calculate as follows:

$$\dot{q} = J^{-1} * \dot{\psi} \tag{3.62}$$

Pseudo-inverse, can be used to calculate the inverse kinematics for non square matrix's and presented as follows

$$J^{-1} = J^{T}.(J.J^{T})^{-1}$$

The joint velocity vector $\dot{q}$, in Equation (3.62), is the least norm solution of Equation (3.59) [18], which provides $\dot{q}_{sys}$ with minimum Euclidean norm ($min||\dot{q}||$). When an exact solution does not exist (the Jacobian matrix is not full row rank), Equation (3.62) produces a Least-Square (LS) solution which minimizes Euclidean norm of errors while keeping the joint velocity vector norm to a minimum values. This can be represented as follows:

$$min||\dot{\psi} - J.\dot{q}|| \tag{3.63}$$

while keeping $min||\dot{q}||$

## 3.6 Mobile Manipulator Dynamic Modeling

Mathematical modeling of the dynamics of nonholonomic UGV mobile robots and MM systems involves formulating motion equations that describe the total energy of the system. [18]. Dynamic analysis is about finding the relation between the generalized co-ordinates $q$ and the generalized forces $\tau$. In general, the dynamics of a mobile manipulator can be formulated employing two different mechanisms: closed-form Lagrange-Euler formulation and forwardbackward recursive Newton-Euler formulation.The first approach considers the MM as a single entity and performs the analysis by employing Lagrange functions(i.e the difference of kinetic energy and potential energy of the MM), which compose each component of the mobile manipulator. The Newton-Euler method details combined rotational dynamics and transnational dynamics of a rigid body with respect to center of mass of each link's. Dynamics of the entire mobile

manipulator is described employing the recursive forward-backward equations. Thus, the two different formulations offer distinct perspectives on the physical interpretation of dynamics. A preferred approach is to use a closed-form equation like the Lagrange-Euler formulation, as it enables controllers to derive the time evolution of generalized coordinates.

**Lagrange mathematical technique:** This equation represents an energy balance and is thus well-suited for analyzing the interconnections of movement within a system. In this section, the Lagrangian approach will be used to obtain the relevant elements that describe the complete MM in a straightforward yet comprehensive manner.

### 3.6.1 Dynamic Modeling of MR

The Lagrange methodology for the dynamics robot has the following structure.

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \tag{3.64}$$

where: $q_i$ is generalized coordinates, and $\tau_i$ is the torque applied to the robot joints actuators. The Lagrange, $L$ (Eqn. 3.40), is defined as:

$$L = K - P \tag{3.65}$$

where $K$ is the kinetic energy and $P$ potential energies of the robots. The kinetic energy is calculated using:

$$T = \frac{1}{2} m \dot{v}^T \dot{v} + \frac{1}{2} \dot{q}^T I \dot{q} \tag{3.66}$$

where $m$ is the mass of the corresponding component, $v$ is the linear velocity of the robots component, $I$ is inertia , and $q$ is angular velocity.

Center of gravity position coordinates of the mobile robot shown in Fig. (3.3), are formulated as:

$$\begin{aligned} x_b &= x_0 + l_b C_q \\ y_b &= y_0 + l_b S_q \end{aligned} \tag{3.67}$$

The velocity of the mobile platform center of gravity, used for calculating the kinetic energy,

is as follows:

$$\begin{aligned}\dot{x}_b &= \dot{x}_0 - l_b\dot{q}S_q \\ \dot{y}_b &= \dot{y}_0 + l_b\dot{q}C_q\end{aligned} \tag{3.68}$$

From Eqn. (3.66) the kinetic energy of mobile robot is:

$$T = \frac{1}{2}m_b v_b^2 + \frac{1}{2}I_b\dot{q}^2 \tag{3.69}$$

where

$$\begin{aligned}{v_b}^2 &= {\dot{x}_0}^2 + {\dot{y}_0}^2 - 2\dot{x}_0 l_b\dot{q}S_q + 2\dot{y}_0 l_b\dot{q}C_q + {l_b}^2\dot{q}^2 \\ \dot{q}^2 &= \frac{\left(\dot{q}_r - \dot{q}_l\right)^2 r^2}{4b^2}\end{aligned} \tag{3.70}$$

resulting in the final kinetic energy computed using Eqn. (3.49).

$$K = \frac{1}{2}m_b[{\dot{x}_0}^2 + {\dot{y}_0}^2 - 2\dot{x}_0 l_b\dot{q}S_q + 2\dot{y}_0 l_b\dot{q}C_q + l_b^2\dot{q}^2] + \frac{1}{2}I_b[\frac{(\dot{q}_r - \dot{q}_l)^2 r^2}{4l^2}] \tag{3.71}$$

Next, the step-by-step methodology used to formulate the dynamic equations using the aforementioned generalized coordinates and the Lagrangian will be presented.

$$\begin{aligned}\frac{\partial L}{\partial \dot{x}_0} &= m_b\dot{x}_0 - m_b l_b\dot{q}S_q \\ \frac{\partial L}{\partial \dot{y}_0} &= m_b\dot{y}_0 + m_b l_b\dot{q}C_q \\ \frac{\partial L}{\partial \dot{q}} &= 2m_b l_b^2\dot{q} - m_b\dot{x}_0 l_b S_q + m_b\dot{y}_0 l_b C_q + I_b\dot{q}\end{aligned} \tag{3.72}$$

Derivating the above mentioned three equations will results in

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_c}\right) &= m_b\ddot{x}_0 - m_b l_b S_q\ddot{q} - m_b l_b C_q\dot{q}^2 \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}_c}\right) &= m_b\ddot{y}_0 - m_b l_b\ddot{q}C_q - m_b l_b S_q\dot{q}^2\end{aligned} \tag{3.73}$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}}) = I_b\ddot{q} + 2m_b I_b\ddot{q}^2\ddot{q} + m_b l_b\ddot{x}_c S_q + m_b l_b\dot{x}_0\dot{q}C_q - m_b l_b\ddot{y}_0 C_q + m_b\dot{y}_0 l_b\dot{q}S_q \tag{3.74}$$

The remaining derivatives needed to fulfill the Lagrange equations are provided as follows:

$$\frac{\partial L}{\partial x_0} = \frac{\partial L}{\partial y_0} = 0$$

$$\frac{\partial L}{\partial q} = m_b \dot{x}_0 l_b \dot{q} C_q + m_b \dot{y}_0 l_b \dot{q} S_q \tag{3.75}$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}}) - \frac{\partial L}{\partial q} = I_b + m_b l_b^2 \ddot{q} - m_b l_b \ddot{x}_0 S_q + m_b l_b \ddot{y}_0 C_q$$

$$\begin{bmatrix} m_b & 0 & -m_b l_b \sin q \\ 0 & m_b & m_b l_b C_q \\ -m_b l_b S_q & m_b l_b C_q & I_b + m_b^2 \end{bmatrix} \ddot{q} + \begin{bmatrix} -m_b l_b \dot{q}^2 C_q \\ -m_b l_b \dot{q}^2 S_q \\ 0 \end{bmatrix} =$$

$$\frac{1}{r} \begin{bmatrix} C_q & C_q \\ S_q & S_q \\ 2l & -2l \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} - A^T(q)\lambda \tag{3.76}$$

In order to eliminate the Lagrange multiplier in Eqn. (3.76) by having multiplied both sides of the Equation(3.76) by $S(q)^T$ where $S(q)^T A(q)^T = 0$. Which results to the reduced dynamic model in Eqn. (3.77). The reduced state-space model can be obtained by solving for $\dot{v}$ and $\dot{q}$ so dynamic control equation the nonholonomic constrained the MR is:

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

$$\dot{v} = \bar{M}(q)^{-1}[-\bar{C}(q,\dot{q})v - \bar{G}(q) - \bar{\tau}_d + \bar{E}(q)\tau] \tag{3.77}$$

### 3.6.2 Dynamic Modeling of Mobile Manipulator (MR + Manipulator Arm)

Assuming an evenly distributed mass on the MM the coordinates of the body's center of mass of the MM then the components of the MM's position in terms of its frame of reference are:

$$\begin{aligned} x_b &= x_0 + l_b C_q \\ y_b &= y_0 + l_b S_q \\ x_b &= x_1 \\ y_b &= y_1 \end{aligned} \tag{3.78}$$

where $l_b$ represents the distance between the manipulator arm base and the COG of the

mobile robot.

$$\begin{aligned} x_2 &= x_0 + l_b C_q + a_2 C_{q2} C_{qq1} \\ y_2 &= y_0 + l_b S_q + a_2 C_{q2} S_{qq1} \\ z_2 &= a_1 + a_2 S_{q2} \end{aligned} \tag{3.79}$$

(3.80)

$$\begin{aligned} x_3 &= x_0 + l_b C_q + a_2 C_{q2} C_{qq1} + a_3 C_{q2q3} C_{qq1} \\ y_3 &= y_0 + l_b S_q + a_2 C_{q2} S_{qq1} + a_3 C_{q2q3} S_{qq1} \\ z_3 &= a_1 + a_2 S_{q2} + a_3 S_{q2q3} \end{aligned} \tag{3.81}$$

The components of the MM's velocity in terms of its frame of reference are:

$$\begin{aligned} \dot{x}_b &= \dot{x} - l_b \dot{q} S_q \\ \dot{y}_b &= \dot{y} + l_b \dot{q} C_q \end{aligned} \tag{3.82}$$

$$\dot{x}_2 = \dot{x} - l_b \dot{q} S_q - a_2 S_{q2} C_{qq1}(\dot{q}_2) - q_2 C_{q2} S_{qq1}(\dot{q} + \dot{q}_1) \tag{3.83}$$

$$\dot{y}_2 = \dot{y} + l_b \dot{q} C_q + r_2 C_{q2} C_{qq1}(\dot{q} + \dot{q}_1) - a_2 S_{q2} S_{qq1}(\dot{q}_2) \tag{3.84}$$

$$\dot{z}_2 = r_2 C_{q2}(\dot{q}_2) \tag{3.85}$$

$$\begin{aligned} \dot{x}_3 =& \dot{x} - l_b \dot{q} S_q - a_2 S_{q2} C_{qq1}(\dot{q}_2) - a_2 C_{q2} S_{qq1}(\dot{q} + \dot{q}_1) \\ & - a_3 C_{q2q3} S_{qq1}(\dot{q} + \dot{q}_1) - a_3 C_{qq1} S_{q2q3}(\dot{q}_2 + \dot{q}_3) \\ \dot{y}_3 =& \dot{y} + l_b \dot{q} C_q + a_2 C_{q2} C_{qq1}(\dot{q} + \dot{q}_1) - a_2 S_{q2} S_{qq1}(\dot{q}_2) \\ & + a_3 C_{qq1} C_{q2q3}(\dot{q} + \dot{q}_1) - a_3 S_{qq1} S_{q2q3}(\dot{q}_2 + \dot{q}_3) \\ \dot{z}_3 =& a_2 C_{q2}(\dot{q}_2) + a_3 C_{q2q3}(\dot{q}_2 + \dot{q}_3) \end{aligned} \tag{3.86}$$

Using Eqns. (3.82 - 3.86) the kinetic energy of the mobile manipulator can be calculated. These equations are going to be used for combining the dynamics of the MR and robotic arm in to a single MM model. The kinetic energy is computed as:

$$T = T_m + T_1 + T_2 + T_3 \tag{3.87}$$

$$= \frac{1}{2}m_b\left(\dot{x}^2+\dot{y}^2\right)+\frac{1}{2}I_b\dot{q}^2+\frac{1}{2}m_1\left(\dot{x}_1^2+\dot{y}_1^2\right)+\frac{1}{2}I_1\left(\dot{q}+\dot{q}_1\right)^2$$
$$+\frac{1}{2}m_2\left(\dot{x}_2^2+\dot{y}_2^2+\dot{z}_2^2\right)+\frac{1}{2}I_2\left(\left(\dot{q}+\dot{q}_1\right)^2+\frac{1}{2}I_2\dot{q}_2^2\right)$$
$$+\frac{1}{2}m_3\left(\dot{x}_3^2+\dot{y}_3^2+\dot{z}_3^2\right)+\frac{1}{2}I_3\left(\dot{q}+\dot{q}_1\right)^2+\frac{1}{2}I_3\left(\dot{q}_1^2+\dot{q}_2^2\right)$$

The mobile manipulators potential energy equation is defined as:

$$P = m_1ga_1+m_2g\left(a_1+a_2S_{q2}\right)+gm_3\left(a_1+a_2S_{q2}+a_3S_{q2q3}\right))\quad(3.88)$$

recalling the above mentioned Lagrangian equation and inserting the kinetic and potential energy equation of MM and derivating w.r.t generalized coordinates and time

$$L = K - P \quad(3.89)$$

This results in the following form, which is a standard compact format commonly used in traditional robotic manipulator arm control:

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+F(\dot{q})+G(q)+\tau_d = E(q)\tau+A^T(q)\lambda \quad(3.90)$$

where $m_b$, and $I_b$, are mass of the mobile platform and moment of inertia of mobile platform $m_1, m_2, m_3$ and $I_1, I_2, I_3$ are masses and moments of inertia of links 1,2 and 3, respectively, $A(q)$ is the nonholonomic constraint, and $S(q)$ is the null state space matrix with: and

$$A(q) = \begin{bmatrix} -sin(q) & cos(q) & -1 & 0 & 0 & 0 \end{bmatrix} \quad(3.91)$$

$$S(q) = \begin{bmatrix} \frac{r}{2}C_q+\frac{r}{2l}S_q & \frac{r}{2}C_q-\frac{r}{2l}S_q & 0 & 0 & 0 \\ \frac{r}{2}S_q-\frac{r}{2l}C_q & \frac{r}{2}S_q+\frac{r}{2l}C_q & 0 & 0 & 0 \\ -\frac{r}{2l} & \frac{r}{2l} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad(3.92)$$

The inertia matrix $M(q)$ of the dynamics of the MMs is defined as:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} & M_{36} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} & M_{56} \\ M_{61} & M_{62} & M_{63} & M_{64} & M_{65} & M_{66} \end{bmatrix} \tag{3.93}$$

$$C(q,\dot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \tag{3.94}$$

$$E(q) = \begin{bmatrix} \frac{1}{r}C_q & \frac{1}{r}C_q & 0 & 0 & 0 \\ \frac{1}{r}S_q & \frac{1}{r}S_q & 0 & 0 & 0 \\ -\frac{r}{2l} & \frac{r}{2l} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.95}$$

$$G = \begin{bmatrix} G_r & G_l & G_1 & G_2 & G_3 \end{bmatrix}^T \tag{3.96}$$

$$\tau = \begin{bmatrix} \tau_r & \tau_l & \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T \tag{3.97}$$

where $\tau_r$ and $\tau_l$ are the right and left wheel driving motor torques , and $\tau_1, \tau_2$ and $\tau_3$ are manipulator's joint torques. The detailed components of the interial, corolis and centrifugal and gravitional matrices are provided [2]. The gravitational term of the MM including MR and robot arm is given as: where:

$$\begin{aligned} \frac{\partial G}{\partial x} &= \frac{\partial G}{\partial y} = \frac{\partial G}{\partial q_1} = 0 \\ \frac{\partial G}{\partial q_2} &= gm_3 \left(a_2 C_{q2} + r_3 C_{q2q3}\right) + gm_2 r_2 C_{q2} \end{aligned} \tag{3.98}$$

and

$$\frac{\partial G}{\partial q_3} = gm_3 r_3 C_{q2q3}$$

.

Mobile Manipulator robot dynamic model characteristics: When defining a model for a mobile manipulator (MM), it is essential to ensure the following three properties are fulfilled.:
Property 1: Inertia matrix M(q) is symmetric, $M(q) = M^T(q)$, positive definite.
Property 2: Inverse of the inertia matrix $M(q)$ exists and its positive definite.
Property 3: The matrix $N(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q})$ is a skew matrix: That is $N_{ij}^T(q, \dot{q} = -N_{ij}(q, \dot{q})$

As before, the state space reduced model found by solving for $v$ and $\dot{q}$. The result is the nonholonomic constrained dynamic control equation of the MM defined as:

$$\dot{q} = S(q)v \tag{3.99}$$

$$v = \bar{M}^{-1}(q)[-\bar{C}(q, \dot{q})v - \bar{G}(q) - \bar{\tau_d} + \bar{E}(q)\tau] \tag{3.100}$$

To complete the terms needed in Equation (3.100) the Inertia, Coriolis and Centrifugal and Gravitational matrix are derived in Appendix A

## 3.7 Model Verification

Model verification in control engineering refers to the process of assessing the accuracy and validity of mathematical models used to represent the a physical system. These models could be mathematical equations, simulations, or computational models that describe how the system behaves under various conditions.

The verification process typically involves comparing the predictions of the model with experimental data, empirical observations, or known physical principles. This comparison helps ensure that the model accurately captures the essential characteristics and behaviors of the real system.

### 3.7.1 Straight Line Motion Verification of MR

Giving positive and negative values of linear velocity input for the model results in straight line motion in positive and negative $X$ direction, while the motion in $Y$ direction remains zero since there is no angular velocity input.

$v$=0.1m/s and $\omega$=0

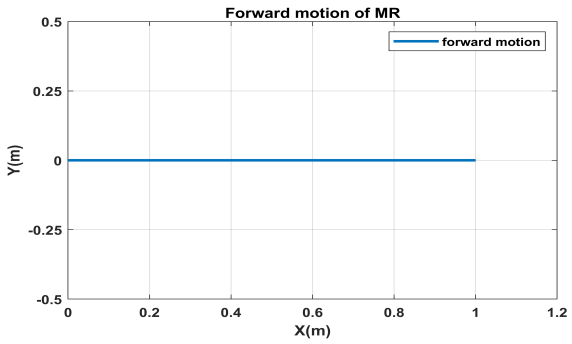



Figure 3.4: Straight line motion of mobile base in XY plane

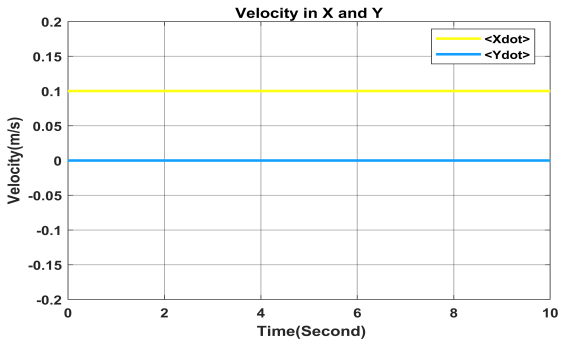



Figure 3.5: Corresponding velocity in X and Y direction

### 3.7.2 Circular motion verification of MR

Giving both linear and angular velocity values results in circular motion or more generally a curved motion of mobile platform.

$v$=0.5m/s and $\omega$=1 rad/s

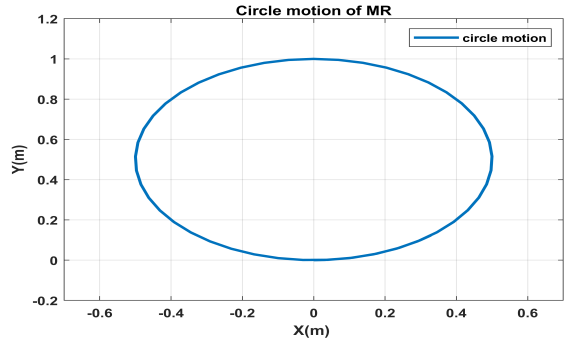



Figure 3.6: Circular motion of mobile base in XY plane

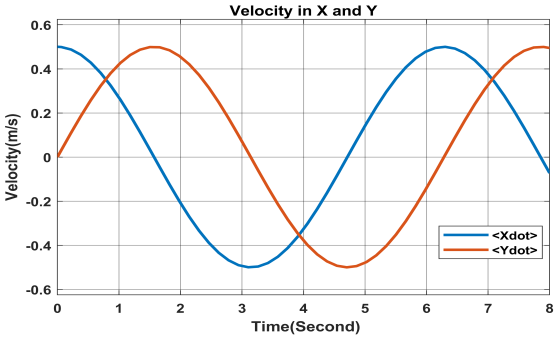



Figure 3.7: Corresponding velocity of X and Y

### 3.7.3 Spiral motion verification of MR

$v$=-0.5 m/s and $\omega$=1 rad/s

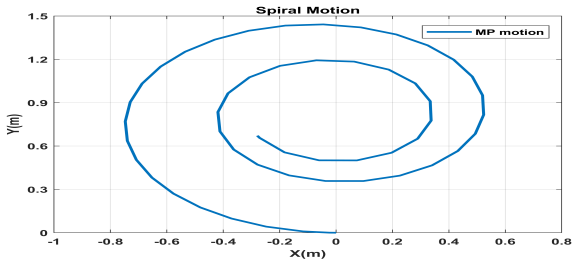



Figure 3.8: Spiral motion of mobile base in XY plane

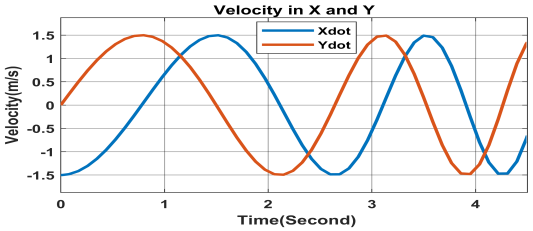



Figure 3.9: Corresponding velocity in X and Y

Giving an angle input for manipulator joints individually and together will results in different types of movement in the endeffector of the robot without exceeding its physical parameters.

Input angle is given to joint two

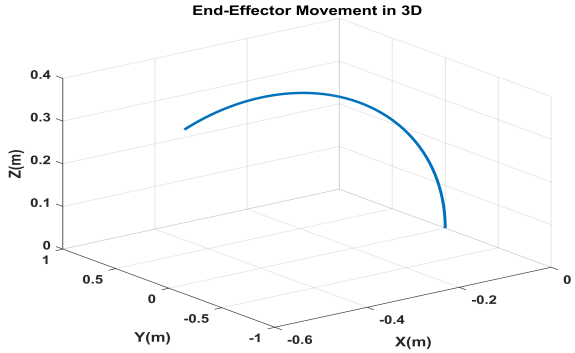



Figure 3.10: End-effector motion in space

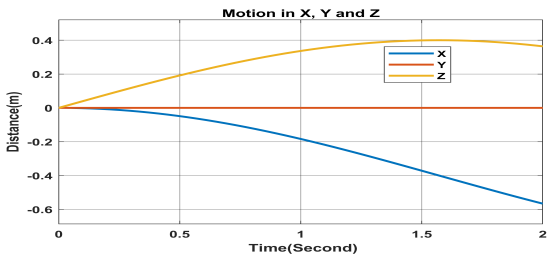



Figure 3.11: End-effector motion in X,Y,Z

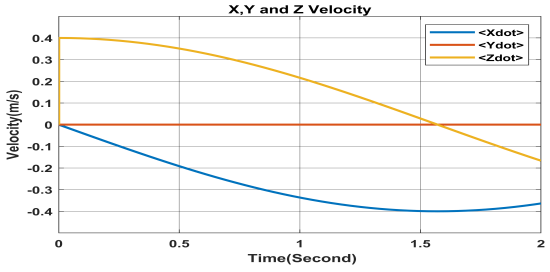



Figure 3.12: End-effector velocity in X,Y,Z

Input angle given to joint 2 and joint 3

Motion of the endeffector in space after giving the two joints an input ramp angel.

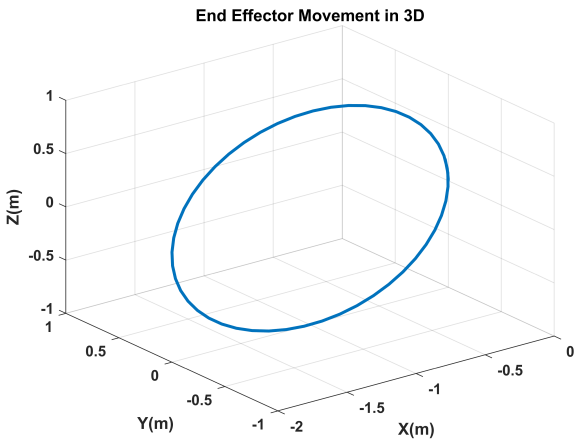



Figure 3.13: End-effector motion in space

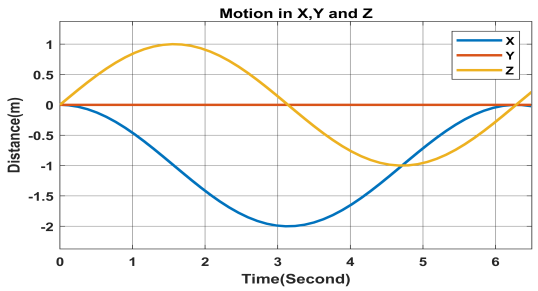



Figure 3.14: End-effector motion in X,Y,Z

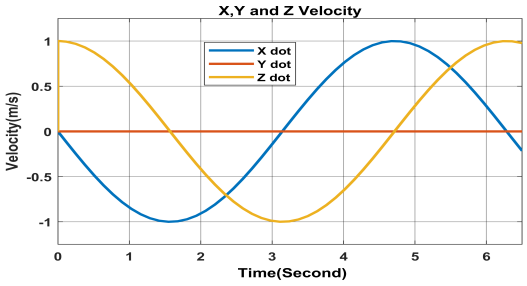



Figure 3.15: End-effector velocity in X,Y,Z

Applying a linear velocity value to the mobile base and angle values to the joints of the robot arm will results in an increase in dimension of the trajectory that is followed by the end- effector.

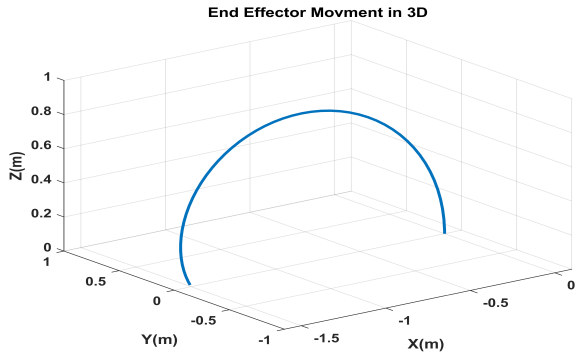



Figure 3.16: End-effector motion in 3D Space With Static MR

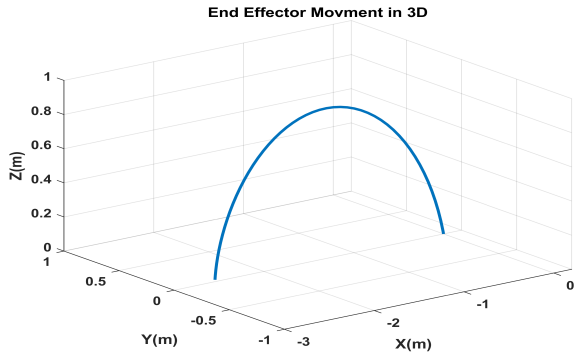



Figure 3.17: End-effector motion in 3D Space With Moving MR

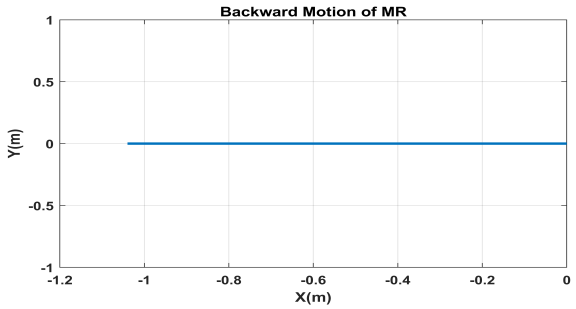



Figure 3.18: Motion of MR while the robot arm making a semicircle

In the above figure a linear velocity of $v = -0.5m/s$ for $2sec$ is given to the mobile base while the robot arm is making a semicircle path and the diameter of the semicircle path increased proportionally as it is observed.

# Chapter 4

# Controller Design

## 4.1 Introduction

This chapter provides a comprehensive and detailed explanation of fully developed control methods for MMs. Managing MMs with uncertainties is crucial in numerous practical applications, particularly when the force exerted by the end-effector required to taken into consideration. Robust and adaptive controls have been extensively researched to manage the unknown dynamics of mechanical systems, including manipulators arms and dynamic nonholonomic systems. Robust controls operate under the assumption that the systems unknown dynamics are bounded within known limits, whereas adaptive controls can learn the unknown parameters of interest using adaptive tuning laws. The proposed control architecture, a Backstepping Fuzzy Sliding Mode, was developed to handle the entire MM system, which includes both the mobile robot (MR) and the robot arm. The aim of this control is to make the end-effector of the MM the exclusive point of interaction with the environment, encompassing any entity the mobile manipulator is designed to assist. The primary objective of the developed architecture is for the end-effector, through cooperation between the MR and the arm comprising the mobile manipulator, to accurately follow a specified reference trajectory in 3D space generated by a trajectory generator.

## 4.2 Control Architecture

For improved motion control performance, it is essential to consider both the specific vehicle dynamics and kinematics. In this case, the controller should be divided into two stages. [19]. Inner loop or dynamic level control, depending on the robot dynamics, that can be used for controlling linear and angular velocity of MR and joint angle of manipulator arm according to the given required values of angular and linear velocity of MR and desired joint angle values of the robot manipulator by generating the appropriate torque values for MR wheels and joints of the robot manipulator.

Outer loop or kinematic level controller produce the necessary angular and linear velocity of the mobile platform and joint angle velocities of the manipulator arm to be an input to the inner loop dynamic controller, by taking as an input the error in end-effector trajectory. MM systems

necessitate a kinematic controller to manage the kinematic redundancy inherent in the MM, along with a dynamic controller to track the reference trajectory specified by the kinematic controller.[20] In this case the in accordance to the desired trajectory of the end-effector the backstepping kinematic controller will generate the required signals for mobile platform and also required joint angles for manipulator arm. Thus the the outputs of the kinematic controller will be an input to the inner loop dynamic controller. The dynamic controller also produce the torque value which reduces the error in mobile platform linear and angular velocities and the error in joint angles of the manipulator arm.

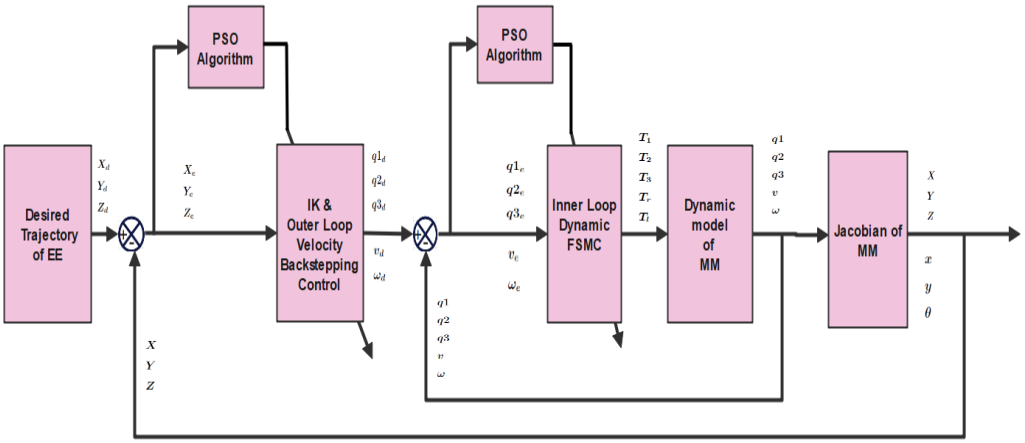



Figure 4.1: Overall Controller Architecture

Integrating both kinematic and dynamic control enables robots to achieve greater accuracy and smoother motion. Kinematic control aims to achieve desired positions or velocities, whereas dynamic control manages forces and torques for stability and effective control across different conditions.Dynamic control stabilizes the robot by addressing external disturbances, nonlinearities and uncertainties within the system. Integrating dynamic control with kinematic control in a cascaded manner enables the system to adapt to changes and maintain stability more effectively. Cascaded control systems generally offer enhanced robustness against disturbances and uncertainties when compared to single-layer control methods.

In summary, integrating both kinematic and dynamic control in a cascaded manner provides a holistic approach to robot control, ensuring a balance of precision, stability, efficiency, and safety across diverse tasks and operating conditions.

## 4.3 Backstepping Controller

Backstepping control mechanism is introduced in 1990 by Petar V. Kokotovic and colleagues to formulate stabilizing controllers for a particular type of dynamical systems which are nonlinear.[21] And those systems were composed of subsystems which extend outward from irreducible subsystem, which can get stabilized employing alternative ways. Due to the shown recursive nature, the designer can initiate the design process from the known stable subsystem and develop controllers that ultimately stabilize each subsequent outer subsystem. The process concludes when the final external control is established. Therefore, this process is referred to as Backstepping. The backstepping technique is especially advantageous for nonlinear systems since it makes possible controller designing process that can effectively manage the intricacies inherent in such systems. It is frequently applied in fields such as robotics, aerospace systems, and other domains characterized by nonlinearities and interconnected subsystems.

The backstepping controller is renowned for its capability to manage complex systems, although its design can be intricate, necessitating a profound grasp of system dynamics and meticulous choice of Lyapunov functions. Moreover, the method is sensitive to measurement noise and uncertainties, making robustness considerations crucial during the design phase.

### 4.3.1 Backstepping controller Desing for MM Veloctiy Control

In this section, the backstepping approach will be utilized to design a kinematic or velocity controller. To develop a motion tracking controller, it is used Eq. (3.42), which represents the kinematic equation specific to the mobile manipulator.

Employing the inverse of the kinematic model which is derived in the previous sections the control law in Eq. 4.38 is formulated, which enables determining the speeds for the mobile manipulator robot to follow the desired trajectory where $\psi$ are the required positions, with the error in position $\mathbf{e}$, $J^{-1}$ is the pseudoinverse of jacobian, By representing the system output as

$$\psi = h(q) \tag{4.1}$$

where $e$ is the error in position of the mobiel manipulator end effector. Derivative of the

Eq.(4.3)

$$\dot{\psi} = \frac{\partial h(q)}{\partial q}\dot{q} = J\sigma \tag{4.2}$$

let $\sigma = \dot{q}$, where $J$ is a extended Jacobian matrix and can be formulated as

$$J = \frac{\partial h(q)}{\partial q} \tag{4.3}$$

From Eqs. (4.4) and (4.5),

$$\dot{e} = \dot{\psi}_d - J\sigma \tag{4.4}$$

Let $\sigma = \sigma_c$ , as the virtual velocity control,Eq.(4.7) can be formulated as,

$$\dot{e} = \dot{\psi}_d - J\sigma_c \tag{4.5}$$

By selecting the Lyapunov function as,

$$V_1 = \frac{1}{2}\mathbf{e}^T\mathbf{e}. \tag{4.6}$$

Taking the derivative gives,

$$\dot{V}_1 = \mathbf{e}^T\dot{\mathbf{e}} \tag{4.7}$$

$$= \mathbf{e}^T\left(\dot{\mathbf{\Psi}}_d - J\boldsymbol{\eta}_c\right) \tag{4.8}$$

Choosing the kinematic control law $\boldsymbol{\sigma}_c$

$$\eta_c = J^{-1}\left[\dot{\Psi}_d + K\mathbf{e}\right], \tag{4.9}$$

where $K > 0$ is the constant positive control gain. Eq.(4.) is the velocity tracking kinematic controller. From Eqs.(4.6) and (4.10)

$$\dot{\mathbf{e}} = \dot{\Psi}_d - J\left[J^{-1}\left(\dot{\Psi}_d + K\mathbf{e}\right)\right] = -K\mathbf{e} \tag{4.10}$$

$$\dot{\mathbf{e}} = -\mathbf{K}\mathbf{e} \tag{4.11}$$

by solving the above ODE error value can be found as

$$\mathbf{e_r} = constant * \mathbf{e_{xp}}^{-Kt} \tag{4.12}$$

From Eq.(4.12) when $K > 0, \lim_{t\to\infty}\mathbf{e}(t) = 0$.

$$\dot{\mathbf{e}} = -\mathbf{K}\mathbf{e}$$

$$\dot{V}_1 \leq -K\|\mathbf{e}\|^2 \tag{4.13}$$

## 4.4 Sliding Mode Control

The idea of variable structure control (VSC) and its related sliding modes was initially developed and used by Vadim Utkin and Stanislav Emelyanov in the begining of 1950s in the Soviet Union. Among the greatest significant aspects of SMC is its capability to produce control algorithms which is robust and that remain invariant given specific conditions. In essence, the concept of invariance means that the system is unaffected by certain kinds of uncertainties and disturbances. [22] From the 1990s onward, there has been growing interest in controlling systems that are affected by model uncertainty and external disturbances. Among the various alternatives available, sliding mode (SM) control has emerged as a compelling option for implementation in electronically controlled systems. It has demonstrated high robustness and resilience against system uncertainties and perturbations. The practicality and advantages of using sliding mode (SM) control with Actuators with electronic control have been well-documented in the literature, as shown by [23]. Furthermore, sliding mode (SM) control offers a relatively straightforward design methodology, capable of handling nonlinear systems effectively. It also facilitates successful integration alongside other nonlinear control methods like fuzzy logic and model predictive control. Consequently, research and development in the design methods of sliding mode (SM) control have seen significant acceleration, both in theoretical advancements and practical applications. [24, 25, 26]

A prominent key feature of sliding mode (SM) control is the nature of its control action which is discontinous. Its main purpose involves shifting between two varied structures to achieve the required new dynamic behavior on the system, which is sliding-mode dynamics. This attribute enables the system to achieve improved performance, including robustness against parametric uncertainties and disturbance rejection, particularly when matching conditions are satisfied. [22, 27]. When addressing uncertainties in parameters, it encompasses uncertainties in both internal and external parameters, which arise from the reduction model process employed in control design. [28, 29]

Sliding mode control employs control laws i.e discontinuous feedback control law, to guide the system state to attain and then maintain a specific surface within the state space, referred

to as the sliding surface.There are two benefits of achieving such motion: firstly, the system operates as a reduced-order system relative to the original plant, and secondly, movement on the sliding surface ensures the system remains unaffected by certain types of perturbations and model uncertainties. Sliding mode control is typically applied to nonlinear systems with uncertain parameters or disturbances, Multiple Input Multiple Output systems, discrete-time models, large-scale systems, and infinite-dimensional systems.

Designing sliding mode control (SMC) offers a structured method to ensure stability and reliable performance despite modeling inaccuracies. Over the years, a variety of techniques have been developed for designing sliding mode controllers. However, the foundations of all these different design techniques are quite similar and typically involve two main steps. Create the sliding surface in the state space to enable the reduced-order sliding motion. Develop the control law so that the closed-loop trajectories move towards the sliding surface and then stay on or near the surface thereafter.

Sliding mode controller design can be divide in two steps:

**1:** Choosing stable hyperplanes in the state or error space where motion is constrained is referred to as selecting the sliding surface, and

**2:** Designing a discontinuous control law that directs the system trajectory to the sliding surface and keeps it there.

$$U = U_{eq} + U_{dis} \tag{4.14}$$

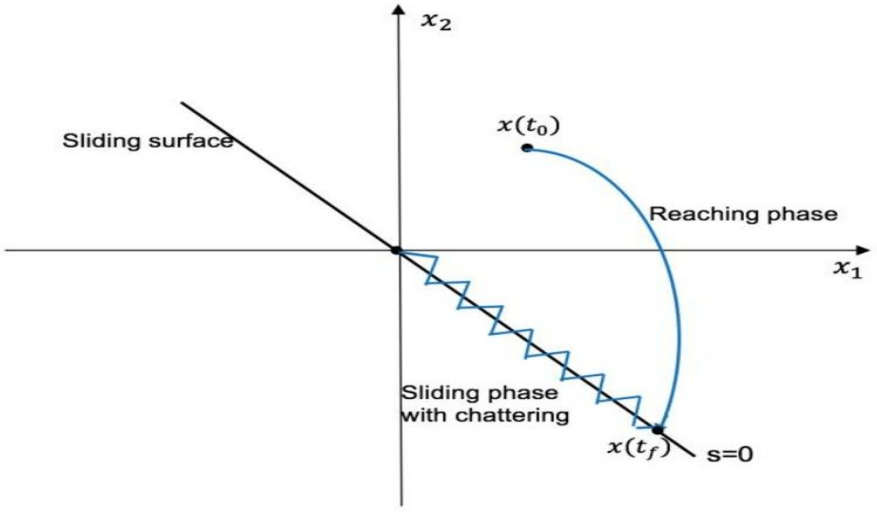



Figure 4.2: Behavior of the sliding mode controlled system

A trajectory of a sliding mode controller, beginning from a non-zero initial condition, occurs in

two different operational modes. In the reaching mode, the trajectory of the controller reaches the sliding surface. In the sliding mode, once the trajectory reaches the sliding surface, it remains on the surface indefinitely, following the dynamics dictated by the sliding surface.

**Design of switching function:** There are many switching functions that can be used in sliding mode control, but all of these functions must have the following properties. The order of the switching function is lower than the order of the plant; for instance, a second-order system typically requires a first-order switching function. Sliding mode control is independent of plant dynamics and is defined by the parameters of the switching function.

### 4.4.1 Sliding Mode Control Design for Mobile Manipulator

Now that the kinematic controller for the MM has been developed (Section 4.3) it's used such work to develop a controller for the MM of interest. The proposed control architecture is illustrated in Figure 4.1. The goal of such architecture is to ultimately compute the control torque to control the mobile manipulator so that the error between the joint position vector $q$ and the desired joint position $q_d$ can decay to zero within finite time.

$$\dot{q} = S(q)v \tag{4.15}$$

$$\dot{v} = \ddot{q} = \bar{M}^{-1}(q)[-\bar{C}(q,\dot{q})v - \bar{G}(q) + \bar{E}(q)\tau] \tag{4.16}$$

$q = [v, q, q_1, q_2, q_3]^T, \tau = [\tau_r, \tau_l, \tau_1, \tau_2, \tau_3]^T, q \in R^n$ is a single vector, $\bar{M}(q) \in R^{n\times n}$ is the inertia matrix, $\bar{C}(q,\dot{q}) \in R^{n\times n}$ is the centrifugal and Coriolis forces, $\bar{G}(q) \in R^n$ is the gravity vector, $\bar{E}(q) \in R^{n\times n}$ is a nonsingular transformation matrix, and $\tau \in R^n$ is the required control torque. Let $q_d$ denote the desired trajectory of the MM must follow. Under these conditions the tracking error and its derivatives needed during the controller design is defined as

$$e = q_d - q \tag{4.17}$$

$$\dot{e} = \dot{q}_d - \dot{q} \tag{4.18}$$

$$\ddot{e} = \ddot{q}_d - \ddot{q} \tag{4.19}$$

Using typical control solutions the sliding variable selected is defined as:

$$s = e + c\dot{e} \tag{4.20}$$

where $c$ is a positive diagonal matrix, $c = c^T > 0$ and $s$ is an $n$ vector defined as:

$$c = \begin{bmatrix} c_1 & 0 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 & 0 \\ 0 & 0 & c_3 & 0 & 0 \\ 0 & 0 & 0 & c_4 & 0 \\ 0 & 0 & 0 & 0 & c_5 \end{bmatrix}, c_i > 0, i = 1, 2, ..n \tag{4.21}$$

$$s = \begin{bmatrix} s_1 & s_2 & \dots & s_n \end{bmatrix}^T \tag{4.22}$$

$$\dot{s} = \ddot{e} + c\dot{e} = 0 \tag{4.23}$$

$$= \dot{v}_d - \dot{v} + c\dot{e} \tag{4.24}$$

$$= \dot{v}_d + c\dot{e} - \bar{M}^{-1}(q)[-\bar{C}(q,\dot{q})v - \bar{G}(q) + \bar{E}(q)\tau] \tag{4.25}$$

$$= \dot{v}_d + c\dot{e} + \bar{M}^{-1}(q)\bar{C}(q,\dot{q})v + \bar{M}^{-1}(q)\bar{G}(q) - \bar{M}^{-1}(q)\bar{E}(q)\tau \tag{4.26}$$

$$\bar{M}^{-1}(q)\bar{E}(q)\tau = \dot{v}_d + ce + \bar{M}^{-1}(q)\bar{C}(q,\dot{q})v + \bar{M}^{-1}(q)\bar{G}(q) \tag{4.27}$$

$$\bar{E}(q)\tau = \bar{M}(q)(\dot{v}_d + ce) + \bar{C}(q,\dot{q})v + \bar{G}(q) \tag{4.28}$$

$$\tau_{eq} = \bar{E}^{-1}(q)[\bar{M}(q)(\dot{v}_d + ce) + \bar{C}(q,\dot{q})v + \bar{G}(q)] \tag{4.29}$$

So the equivalent control law is given by Eq(4.29)

The discontinuous control law has to direct the plant state towards the surface $s = 0$ , so the state will be driven towards $s > 0$ if $s < 0$ , and towards $s < 0$ if $s > 0$ ; at $s = 0$ the discontinuous control law has no effect on the plant. A suitable discontinuous control law can be chosen as:

$$\tau_{dis} = -k.sign(s) \tag{4.30}$$

$$\text{where } sign(s)= \begin{cases} +1 & \text{if} \quad s > 0 \\ 0 & \text{if} \quad s = 0 \\ -1 & \text{if} \quad s < 0 \end{cases}$$

Recall that

$$\tau = \tau_{eq} + \tau_{dis} \tag{4.31}$$

## 4.5 Fuzzy Logic Control

FFuzzy Logic Control (FLC) is a control approach that employs fuzzy logic to handle complex and uncertain systems. It is particularly suitable for systems that involve imprecision, vagueness, and uncertainty, where traditional control methods based on precise mathematical models may encounter difficulties. Introduced by Lotfi Zadeh in the 1960s, fuzzy logic offers a method to model and control systems that exhibit inherent vagueness or ambiguity.

A Fuzzy Logic controller consists of three primary components:

1. Fuzzification,
2. Fuzzy Rule base and Infrence engine,
3. Defuzzification.

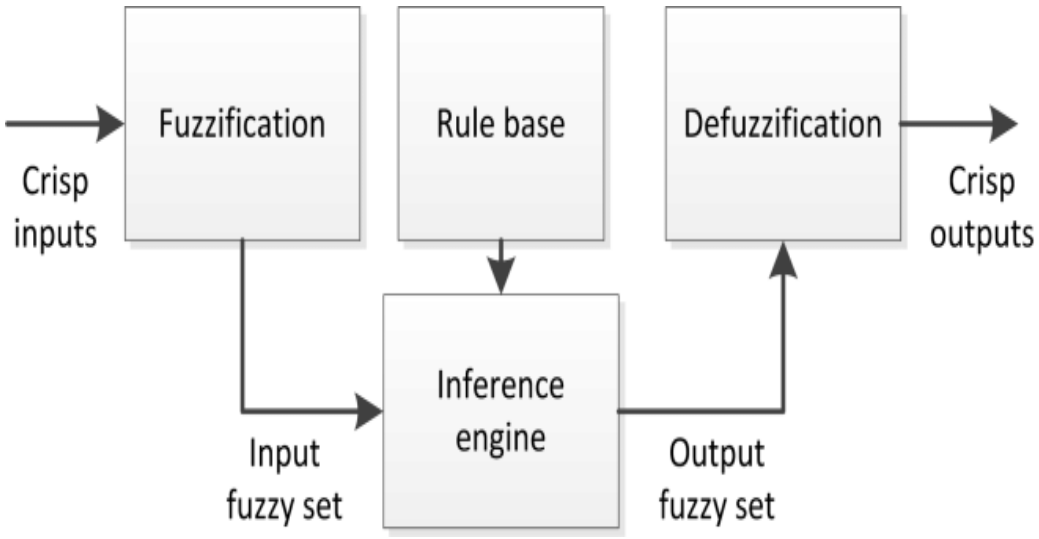



Figure 4.3: Architecture of fuzzy control

**Fuzzification:** The key aspect of designing a fuzzy controller involves identifying the state variables that can proficiently control the plant. Once the state variables are identified, they are

processed through the fuzzification block to convert the inputs into fuzzy sets, as FLC operates exclusively with fuzzy inputs. Because the Fuzzy Rule base uses rules based on linguistic variables, numerical inputs must first be transformed into fuzzy linguistic variables. The act of transforming a numerical state variable into a fuzzy linguistic input variable is termed the fuzzification process. Fuzzification is the process of transforming precise (crisp) input values into fuzzy sets by associating numerical input values with linguistic variables that have fuzzy memberships.

**Membership function:** is a fundamental element used to determine the degree of membership of an element within a fuzzy set. It quantifies how much an input or output belongs to a specific linguistic term or fuzzy set.A membership function allocates a membership value ranging from 0 to 1 to every element within the universe of discourse (the range of possible values for a variable). This value indicates the extent to which an element belongs to a specific fuzzy set. Membership functions can have various shapes, which depend on the characteristics of the variable and the linguistic terms employed. Typical shapes include triangular, trapezoidal, Gaussian, sigmoidal functions, and others. In fuzzy logic control, linguistic terms such as "low," "medium," and "high" are often used to describe input and output variables. Each linguistic term is defined by a membership function that specifies its boundaries and quantifies the degree to which elements belong to that term.

Membership functions are employed to define fuzzy sets, which consist of collections of elements

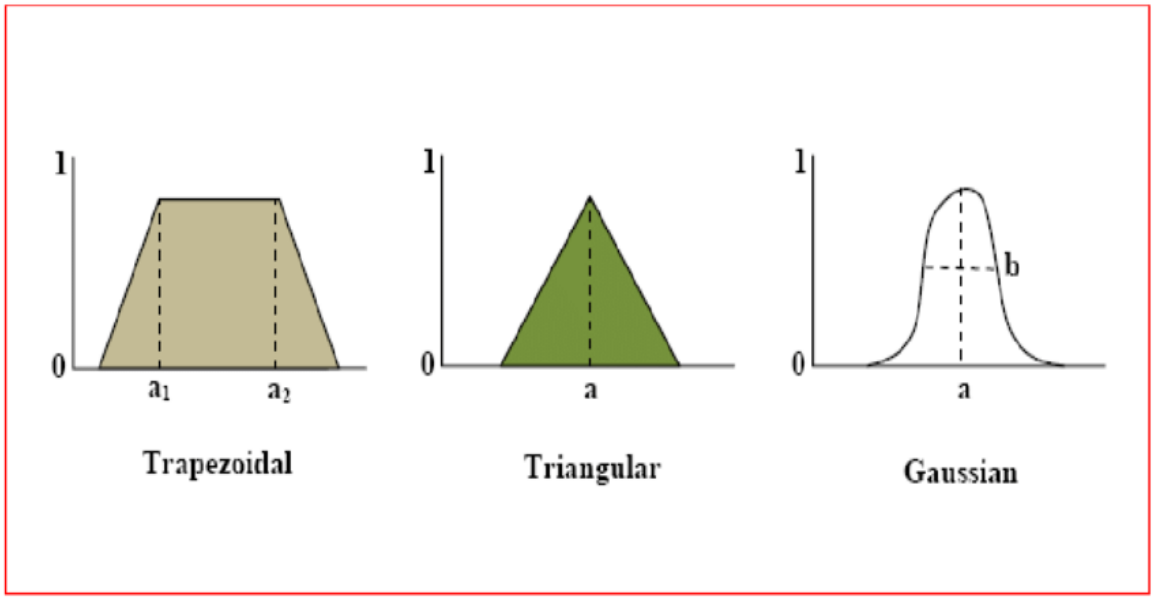



Figure 4.4: Different shapes of membership functions

characterized by varying degrees of membership. Every fuzzy set is associated with a linguistic term and is described by its membership function. In fuzzy logic control systems, inputs are

translated into fuzzy sets using membership functions, enabling the formal representation of imprecise or vague information. This capability allows the system to make decisions based on fuzzy rules and linguistic terms.

**Fuzzy Rule Inference:** Fuzzy rule inference is a crucial process in fuzzy logic systems, where fuzzy rules are used with fuzzy input variables to derive fuzzy output variables. This process entails assessing the extent to which each rule is satisfied by the input values and aggregating these degrees of satisfaction to produce fuzzy output values. After fuzzifying the input variables, each fuzzy rule is assessed to ascertain the extent to which it is satisfied by the fuzzy input values. This evaluation involves matching the antecedent (the "if" part) of each rule with the fuzzy input values. Fuzzy inference is of two methods.

A **fuzzy rule base:** Is a set of rules that encapsulate expert knowledge or heuristics pertaining to the system being controlled. These rules are employed to associate input variables with output variables in a fuzzy logic control system. Each rule typically takes the form of an "if-then" statement and describes how input variables relate to output variables in linguistic terms.

**If-Then Rules:** Fuzzy rules are expressed as conditional statements in the form of "if $X$ is $A$, then $Y$ is $B$," where $X$ and $Y$ represent input and output variables, respectively, and $A$ and $B$ represent linguistic terms associated with these variables. For example, "If temperature is high, then fan speed is fast. Fuzzy rules use linguistic terms to describe the values of input and output variables. These terms are often fuzzy sets defined by membership functions.

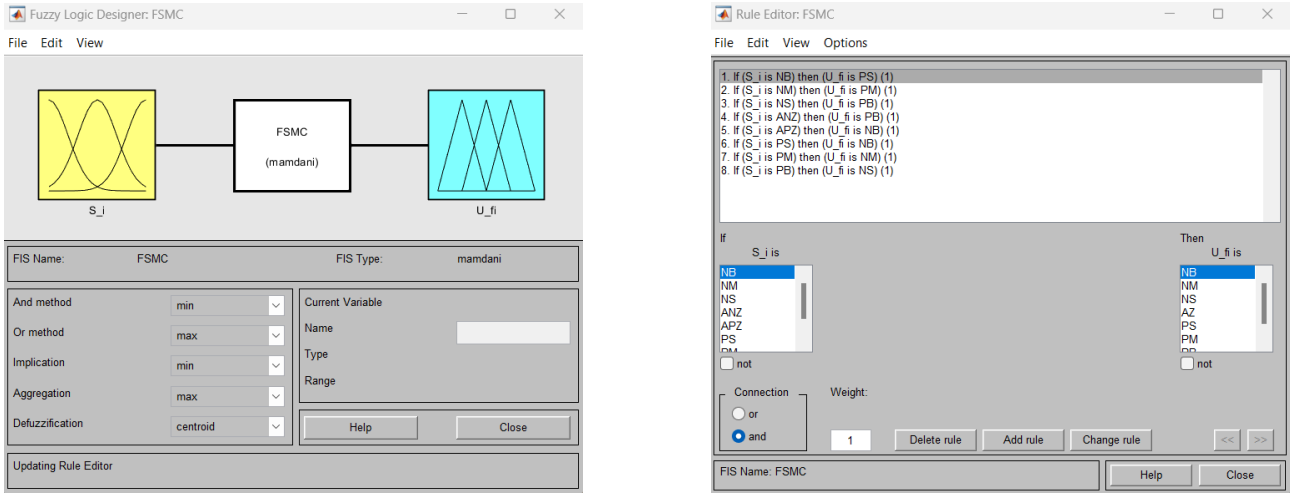
The linguistic terms used are $NB$ - Negative Big, $NM$ - Negative Medium, $NS$ - Negative Small, $ANZ$- Almost Negative Zero, $APZ$- Almost Positive Zero, $PS$- Positive Small, $PM$- Positive Medium, $PB$- Positive Big.

Rule base is selected as follows:

- if $S_i$ is $NB$ then $U_{fi}$ is $PS$
- if $S_i$ is $NM$ then $U_{fi}$ is $PM$
- if $S_i$ is $NS$ then $U_{fi}$ is $PB$
- if $S_i$ is $ANZ$ then $U_{fi}$ is $PB$
- if $S_i$ is $APZ$ then $U_{fi}$ is $NB$
- if $S_i$ is $PS$ then $U_{fi}$ is $NB$
- if $S_i$ is $PM$ then $U_{fi}$ is $NM$
- if $S_i$ is $PB$ then $U_{fi}$ is $NS$

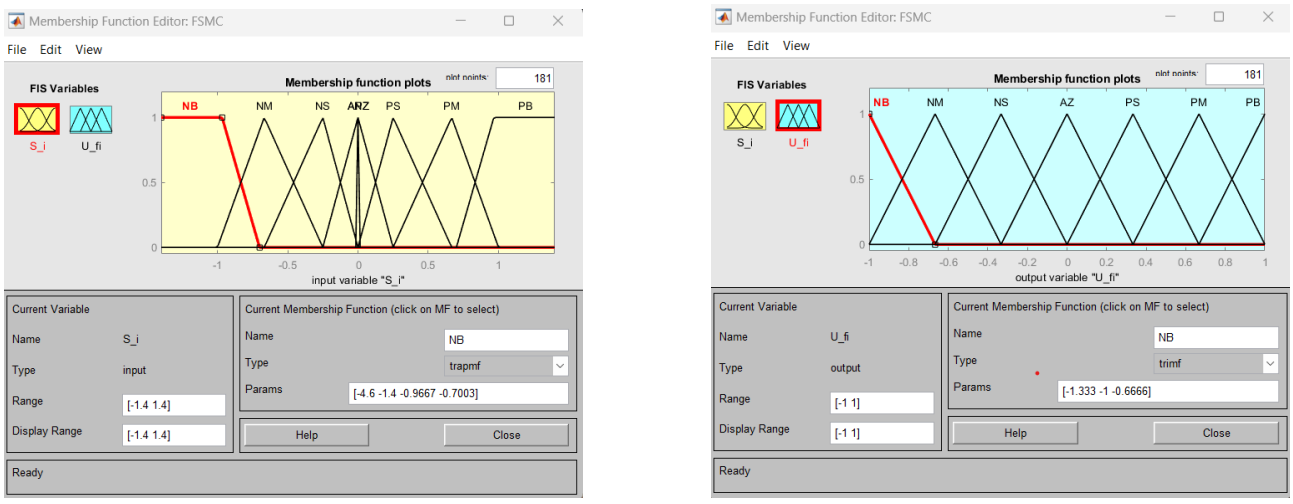
$'i'$ represents for$(q_1, q_2, q_3, v, \omega)$.

The fuzzy inference engine method employed in thesis work is Mamdani's methods of Fuzzy interface.




(a) fuzzy logic desing (b) FLC rule base

Figure 4.5: FLC with rule base




(a) FLC input membership fucntion (b) FLC output membership function

Figure 4.6: FLC membership fucntions

**Defuzzification:** Common techniques used for defuzzification include:

1. Centroid Method,
2. Bisector of Area Method,
3. Mean of minimum Method.

After the complation of fuzzy reasoning FLC generates output in the form of a linguistic variable (fuzzy number). According to the actual requirements, the linguistic variables need to be

converted into crisp outputs. The centroid method is the most widely understood defuzzification technique and is used in this research work. It calculates the center of gravity of the area covered by the fuzzy set.

**Centroid Method:** When employing the centroid method for defuzzification, the crisp output value is determined by computing the centroid, which represents the center of gravity of the fuzzy set that describes the output. In essence, this centroid provides a single value that represents the fuzzy output comprehensively. The centroid method is widely favored and straightforward due to its simplicity in converting fuzzy outputs into crisp values. The algebric expression for centroid method of defuzzification is

$$y_{COG} = \frac{\int \mu_A(x) x dx}{\int \mu_A(x) dx} \cdots \tag{4.32}$$

The fuzzy logic controller is designed on MATLAB/Simulink using fuzzy logic toolbox and it consist of one input variable and one output variable as it is depicted in the figure below. The input and output variables have eight linguistic values explained above with a trapezoidal and triangular shape membership function(MF).

Then, after finishing the designing of fuzzy logic controller, the overall control effort becomes the sum of the equivalent controller, discontinuous controller and the fuzzy logic controller *i.e*

$$U = U_{eq} + U_{dis} + U_f \tag{4.33}$$

## 4.6 Stability analysis

Sliding mode control law will have two components, the equivalent and the discontinuous control law. In order to design the sliding mode control law, the so called Lyapunov stability method is used. This method assesses the stability characteristics of an equilibrium point without needing to solve the state equation [30] by stating that: If all points that begin close to a point $x$ remain close to $x$ indefinitely, then $x$ is considered Lyapunov stable." Moreover, if all points that start out near $x$ converge to $x$, then $x$ is asymptotically stable.

The primary task in sliding mode control is to design a switched control strategy that directs the plant's state to the switching surface and maintains it on the surface once it has been attained.[31] Consider $V(x)$ as a continuously differentiable scalar function defined within a domain $D$ that encompasses the origin. $V(x)$ is said to be negative definite if $V(0) = 0$ and $\dot{V}(x) < 0$ for $x$. Lyapunov's criterion states that a linear time-invariant system is stable if there exists a scalar function $V(x)$ or which $V(x) > 0$ and $\dot{V}(x)$ is negative definite when associated

with the system.[32]

In sliding mode control, to ensure that the state trajectory moves towards the switching surface, a Lyapunov candidate function is formulated with respect to the surface. The gains of this function are chosen such that its derivative is negative definite. Once the sliding surface is designed, a switched controller is developed to steer the state towards and keep it on the sliding surface.[32] In this thesis work, Lyapunov stability theory is employed to determine the gains that enable the plant's state to reach the sliding surface in finite time and remain on it thereafter.

Define the Lyapunov function as follows:

$$V = \frac{1}{2} s^T M s. \tag{4.34}$$

And differentiating the Lyapunov function is:

$$\dot{V} = \frac{1}{2}\dot{s}^T M s + \frac{1}{2} s^T \dot{M} s + \frac{1}{2} s^T M \dot{s} = \frac{1}{2} s^T \dot{M} s + s^T M \dot{s}. \tag{4.35}$$

The dynamic model is :

$$\tau = J^T f = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \tag{4.36}$$

Then,

$$M\ddot{q} = J^T f - C\dot{q} - G. \tag{4.37}$$

Since $\dot{s} = \ddot{e} + \lambda\dot{e} = \ddot{q} - \ddot{q}_r$, where $\ddot{q}_r = \ddot{q}_d - \lambda\dot{e}$

$$\dot{V} = \frac{1}{2} s^T \dot{M} s + s^T \left( J^T f - C\dot{q} - G - M\ddot{q}_r \right). \tag{4.38}$$

Note that, with the particular property of mobile manipulator, the matrix $\dot{M}(q) - 2C(q,\dot{q})$ is indeed skew-symmetric, and based on the skew-symmetry of the matrix, $s^T[\dot{M} - 2C]s = 0$. By using this property, the term of $1/2 s^T \dot{M} s$ can be rewritten as:

$$\frac{1}{2} s^T \dot{M} s = s^T C s \tag{4.39}$$

Then, substituting (4.39) in to (4.38) yields:

$$\dot{V} = s^T(\tau - M\ddot{q}_r - C\dot{q}_r - G). \tag{4.40}$$

In the designed FSMC with fuzzy compensational signal, the controller output is

$$\tau = \tau_{eq} + \tau_{dis} + \tau_f = \hat{M}\ddot{q}_r + \hat{C}\dot{q}_r + \hat{G} - ksign(s) + \tau_f \tag{4.41}$$

Substituting $\tau$ in to (4.40 ) yields

$$\dot{V} = s^T((\hat{M} - M)\ddot{q}_r + (\hat{C} - C)\dot{q}_r + (\hat{G} - G)) - s^T ksign(s) + s^T\tau_f \tag{4.42}$$

Let us define $\Delta M = \hat{M} - M, \Delta C = \hat{C} - C, \Delta G = \hat{G} - G$
And having $\hat{M} = M(q), M = M(q_d), \hat{C} = C(q, \dot{q}), C = C(q_d, \dot{q}_d), \Delta G = 0$, So, the final Lyapunov function is expressed as:

$$\dot{V} = s^T(\Delta M\ddot{q}_r + \Delta C\dot{q}_r) - s^T ksign(s) + s^T\tau_f \tag{4.43}$$

In order to satisfy the stability condition of such control system as: $\dot{V} \leq -\eta|s|$, where the constant $\eta$ is strictly positive. The vector $k$ can be chosen as following equation:

$$k \geq |\Delta M\ddot{q}_r + \Delta C\dot{q}_r| + |\tau_f| + \eta. \tag{4.44}$$

Therefore, if the chosen control gain $k$ is large enough to satisfy the above equation, one can conclude that the reaching condition is always satisfied. As a result, the closed-loop system achieves asymptotic stability, and the trajectory of the error state converges to the sliding surface $s(t) = 0$. The sliding condition guarantees that the trajectory attains the sliding surface within a finite duration. Once intercepted, the trajectories will remain on the surface, and thus drive robot following the desired trajectory.[31]

## 4.7 Particle Swarm Optimization

The performance of a BFSMC heavily depends on effective tunning some parameter gains.[33] Employing the conventional tunning techniques shuch as trial and error method for getting optimal parameter values found to be inefficient. While trial-and-error tuning may be simple and straightforward for very basic systems, it is generally time consuming, inefficient, inconsistent, and often suboptimal for more complex control systems. More systematic and automated methods, such as Particle Swarm Optimization (PSO) or other optimization techniques, provide significant advantages in finding optimal control parameters efficiently and reliably.
Particle Swarm Optimization , introduced by Kennedy and Eberhart, is one of the contemporary heuristic algorithms.[34][35]. It was developed by modeling a simplified social system and

has demonstrated effectiveness in addressing continuous nonlinear optimization problems. The PSO technique can produce a high-quality solution within short calculation time and stable convergence characteristic [36][37]. Moreover, PSO is relatively simple compared to other optimization techniques and is an effective and efficient method for gain tuning in various systems and controller type [38][39][40]. The basic algorithm of PSO are:

1. **Initialization:** Initialize a population (swarm) of particles with random positions and velocities in the search space. Set the initial positions $x_i$ and velocities $v_i$ for each particle $i$. Each particle's position represents a potential solution to the optimization problem.

2. **Evaluate:** Evaluate the fitness of each particle using the objective function.

3. **Update Personal Best:** For each particle, compare its current fitness with its best fitness (personal best,$p_i$) achieved so far.If the current fitness is superior, update the personal best position $p_i$ and fitness value.

4. **Update Global Best:** Identify the particle with the best fitness in the entire swarm (global best,$g$). If the current fitness of any particle is better than the global best, update the global best position $g$ and fitness value.

5. **Update Velocity and Position:** For each particle, update its velocity $v_i$ using the following equation

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_i - x_i(t)) + c_2 \cdot r_2 \cdot (g - x_i(t)) \tag{4.45}$$

   where:

   - $w$ is the inertia weight, controlling the influence of the previous velocity.
   - $c_1$ and $c_2$ are cognitive and social coefficients, respectively.
   - $r_1$ and $r_2$ are random numbers between 0 and 1.

   Update the position of each particle using the new velocity

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{4.46}$$

6. **Termination:** Check the termination criteria. If the termination criteria are met, stop the algorithm. Otherwise, go back to step 2.

# Chapter 5

# Simulation Results and Analysis

## 5.1 Introduction

In this section the mathematically develop Mobile Manipulator plant model (Chapter 3) and proposed controller for the plant(Chapter 4, both BSMC and BFSMC) designed and interconnected on MATLAB®/Simulink® to perform a simulation test. The behavior of the Mobile Manipulator model is observed, and the performance of the proposed control system is evaluated. Diverse test scenarios are used (Sinusoid, Helical and Square trajectories) to test and evaluate the performance and effectiveness of the Backstepping Fuzzy sliding Mode control mechanism. Simulation results are provided with graph which demonstrates how effective is the proposed controller in tracking the given trajectories .

The parameters of the robots (Mobile Platform and Robot arm) used in the tests/simulations are shown in Table 5.1. The tests consisted on having the robot follow a referenced trajectory in 2D and 3D space. Furthermore, the performance of the proposed controller is compared to that of an conventional BSM controller. Step response and disturbance checking are performed to evaluate the controller's performance by using performance indices. A comparative analysis is conducted for both controllers in terms of trajectory tracking, considering the effects of disturbances and parameter variations. The results are discussed and interpreted based on performance metrics.

During the test the mobile platform and the robot arm are observed cooperating to make possible the end effector tracked the given trajectory. It must be noted that the selected curve always within the configuration space of the mobile platform + arm (the joint limitations are never exceeded).

Table 5.1: Mobile Manipulator robot parameters.[2]

| Parameter | Value | Units |
|---|---|---|
| Mass of the MP: $m_b$ | 50 | kg |
| Mass of robot's arm link 1: $m_1$ | 4 | kg |
| Mass of robot's arm link 2: $m_2$ | 6 | kg |
| Mass of robot's arm link 3: $m_3$ | 4 | kg |
| Radius of the wheel of the MP: $r$ | 0.06 | meter |
| Length of robot's arm link 1: $l_1$ | 0.5 | meter |
| Length of robot's arm link 2: $l_2$ | 0.5 | meter |
| Length of robot's arm link 3: $l_3$ | 0.5 | meter |
| The distance between the two wheels on the MP: $b$ | 0.2 | meter |
| Distance between the center of the wheels and manipulator base: $l_b$ | 0.06 | meter |

## 5.2 Simulink Model of Mobile Manipulator

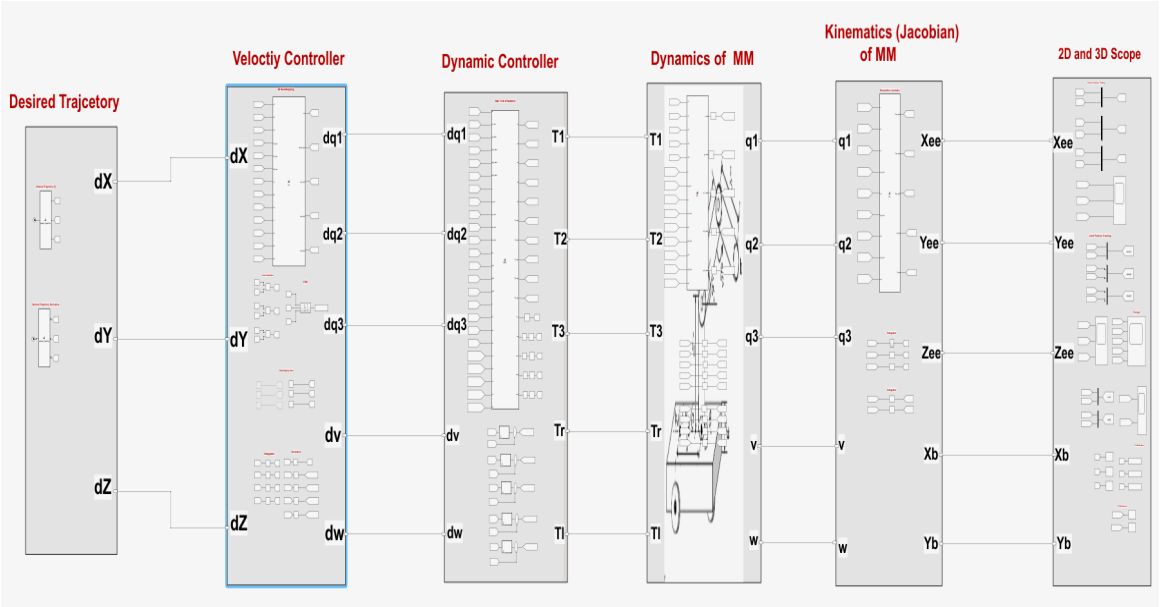



Figure 5.1: Matlab Simulink Model

The overall design comprises of the following subsystems:

- The purpose of a trajectory generator is to plan and generate a smooth and feasible path or trajectory for the Mobile Manipulator and feed it in to velocity controller as a reference trajectory.

- The kinematic (velocity )controller is the outer loop controller designed using backstepping controling technique and inverse kinematics model of Mobile Manipulator.
- The dynamic controller block is the inner loop controller designed using Fuzzy Sliding Mode and the dynamic model of the Mobile Manipulator.
- Dynamic Model block contains the dynamic model of Mobile Manipulator derived using Lagrangian method.
- The Kinematics block contains the combined kinematics or Jacobian of the Mobile Manipulator.
- The last block in the Simulink model comprises of scopes and other display tools which helps to performance analysis plots that compare the performance of the developed BFSM and conventional BSM controller.

## 5.3 3D Sinusoidal Trajectory Tracking

At the start of the simulation, the manipulator is guided to follow a specific trajectory while the platform remains stationary. Afterward, the platform begins moving horizontally directly beneath the manipulator's endpoint, ensuring that the manipulator avoids entering a singular configuration. The reference trajectory moves along a sinusoidal curve in the $XY$-plane, with an additional sinusoidal fluctuation in the $Z$-axis. This trajectory is challenging because the mobile platform must continuously adjust its heading in a sinusoidal pattern to account for the manipulator's movements in the $XY$-plane. These 3D movements of the MM result from the combined effects of the robot arms spatial motions and the mobile robot's yaw. The 3D sinusoidal reference trajectory which is given to manipulator arm to follow is obtained by

$$x_d = 0.5 + 0.2t \tag{5.1}$$

$$y_d = 0.3\sin(0.4t) \tag{5.2}$$

$$z_d = 0.5 + 0.1\sin(0.4t). \tag{5.3}$$

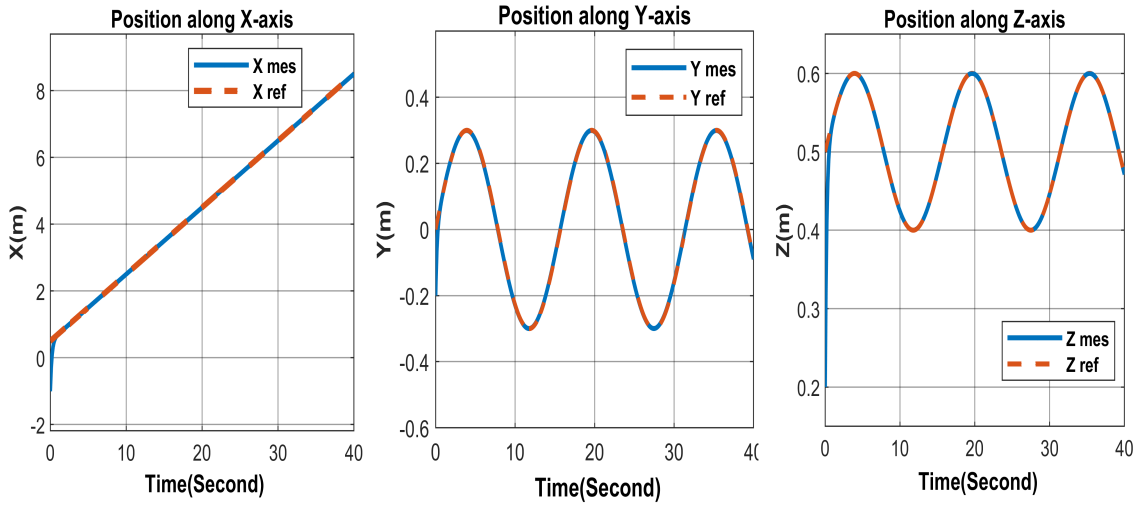



Figure 5.2: 3D Sinusoidal Trajectory Position Tracking Performance

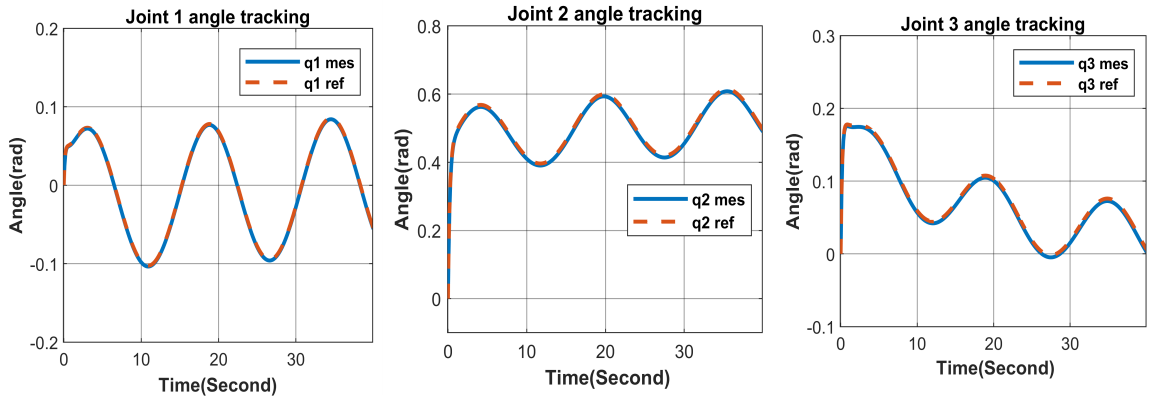



Figure 5.3: 3D Sinusoidal Trajectory Joint Angle Tracking Performance

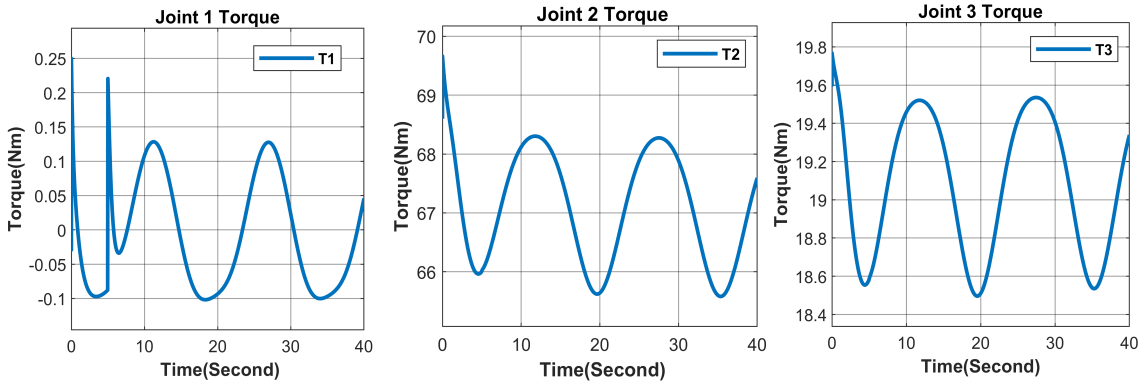



Figure 5.4: Control Input to Robot Arm Joints

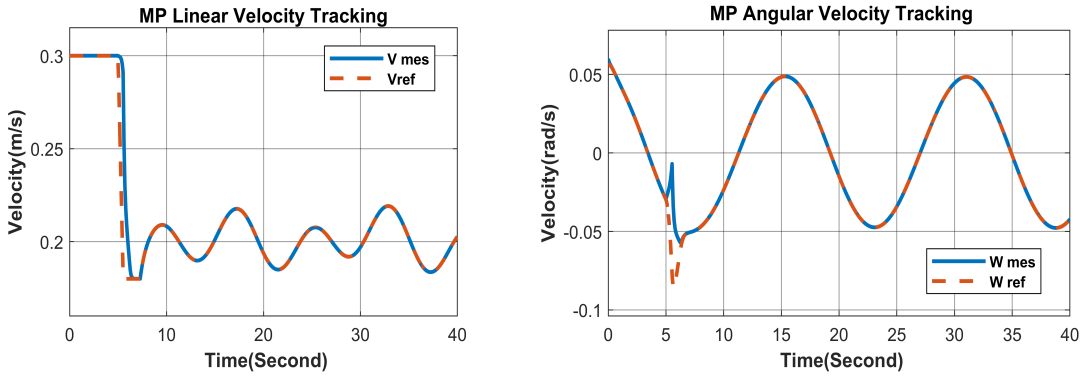



Figure 5.5: MP Linear and Angular Velocity tracking performance

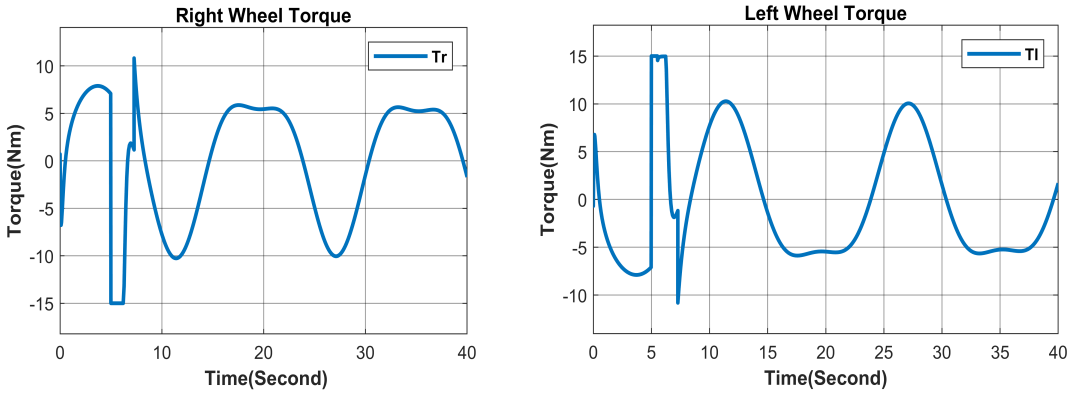



Figure 5.6: MP Control Input to Mobile Robot Wheels

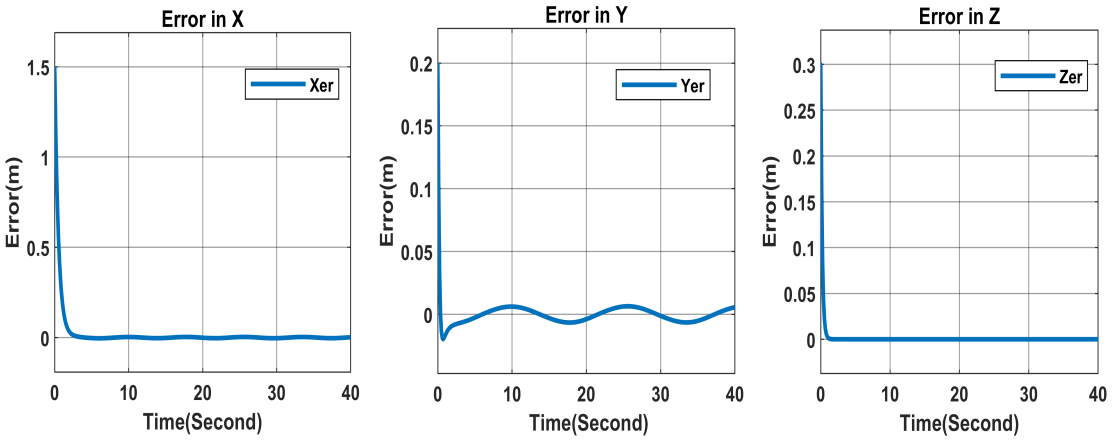



Figure 5.7: Trajectory Tracking Error in X,Y and Z

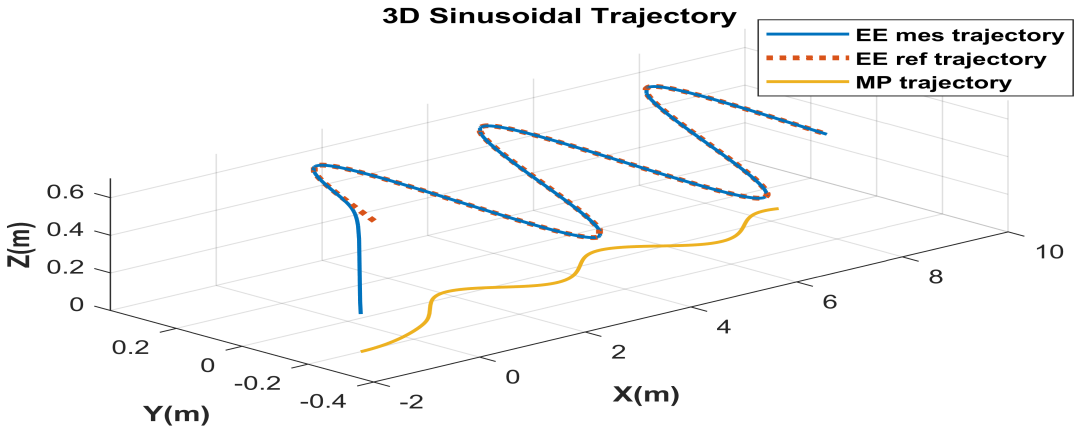



Figure 5.8: 3D Sinusoidal Trajectory Tracking

Sinusoidal trajectory is one of the most complicated trajectory that has been used to test the capability of the controller. Figure 5.2 shows that the mobile manipulator end effector position very smoothly converges to its target configuration. Despite the mobile manipulator undergoing significant attitudinal changes during the trajectory, the controller successfully maintains tracking. Furthermore, it is evident that tracking is maintained using relatively smooth control energy, as shown in Figure 5.6.

## 5.4 3D Spiral Trajectory Tracking

Spiral trajectories provide a continuous and smooth path, allowing for comprehensive testing of the controller's ability to handle continuous motion. The continuous and dynamic nature of the spiral trajectories allows for testing the controller's ability to respond to changes in velocity, acceleration, and direction in real-time. Therefore, in this section, the performance of the controllers for this trajectory was examined, and the results were presented in the form of graph plots shown in Figures(5.9)-(5.15). The $X, Y$ and $Z$ coordinates of the end effector's trajectory during the test are provided in Figure 5.9. Figure 5.10 shows the three robot arm joint angles used to achieve the desired trajectory. The desired 3D Spiral trajectory and the Mobile Manipulator tracking response is shown in Figure 5.15. These plots show that through mobile platform and arm cooperation the tracking was achieved. The Spiral reference trajectory

which is given to the manipulator arm is obtained by

$$x_d = 0.1t\sin(0.25t) \tag{5.4}$$

$$y_d = 0.1t\cos(0.25t) \tag{5.5}$$

$$z_d = 0.3 + 0.008t. \tag{5.6}$$

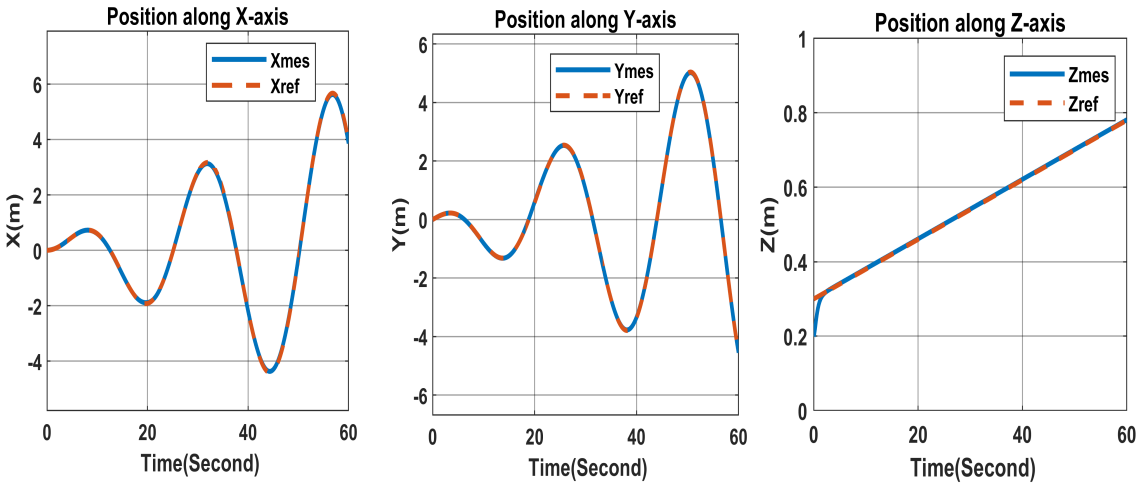



Figure 5.9: 3D Spiral Trajectory Position Tracking Performance

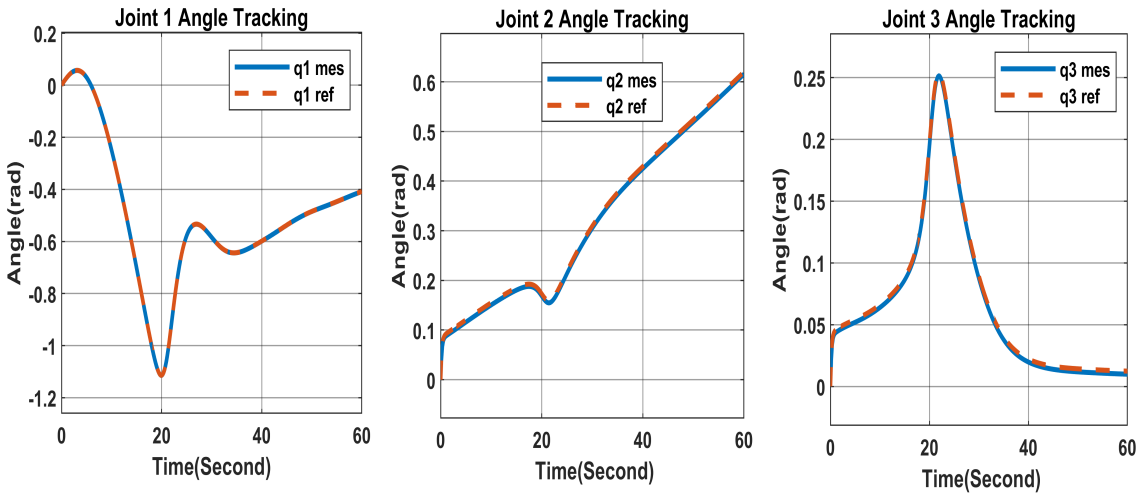



Figure 5.10: 3D Spiral Trajectory Joint Angle Tracking Performance

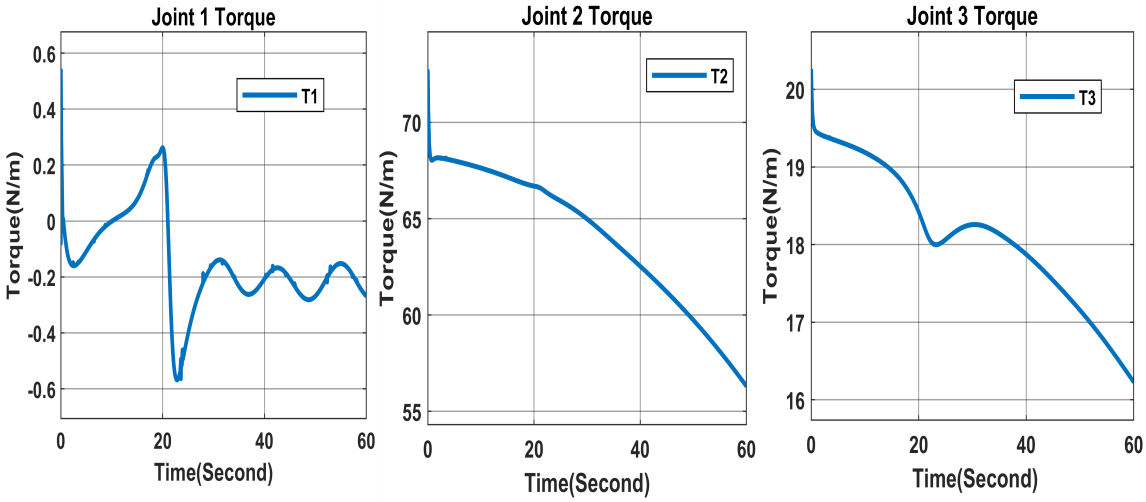



Figure 5.11: Control Input to Robot Arm Joints

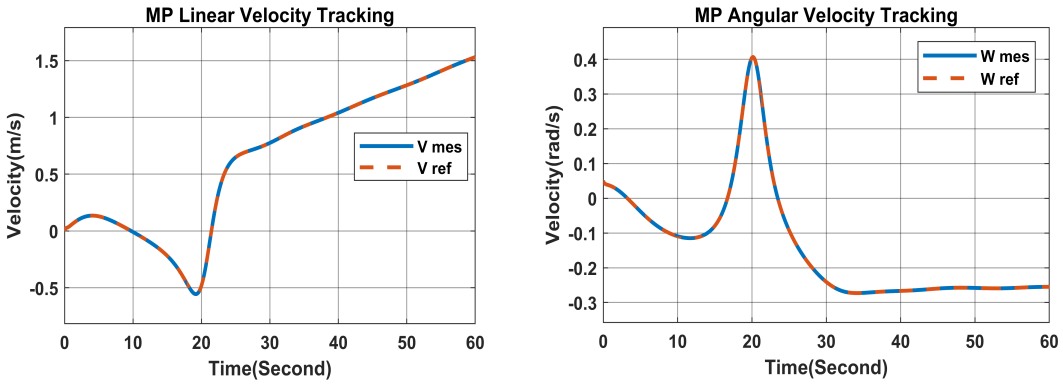



Figure 5.12: MP Linear and Angular Velocity tracking performance

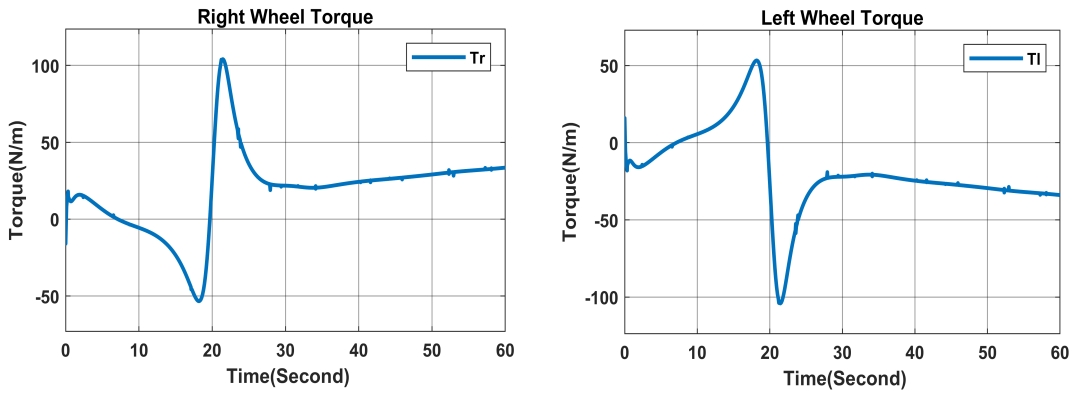



Figure 5.13: Control Input to Mobile Robot Wheels

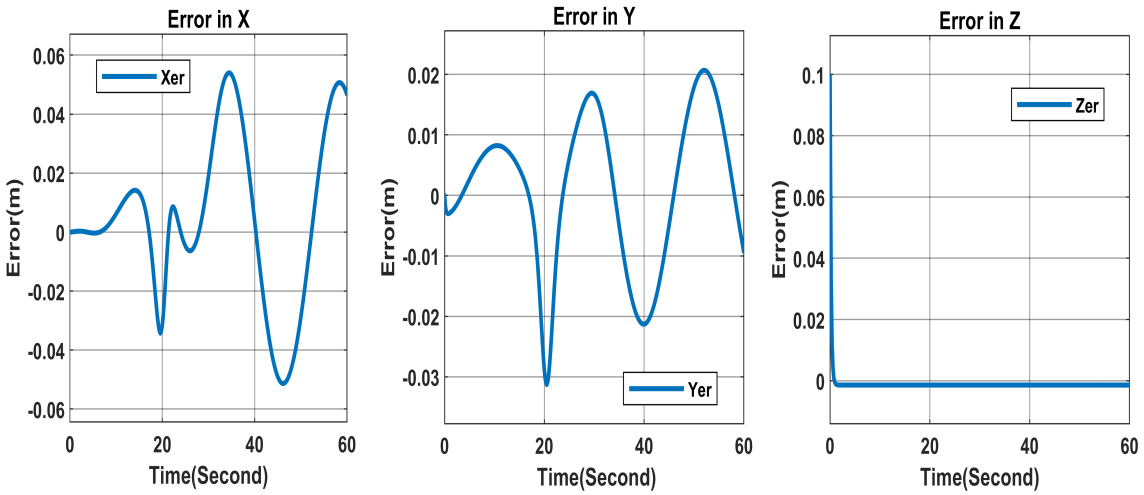



Figure 5.14: Trajectory Tracking Error

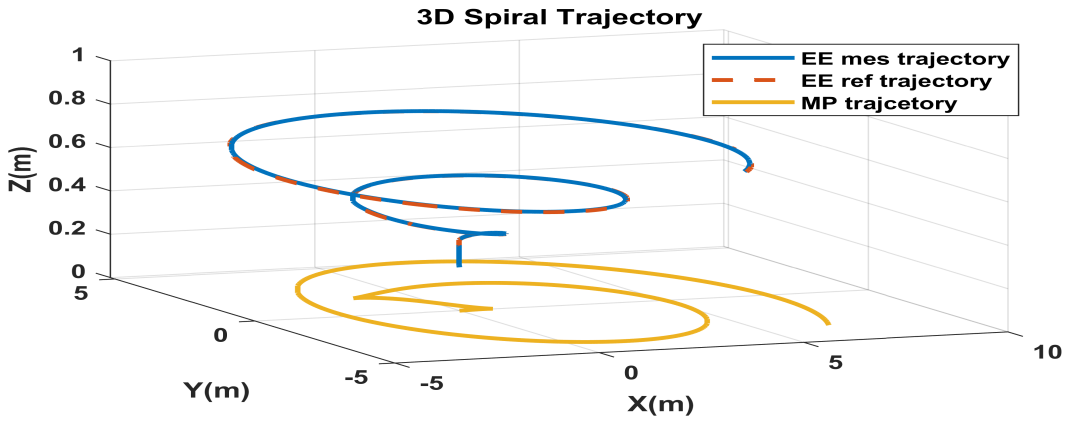



Figure 5.15: 3D Plot for Spiral Trajectory Tracking

Mobile Manipulator end effector translational position is illustrated in Figure 5.9, demonstrating how the developed control scheme ensures that the MM end effector accurately follows the desired trajectory. The attained linear and angular velocities of the mobile manipulator base are also shown to track the velocity configuration derived from the reference trajectory by the velocity controller. The position error is also seen to be smoothly converge to zero as seen in Figure 5.14 with the smooth control inputs shown in Figure 5.11 and Figure 5.13.

## 5.5 Square Trajectory Tracking

The final trajectory tracking testing scenario is the smoothed square trajectory which has continuous smooth motion and can be more representative of real-world scenarios. This helps assess how well the controller handles dynamic and continuous paths. Additionally, a pseudo code for the square wave trajectory was generated and putted in Appendix B. The smoothing of square trajectory is required in order to improve the control performance in accuracy, energy efficiency and improved stability by avoiding sudden changes in the trajectory.

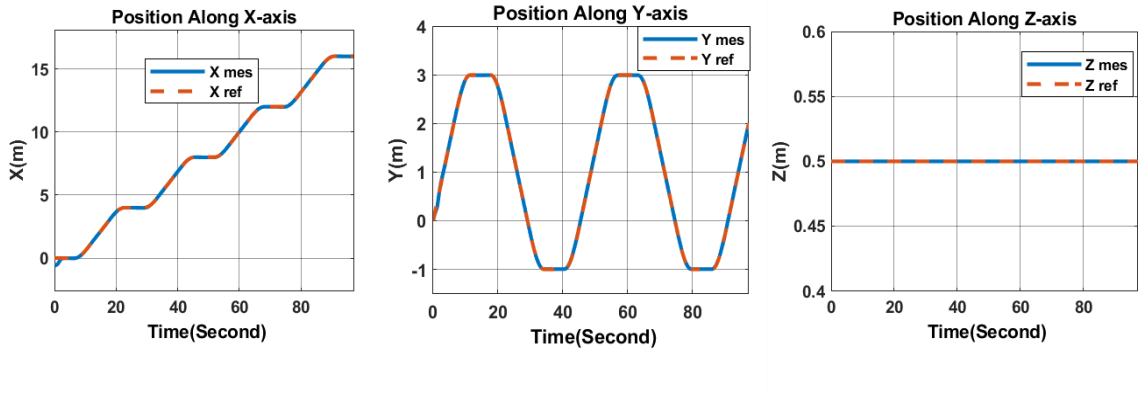



Figure 5.16: Square Trajectory Position Tracking Performance

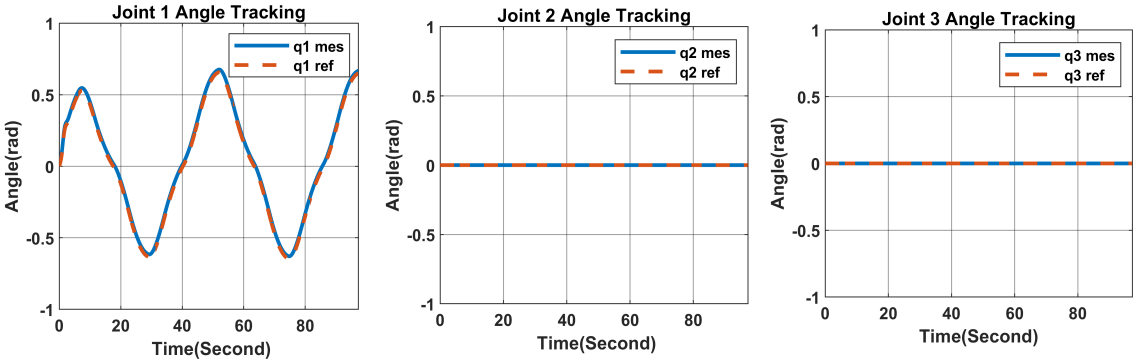



Figure 5.17: Square Trajectory Joint Angle Tracking Performance

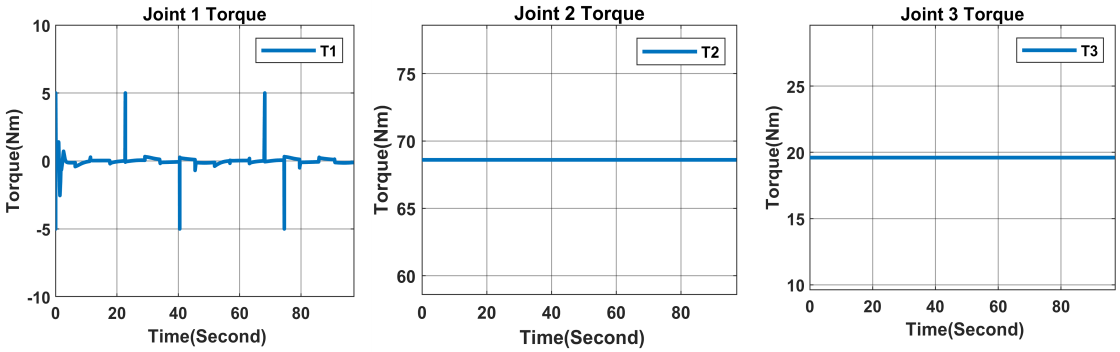



Figure 5.18: Control Input to Robot Arm Joints

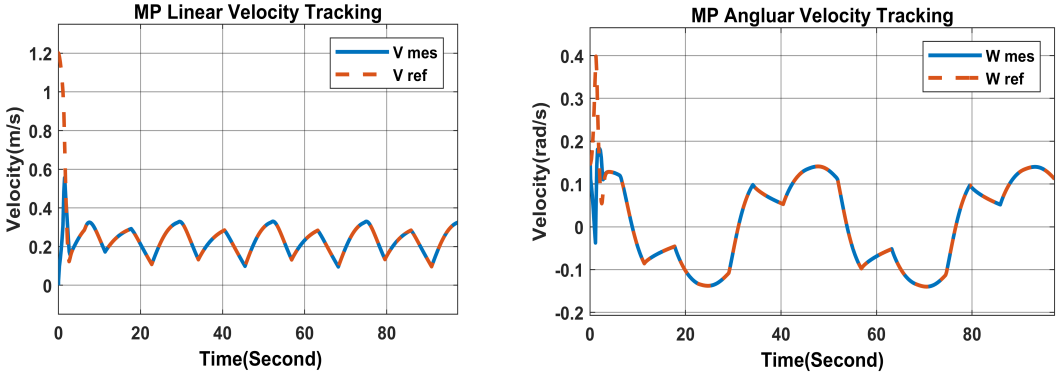



Figure 5.19: MP Linear and Angular Velocity tracking performance

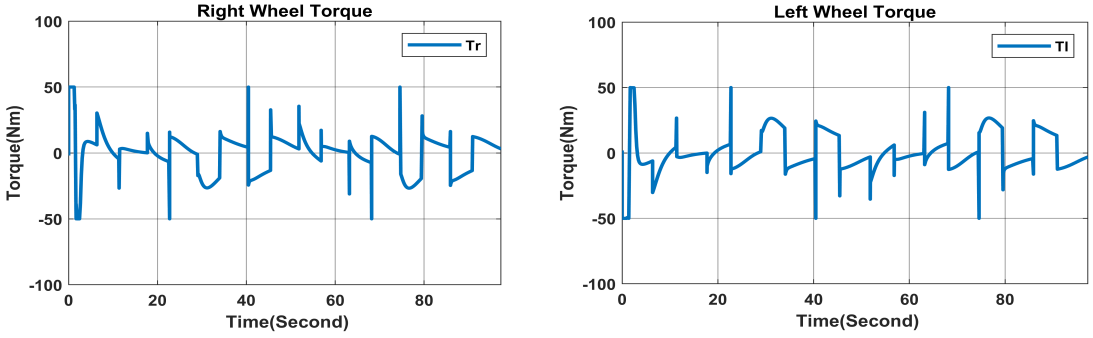



Figure 5.20: Control Input to Mobile Robot Wheels

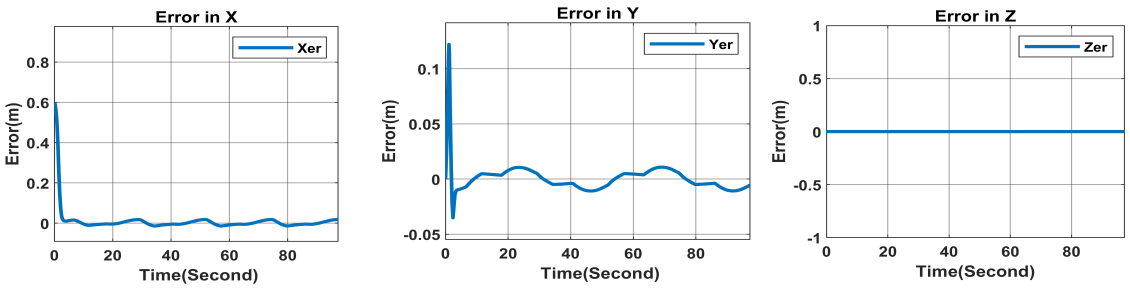



Figure 5.21: Trajectory Tracking Error in X,Y and Z

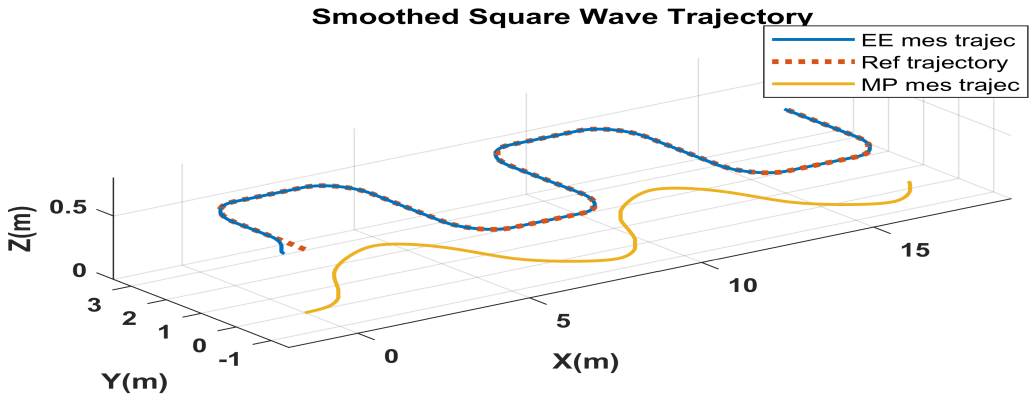



Figure 5.22: 3D Plot for Square Trajectory Tracking

The navigation coordinates of the mobile manipulator arm are depicted in Figure 5.16, where they can be seen tracking the given square reference trajectory. The attained linear and angular velocity of the mobile manipulator base is also seen in Figure 5.19. Tracking is ensured given the generated control input presenteed in Figure 5.18 and 5.20.

## 5.6 Disturbance Rejection Capability

In this subsection the capability of the proposed Backstepping Fuzzy Sliding Mode Control to reject disturbances and overcome parametric uncertainties is assessed. The system (Mobile Manipulator) subjected with a disturbance torque to one of it's joints and the effect of the applied disturbance on the trajectory tracking of the Mobile Manipulator and how the the controller tries to reject the applied disturbance for a Spiral trajectory and for step reference input in $X, Y$ and $Z$ axis is assessed, and the results clearly demonstrate the robustness of the

proposed controller.

### 5.6.1 Spiral Trajectory Tracking with Disturbance

In this section Spiral trajectory tracking of the proposed controller is further investigated by applying a disturbance torque to joint focuses on the study of sinusoidal trajectory tracking Mode Control with an added input disturbance force. The effect of the disturbance torque and the counter action of the controller was analyzed using graph plots, with Figure (6.29) showing the added input disturbance torque.

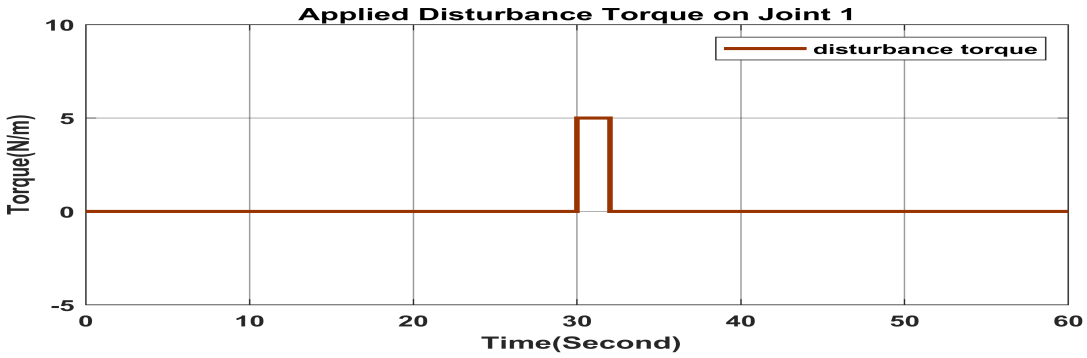



Figure 5.23: The Input Disturbance Torque Added on Joint 1

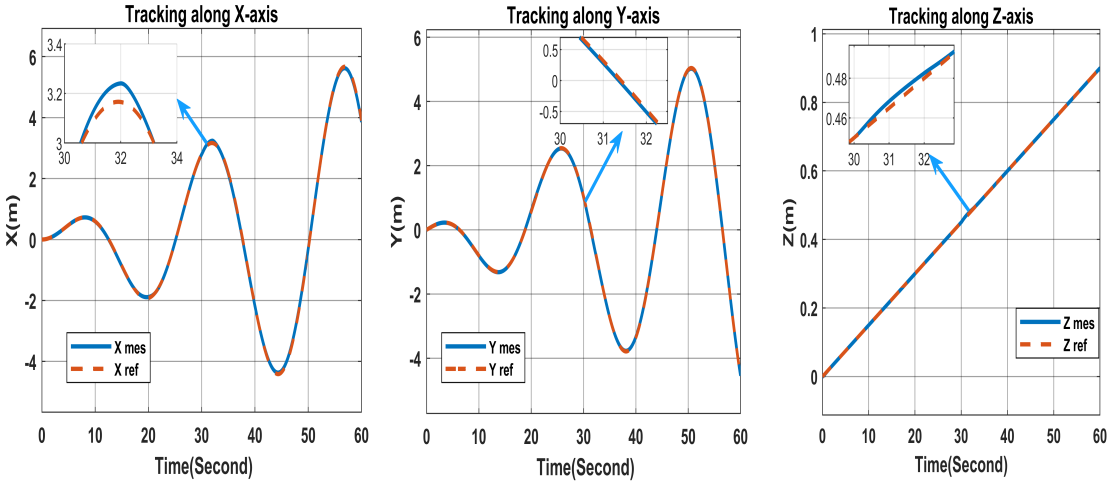



Figure 5.24: Position Tracking Response for Spiral Trajectory with Added Input Disturbance

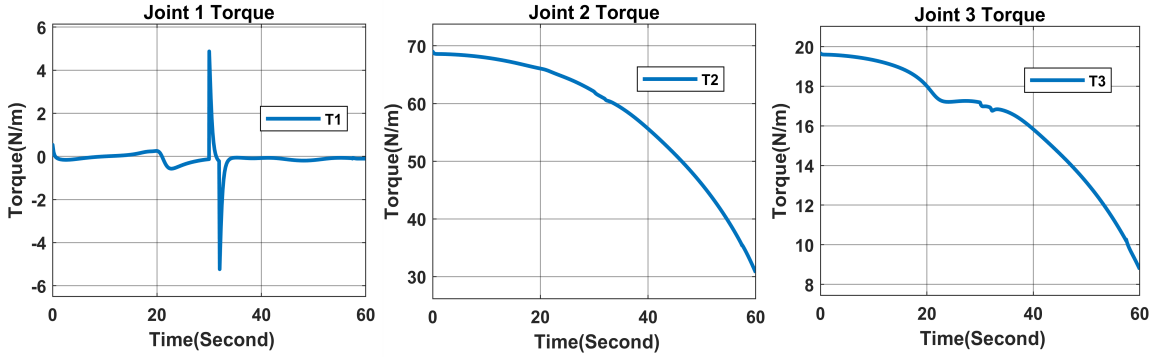



Figure 5.25: Control Input to Robot Joints with Added Disturbance

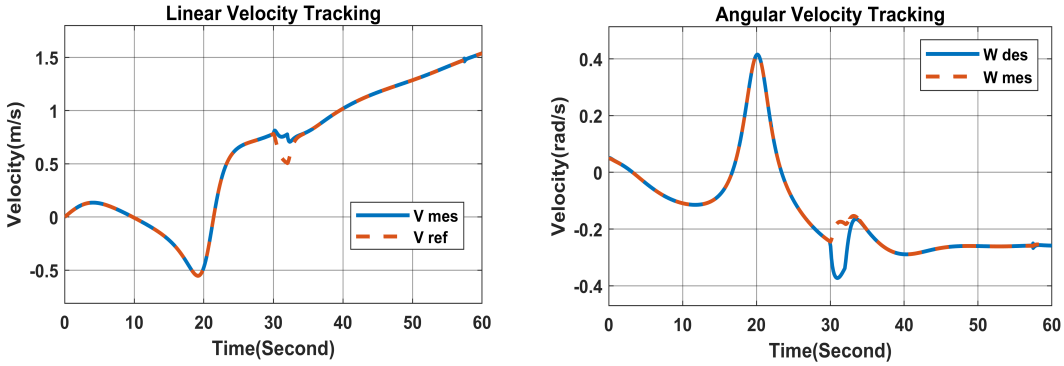



Figure 5.26: Velocity Tracking of MP with Applied Disturbance

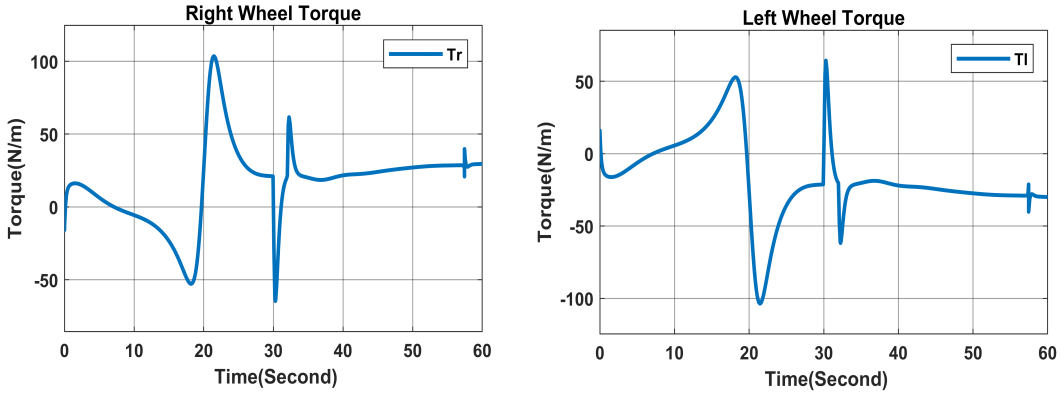



Figure 5.27: Control Input to MP with Added Input Disturbance

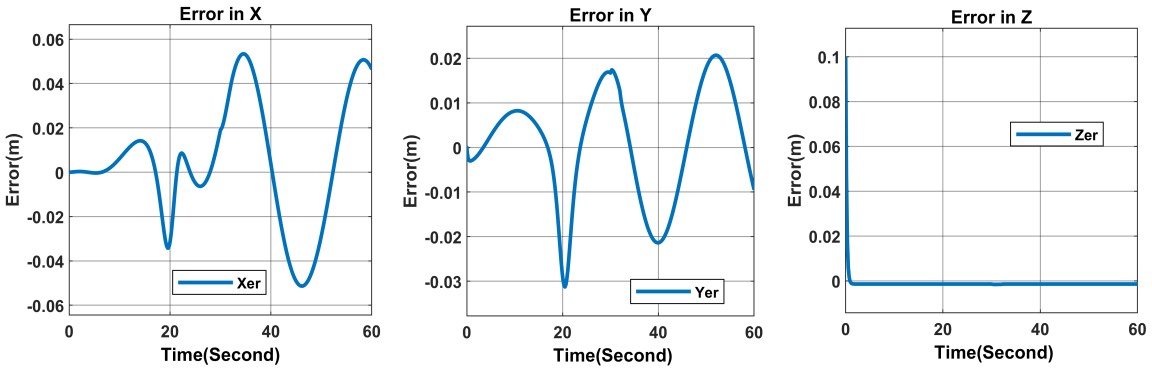



Figure 5.28: Trajectory Tracking Error with Added Input Disturbance

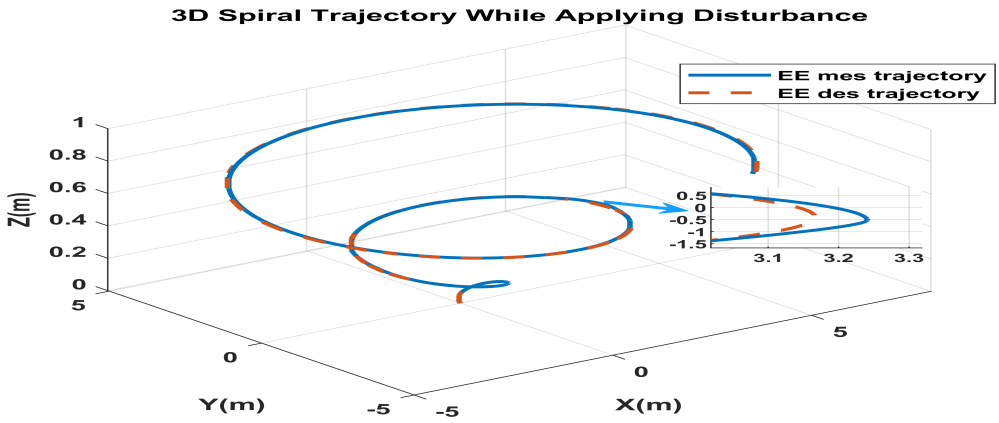



Figure 5.29: 3D Plot for Spiral Trajectory Tracking with Added Input Disturbance

Spiral trajectory is generated to incorporate both linear and rotational components, they are effective in testing the robustness of the controller against various disturbances and inaccuracies in both movement types, which has shown in the above Figures clearly i.e Figure 5.24-5.29.

### 5.6.2 Step Response with Disturbance

In order to analyze the effect of disturbance for step trajectory in $X, Y$ and $Z$ a disturbance torque is applied on joint 3.

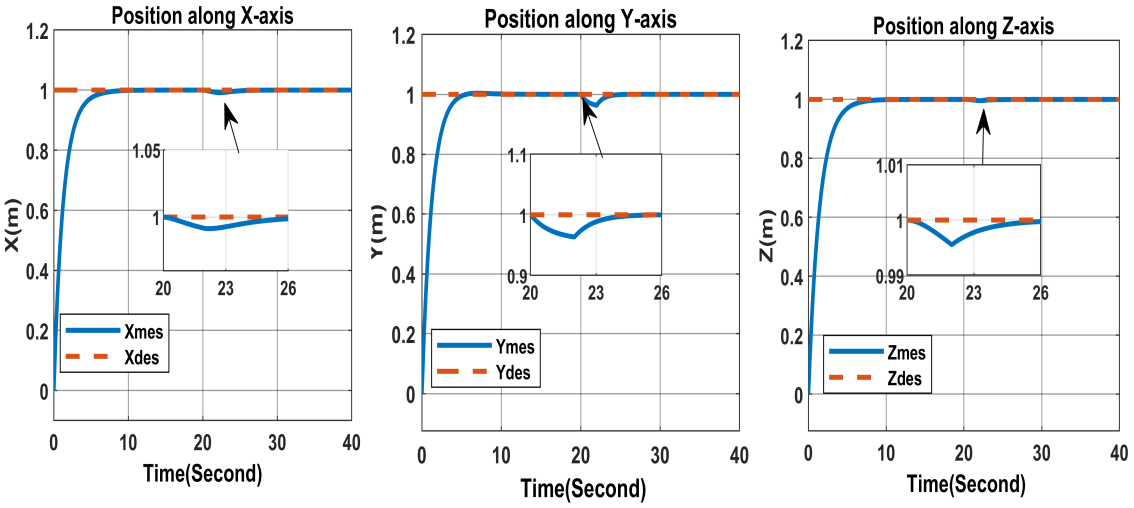



Figure 5.30: Step Position Tracking with applied disturbance

## 5.7 Comparison Between BFSMC and BSMC in Terms of Disturbance Rejection on Spiral Trajectory

The integral time absolute error (ITAE) is used as a performance index so as to compare the performance of the Conventional Backstepping Sliding Mode to that of the Backstepping Fuzzy Sliding Mode controller.

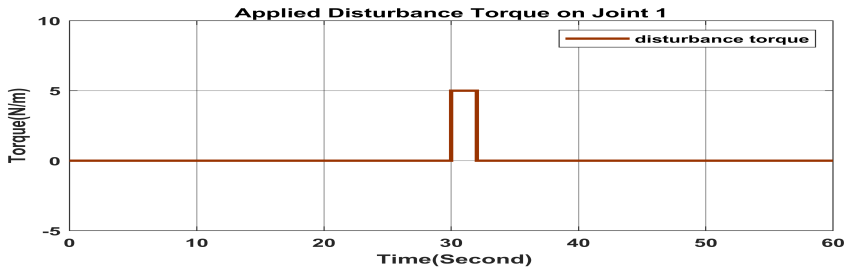



Figure 5.31: Input Disturbance

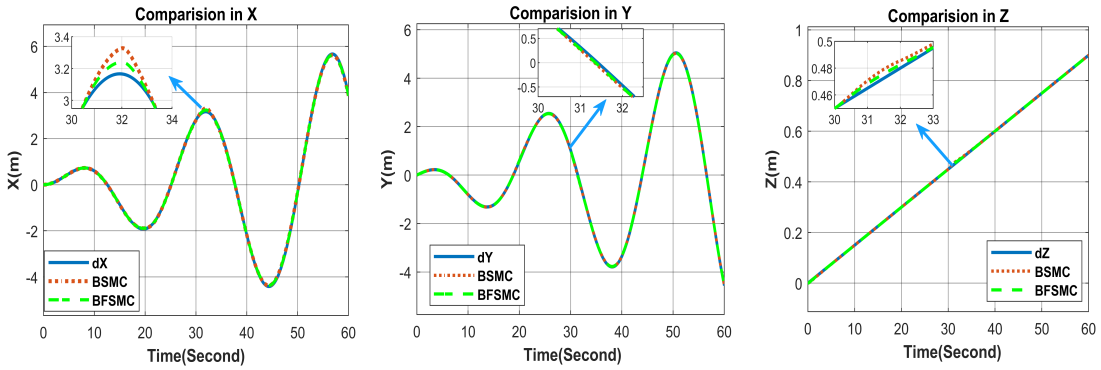



Figure 5.32: Position Tracking with applied disturbance in Spiral Trajectory

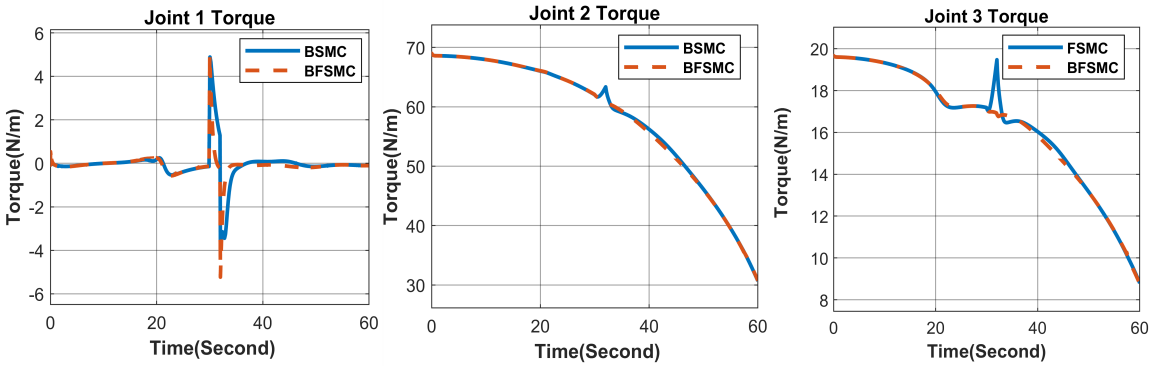



Figure 5.33: BSMC and FBSMC Control Input to Robot Joints

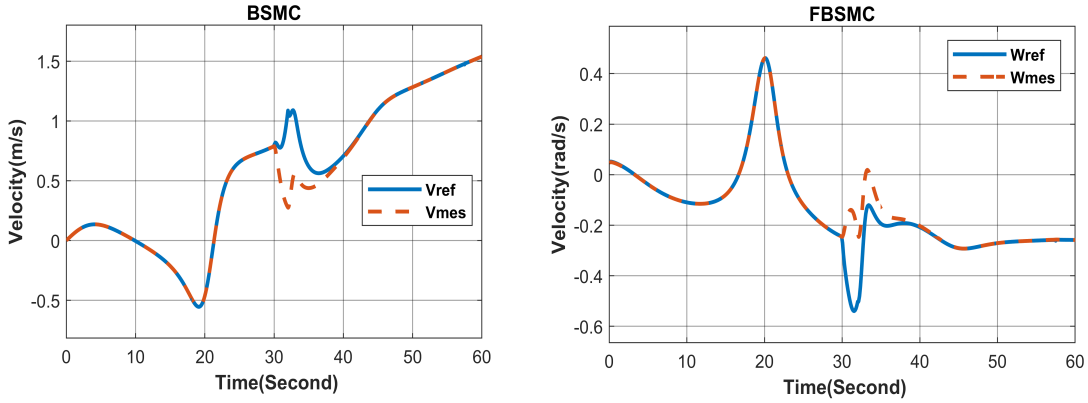



Figure 5.34: BSMC MP Linear and Angular Velocity Tracking Performance With Added Input Disturbance

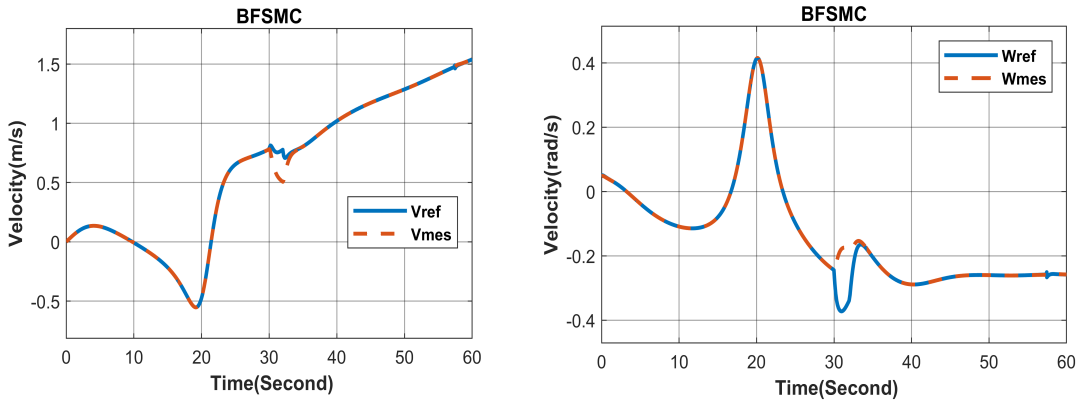



Figure 5.35: FBSMC MP Linear and Angular Velocity Tracking Performance With Added Input Disturbance

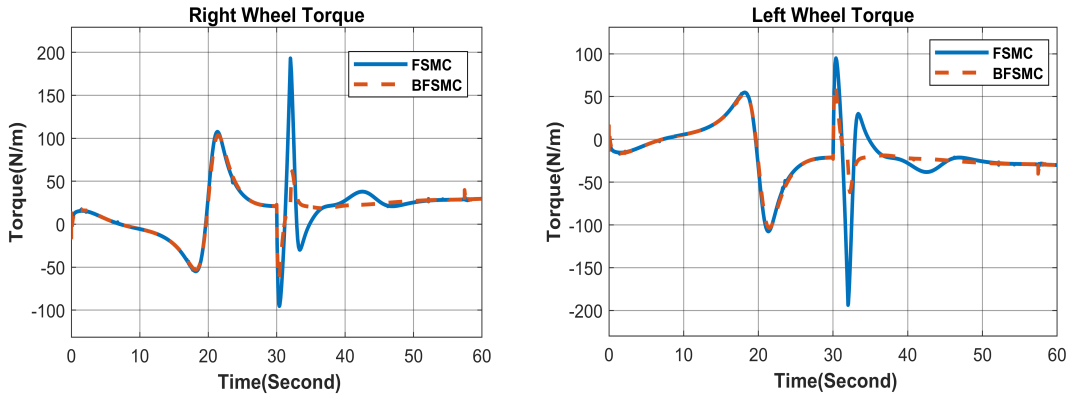



Figure 5.36: Control Input to MP with Added Input Disturbance

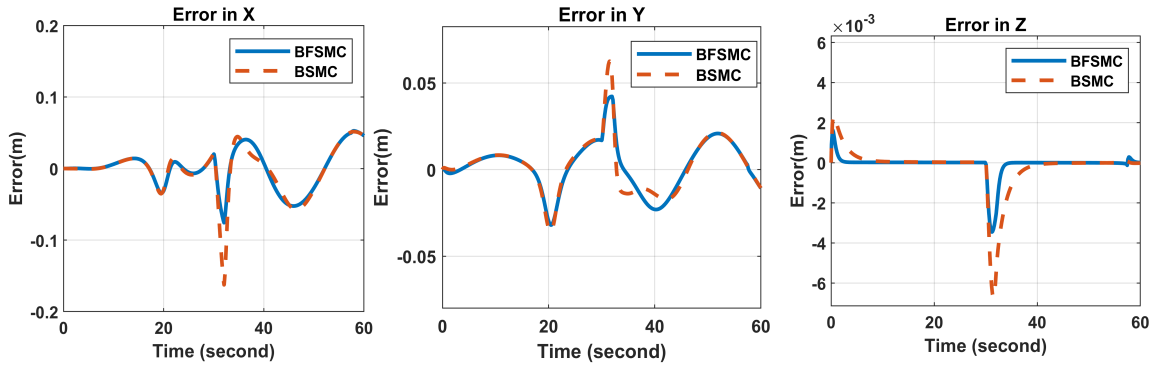



Figure 5.37: Control Input to MP with Added Input Disturbance

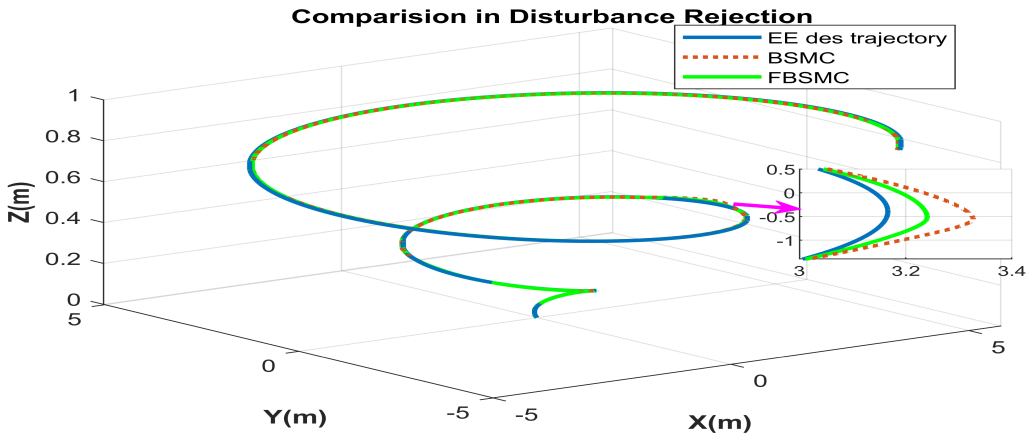



Figure 5.38: 3D Plot for Spiral Trajectory Tracking with added Input Disturbance

# 5.8 Comparison Between BFSMC and BSMC in Terms of Disturbance Rejection + Parameter Variation

In this section along side with the added disturbance torques on the robot joints the mass of the robot also increased by 25% to asses how the proposed controller handles the applied disturbance and parameter variation.

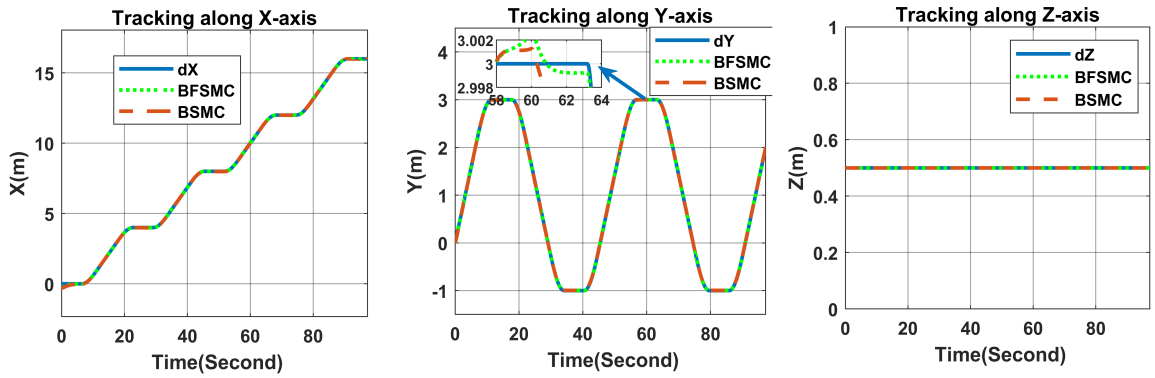



Figure 5.39: Position Tracking with applied disturbance and Parameter Variation in Square Trajectory

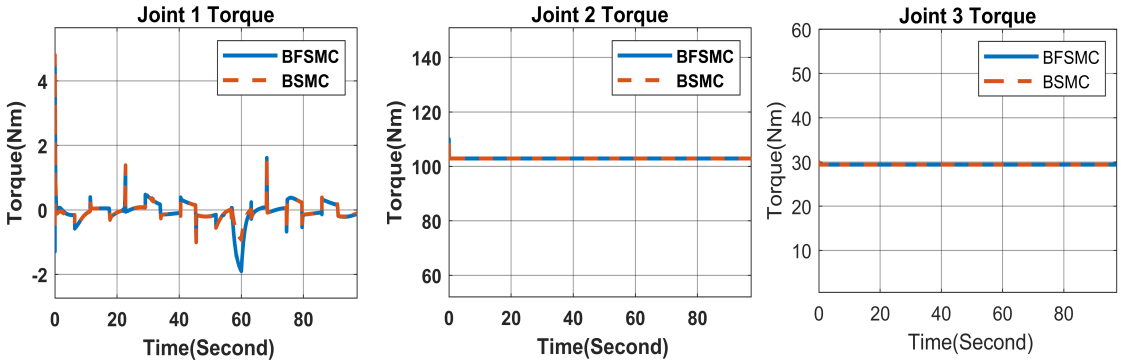



Figure 5.40: BSMC and FBSMC Control Input to Robot Joints

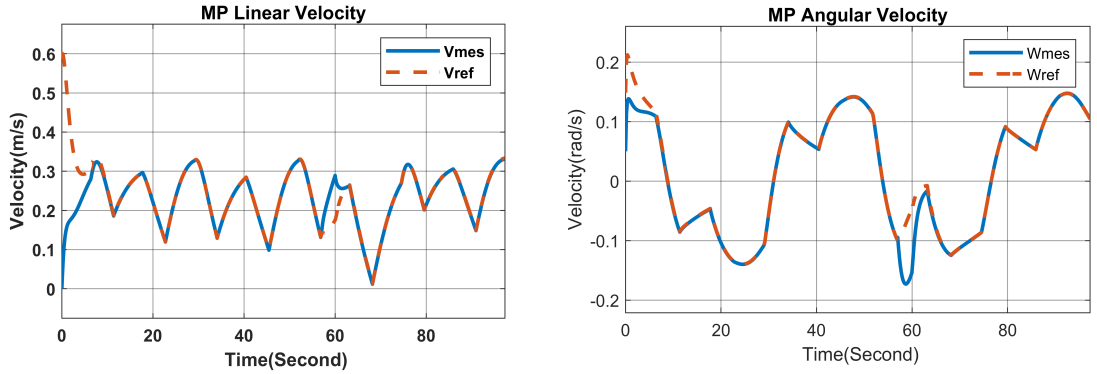



Figure 5.41: BSMC MP Linear and Angular Velocity Tracking Performance With Added Input Disturbance and Parameter varations

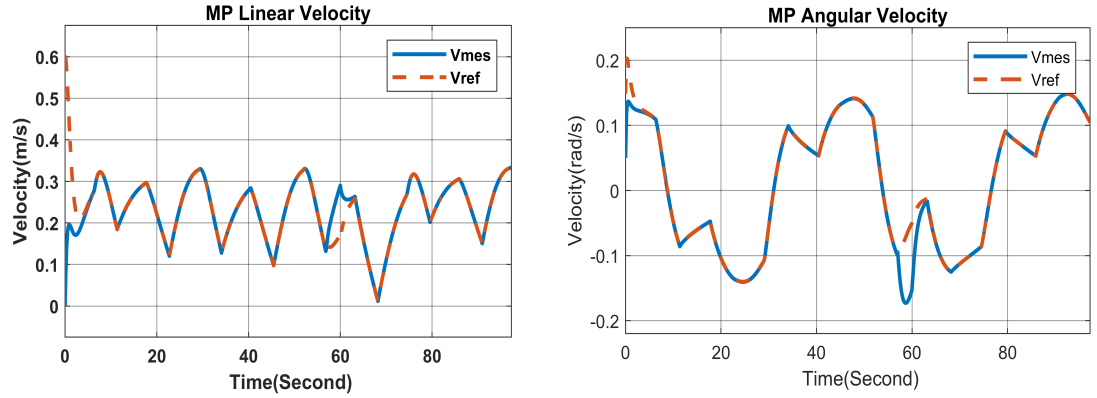



Figure 5.42: BFSMC MP Linear and Angular Velocity Tracking Performance With Added Input Disturbance and Parameter varations

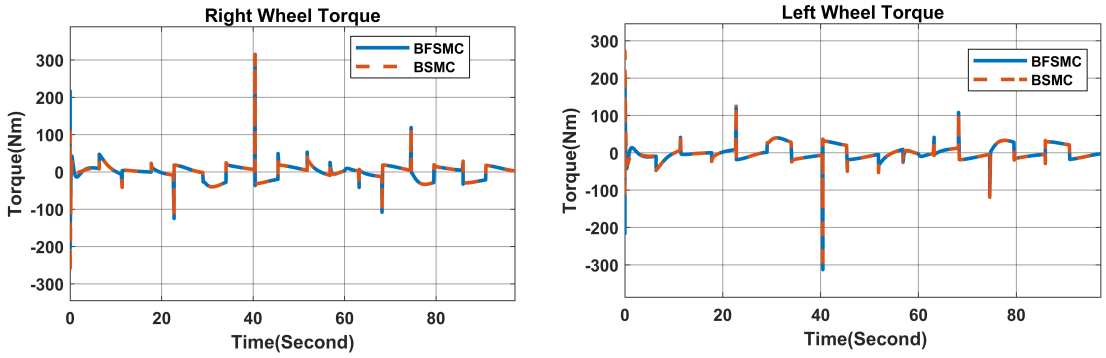



Figure 5.43: Control Input to MP with Added Input Disturbance

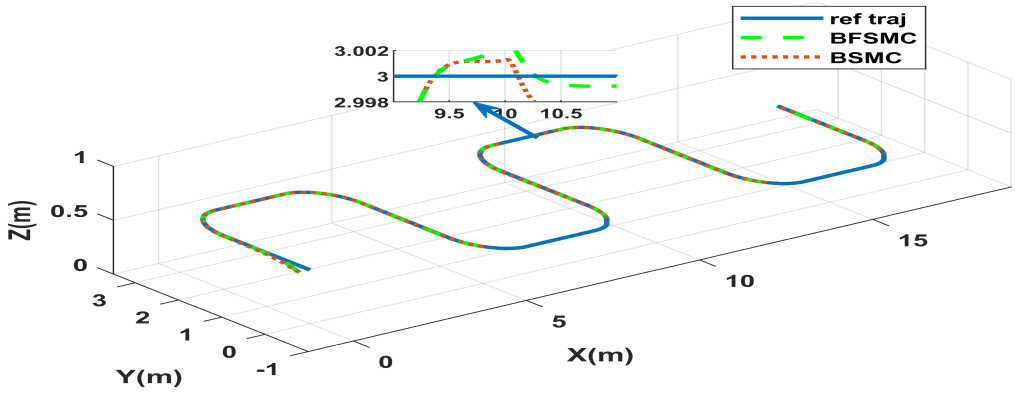



Figure 5.44: 3D Plot for Square Trajectory Tracking with added Input Disturbance and Parameter Variation

## Controllers Performance Comparison for Rectangular Trajectory with input added Disturbance and Parameter Variation

The ITAE criterion places more weight on large errors that persist for a long time. By integrating the product of time and absolute error over the duration of the experiment, it provides a quantitative measure of the overall tracking performance.

In trajectory tracking control experiments, the goal is often to minimize the ITAE value, as it indicates better tracking performance. Controllers can be tuned or designed using optimization techniques to minimize the ITAE criterion, resulting in improved system performance and better trajectory tracking.

To calculate the respective ITAE values and the performance improvement that is shown by Backstepping Fuzzy Sliding Mode control over Backstepping Sliding Mode control the following formulas has been used

$$ITAE = \int_0^{\infty} t|e(t)|dt \tag{5.7}$$

$$PerformanceImprovment(\%) = \frac{BSMC - BFSMC}{BSMC} * 100 \tag{5.8}$$

Table 5.2: Performance Comparsion Between BSMC and BFSMC on Square Trajectory Tracking

| | ***BSMC*** | ***FBSMC*** | ***Improvment*** |
|---|---|---|---|
| X | 3.91 | 2.06 | 47.31% |
| Y | 4.637 | 4.265 | 8.02% |
| Z | 2.431 | 1.232 | 49.32% |

The overall performance improvement in the given square trajectory of BFSMC over BSMC is also calculated after adding each errors in $X, Y$ and $Z$ and inserting the error sum values on the above mentioned performance improvement formula and the BFSMC result improves the result of BSMC ones by 31.16% indicating that the superiority of the BFSMC over BSMC under various operating conditions by resulting better tracking performance.

# Chapter 6

# Conclusion and Recommendation

## 6.1 Conclusion

This thesis addressed the problem of designing of a robust controller capable of tracking a reference trajectory in 3D space for nonholonomic MM in the presence of parametric uncertainties and external disturbances. DH parameters and homegenus transformation matrices are used to obtain the kinematic model of mobile manipulator. Lagrange dynamic methodology is used to drive the bulky and mathematically intensive dynamics of the plant. A redundancy resolution scheme and inverse kinematics to deal with the singularities and the robot arm joints limits was used. This includes the use of the manipulability and Pseudo inverse Jacobian concepts. The purpose of such approach was to avoid online inverse kinematic calculations and the associated singularity problems that have been problematic in the control of MM. The stability of the developed system is checked using Lyapunov functions.

Backstepping controller has been used for velocity controlling and fuzzy sliding mode control also designed and used for controlling dynamics of the system. The designed velocity controller enables trajectory tracking by reducing the error in position and the respective dynamic control by reducing the error in joint angle tracking. The designed BFSMC tested in both disturbed environment and parameter variation.

The trajectory tracking performance of the developed BFSMC is tested by various trajectory scenarios. From the results its conclude that the Backstepping Fuzzy sliding Mode controller of the MM is making the system stable, while enabling effective Mobile Platform and Robot Arm cooperation.

## 6.2 Recommendation for Future Work

- Integrating path planning algorithms and image recognition in controller designing since path planning helps robots avoid obstacles and navigate around dynamic or static obstacles in their environment. This is crucial for ensuring the safety of the robot, preventing collisions, and protecting both the robot and its surroundings. It also finds optimal paths or trajectories for the robot to reach its goal while considering various factors such as distance, time, energy consumption, and other constraints. This leads to more efficient and resource-aware movements.
- Modelling the motor dynamics of the Mobile Manipulator to test the performance of the controller, taking into account the motor's behavior.
- Experimental testing and validation of the proposed controllers.

# References


[1] Mustafa Mashali. *Kinematic control of redundant mobile manipulators*. University of South Florida, 2015.

[2] Mahmoud Mustafa. Modeling and dynamic control of autonomous ground mobile manipulators, 2016.

[3] Yoshio Yamamoto and Xiaoping Yun. Coordinating locomotion and manipulation of a mobile manipulator. In *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, pages 2643–2648. IEEE, 1992.

[4] Sheng Lin and Andrew A Goldenberg. Neural-network control of mobile manipulators. *IEEE Transactions on Neural networks*, 12(5):1121–1133, 2001.

[5] Felix Huber, Konstantin Kondak, Kai Krieger, Dominik Sommer, Marc Schwarzbach, Maximilian Laiacker, Ingo Kossyk, Sven Parusel, Sami Haddadin, and Alin Albu-Schäffer. First analysis and experiments in aerial manipulation using fully actuated redundant robot arm. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 3452–3457. IEEE, 2013.

[6] Yu Zuyao, Lei Jie, Li Weijia, Xiang Zhongxiang, and Wu Jinbo. Design and analysis of a three-dof underwater manipulator. In *Proceedings of 2011 International Conference on Fluid Power and Mechatronics*, pages 237–241. IEEE, 2011.

[7] Mads Hvilshøj, Simon Bøgh, Oluf Skov Nielsen, and Ole Madsen. Autonomous industrial mobile manipulation (aimm): past, present and future. *Industrial Robot: An International Journal*, 39(2):120–135, 2012.

[8] KV Shihabudheen, Nimmi George, KA Chinmaya, and Jose Thankachan. Stability control and trajectory tracking of two wheeled mobile manipulator. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–6. IEEE, 2015.

[9] K.V. Shihabudheen, Nimmi George, K.A. Chinmaya, and Jose Thankachan. Stability control and trajectory tracking of two wheeled mobile manipulator. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–6, 2015.

[10] Waladin Said, Shibly A. Al-Samarraie, and Abdulmohaimen Kassim. Nonlinear controller design for a mobile manipulator trajectory tracking. *IRAQI JOURNAL OF COMPUT-*

*ERS, COMMUNICATION AND CONTROL SYSTEMS ENGINEERING*, 14:30–41, 01 2014.

[11] Nor Mohd Haziq Norsahperi and Kumeresan A Danapalasingam. A comparative study of lqr and integral sliding mode control strategies for position tracking control of robotic manipulators. *International journal of electrical and computer engineering systems*, 10(2.):73–83, 2019.

[12] A. Safo. Trajectory tracking control of mobile manipulator using sliding mode control (smc). 2019.

[13] Bhavik Patel, Ya-Jun Pan, and Usman Ahmad. Adaptive backstepping control approach for the trajectory tracking of mobile manipulators. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1769–1774, 2017.

[14] Manju Rani, Dinanath, and Naveen Kumar. A new hybrid backstepping approach for the position/force control of mobile manipulators. In Manish Prateek, Durgansh Sharma, Rajeev Tiwari, Rashmi Sharma, Kamal Kumar, and Neeraj Kumar, editors, *Next Generation Computing Technologies on Computational Intelligence*, pages 183–198, Singapore, 2019. Springer Singapore.

[15] Mohamed Boukattaya, Neila Mezghani, and Tarak Damak. Computed-torque control of a wheeled mobile manipulator. *International Journal of Robotic Engineering*, 2018.

[16] Rached Dhaouadi and A Abu Hatab. Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework. *Advances in Robotics & Automation*, 2(2):1–7, 2013.

[17] Spyros G Tzafestas. *Introduction to mobile robot control*. Elsevier, 2013.

[18] T.F. Chan and R.V. Dubey. A weighted least-norm solution based scheme for avoiding joint limits for redundant manipulators. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 395–402 vol.3, 1993.

[19] I. Kolmanovsky and N.H. McClamroch. Developments in nonholonomic control problems. *IEEE Control Systems Magazine*, 15(6):20–36, 1995.

[20] Ida-Louise G Borlaug, Kristin Y Pettersen, and Jan Tommy Gravdahl. Combined kinematic and dynamic control of vehicle-manipulator systems. *Mechatronics*, 69:102380, 2020.

[21] Sundarapandian Vaidyanathan and Ahmad Taher Azar. Call for book chapters - backstepping control of nonlinear dynamical systems (elsevier). 08 2018.

[22] B Draženović. The invariance conditions in variable structure systems. *Automatica*, 5(3):287–295, 1969.

[23] K David Young, Vadim I Utkin, and Umit Ozguner. A control engineer's guide to sliding mode control. *IEEE transactions on control systems technology*, 7(3):328–342, 1999.

[24] Wilfrid Perruquetti and Jean Pierre Barbot. *Sliding mode control in engineering*, volume 11. Marcel Dekker New York, 2002.

[25] Christopher Edwards, Enric Fossas Colet, Leonid Fridman, Enric Fossas Colet, and Leonid M Fridman. *Advances in variable structure and sliding mode control*, volume 334. Springer, 2006.

[26] Xinghuo Yu and Jian-Xin Xu. *Variable structure systems: towards the 21st century*, volume 274. Springer Science & Business Media, 2002.

[27] Hebertt Sira-Ramirez. Differential geometric methods in variable-structure control. *International Journal of Control*, 48(4):1359–1390, 1988.

[28] Christopher Edwards and Sarah Spurgeon. *Sliding mode control: theory and applications*. Crc Press, 1998.

[29] Vadim Ivanovich Utkin. Sliding modes and their applications in variable structure systems. *Mir, Moscow*, 1978.

[30] Hak-Keung Lam. Polynomial fuzzy model-based control systems. *Stability Analysis and Control Synthesis Using Membership Function-Dependent Techniques. Cham: Springer-Verlag*, 307, 2016.

[31] Quan Liu, Dong Liu, Wei Meng, Zude Zhou, and Qingsong Ai. Fuzzy sliding mode control of a multi-dof parallel robot in rehabilitation environment. *International Journal of Humanoid Robotics*, 11(01):1450004, 2014.

[32] ABDEL-RAZZAK MERHEB. *NONLINEAR CONTROL ALGORITHMS APPLIED TO 3 DOF PUMA MANIPULATOR*. PhD thesis, MIDDLE EAST TECHNICAL UNIVERSITY, 2008.

[33] Adrian Emanoil Serbencu, Adriana Serbencu, Daniela Cristina Cernega, and M Viorel. Particle swarm optimization for the sliding mode controller parameters. In *Proceedings of the 29th Chinese Control Conference*, pages 1859–1864. IEEE, 2010.

[34] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[35] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 69–73, 1998.

[36] Zwe-Lee Gaing. A particle swarm optimization approach for optimum design of pid controller in avr system. *IEEE Transactions on Energy Conversion*, 19(2):384–391, 2004.

[37] Chih-Cheng Kao, Chin-Wen Chuang, and Rong-Fong Fung. The self-tuning pid control in a slider–crank mechanism system by applying particle swarm optimization approach. *Mechatronics*, 16(8):513–522, 2006.

[38] Nada El Gmili, Mostafa Mjahed, Abdeljalil El Kari, and Hassan Ayad. Particle swarm optimization based proportional-derivative parameters for unmanned tilt-rotor flight control and trajectory tracking. *Automatika*, 61:189 – 206, 2020.

[39] Nada El Gmili, Mostafa Mjahed, Abdeljalil Elkari, and Hassan Ayad. Particle swarm optimization and cuckoo search- based approaches for quadrotor control and trajectory tracking. *Applied Sciences*, 9:1–25, 04 2019.

[40] Belsty Derseh, Lebsework Negash, and Chala Merga. Robust pso tuned fosmc for altitude stabilization and trajectory tracking of agricultural monitoring uav.

# Appendix A

# MATLAB Simulink Model

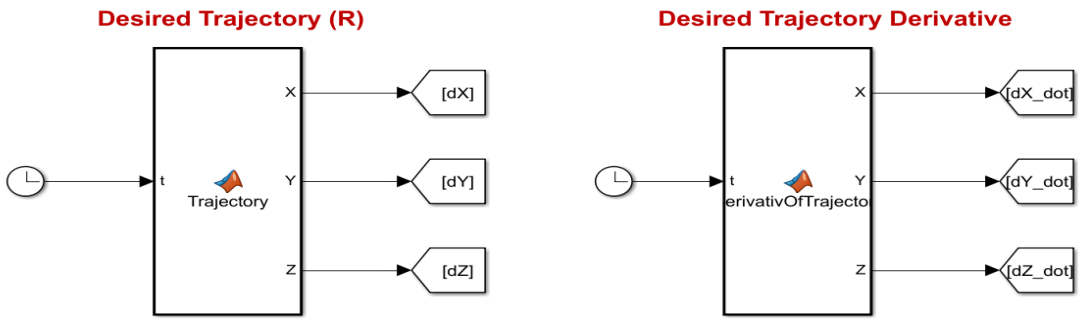



Figure A.1: Desired Trajectory Matlab Function

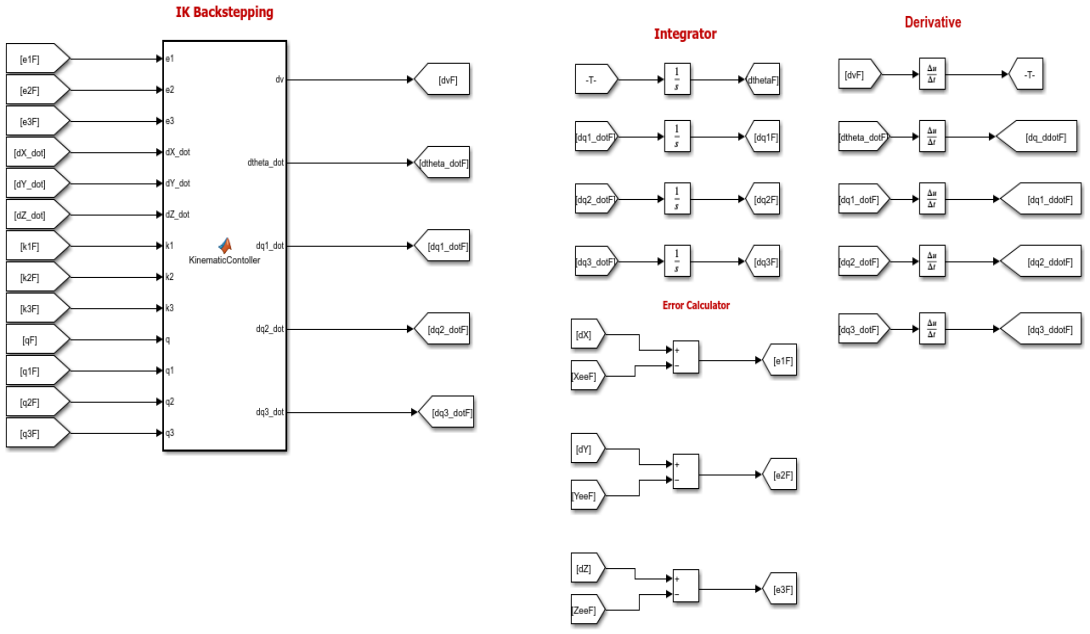



Figure A.2: Simulink Block of Backstepping controller design

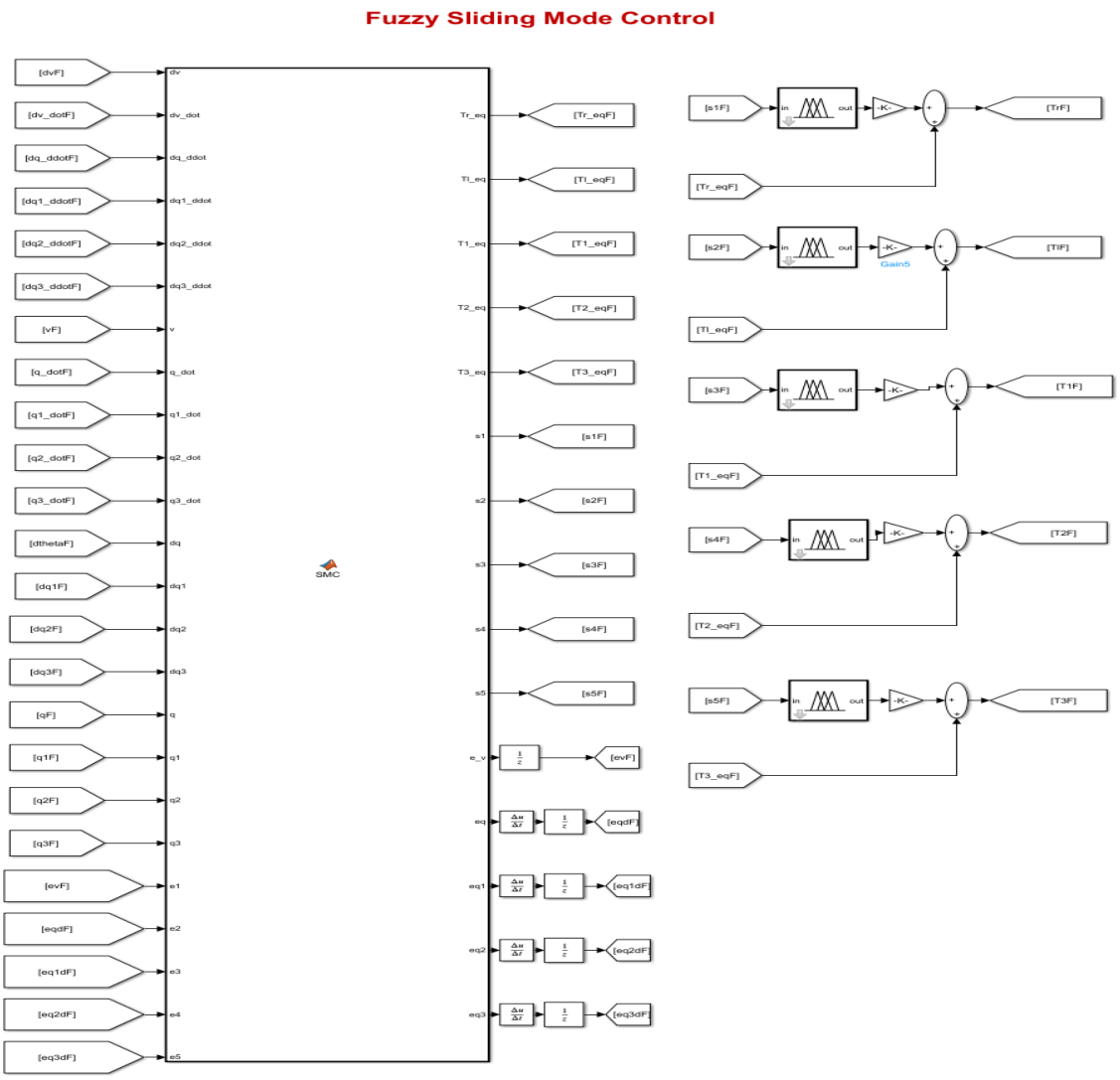



Figure A.3: Simulink Block of Fuzzy Sliding Mode Control Desing

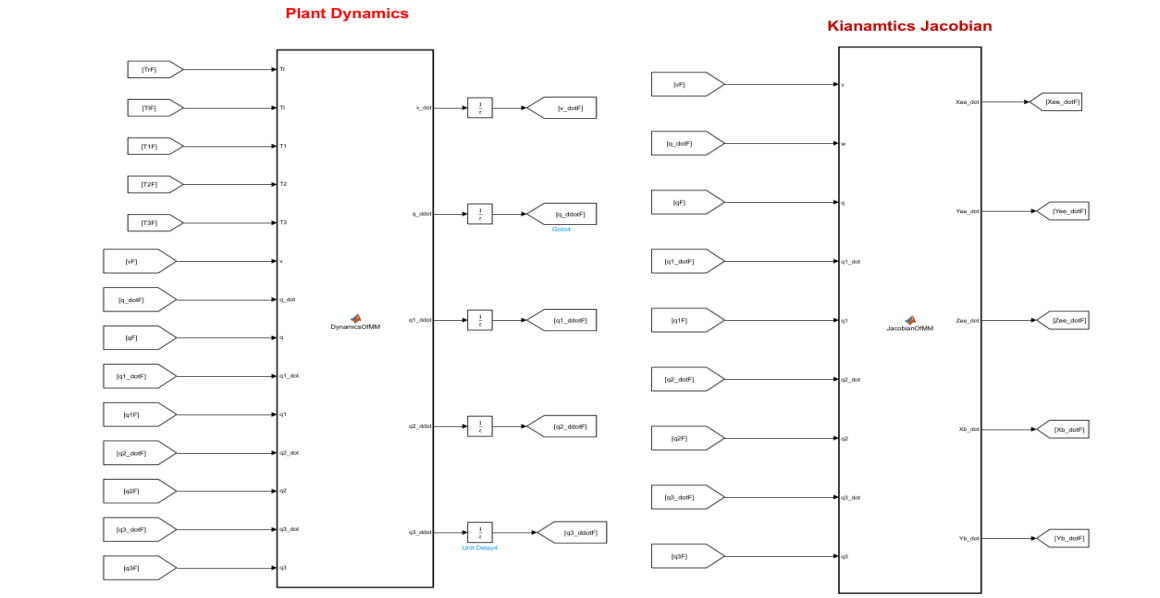



Figure A.4: Simulink model of the plant both kinematics and dynamics

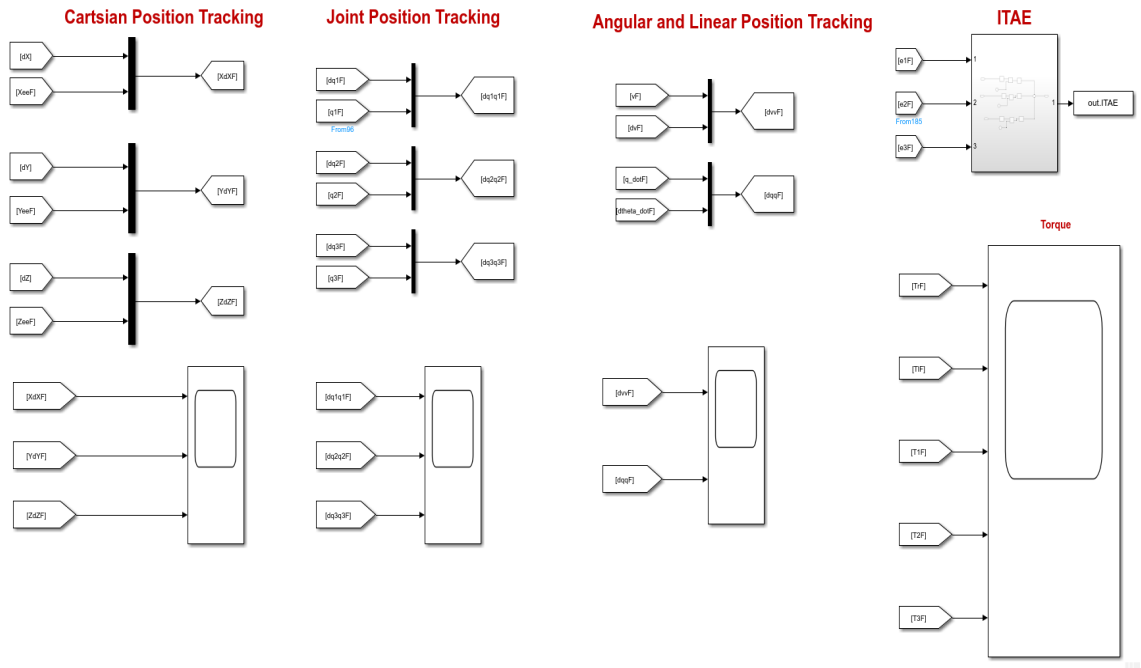



Figure A.5: Scope

# Appendix B

# Desired Square Trajectory Pseudo Code

```
function [X, Y, Z] = square_trajectory(t)
    % Time period of the square wave (in seconds)




      if t <=6.36618
        X = 0;
        Y=0.31416*t;
        Z=0.5;

    elseif t>6.36618 && t<=11.36618
        X = 1-1*cos(0.31416*(t-6.36618));
        Y=2+1*sin(0.31416*(t-6.36618));
        Z=0.5;

    elseif t>11.36618 && t<=17.73236
        X = 0.31416*t-2.5708;
        Y=3;
        Z=0.5;

    elseif t>17.73236 && t<=22.73236
        X=3+1*sin(0.31416*(t-17.73236));
        Y=2+1*cos(0.31416*(t-17.73236));
        Z=0.5;
```

```
  elseif t>22.73236 && t<=29.09855
      X=4;
      Y=9.1416-0.31416*t;
      Z=0.5;

  elseif t>29.09855 && t<=34.09855
      X=5-1*cos(0.31416*(t-29.09855));
      Y=-1*sin(0.31416*(t-29.09855));
      Z=0.5;

elseif t> 34.09855 && t<=40.46447
      X=0.31416*t-5.7124;
      Y=-1;
      Z=0.5;

   elseif t>40.46447 && t<=45.46447
      X=7+1*sin(0.31416*(t-40.46447));
      Y=-1*cos(0.31416*(t-40.46447));
      Z=0.5;
 elseif t>45.46447 && t<=51.83065
      X=8;
      Y=0.31416*t-14.28312;
      Z=0.5;

 elseif t>51.83065 && t<=56.83065
       X=9-1*cos(0.31416*(t-51.83065));
      Y=2+1*sin(0.31416*(t-51.83065));
      Z=0.5;

  elseif t>56.83065 && t<=63.19683
      X=0.31416*t-8.85391;
      Y=3;
      Z=0.5;
```

```
 elseif t>63.19683 && t<=68.19683
      X=11+1*sin(0.31416*(t-63.19683));
      Y=2+1*cos(0.31416*(t-63.19683));
      Z=0.5;

elseif t>68.19683 && t<=74.56301
      X=12;
      Y=23.42472-0.31416*t;
      Z=0.5;

 elseif t>74.56301 && t<=79.56301
      X=13-1*cos(0.31416*(t-74.56301));
      Y=-1*sin(0.31416*(t-74.56301));
      Z=0.5;

 elseif t>79.56301 && t<=85.92919
      X=0.31416*t-11.99552;
      Y=-1;
      Z=0.5;

elseif t>85.92919 && t<=90.92919
      X=15+1*sin(0.31416*(t-85.92919));
      Y=-1*cos(0.31416*(t-85.92919));
      Z=0.5;

     else
      X=16;
      Y=0.31416*t-28.56631;
      Z=0.5;
  end
```

# Appendix C

# Fuzzy .fis Code

```
    [System]
Name='FSMC'
Type='mamdani'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=8
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='input1'
Range=[-1.4 1.4]
NumMFs=8
MF1='NB':'trapmf',[-4.6 -1.4 -0.9667 -0.7003]
MF2='NM':'trimf',[-1 -0.667 -0.25]
MF3='NS':'trimf',[-0.667 -0.25 0]
MF4='ANZ':'trimf',[-0.25 0 0]
MF5='APZ':'trimf',[0 0 0.25]
MF6='PS':'trimf',[0 0.25 0.667]
MF7='PM':'trimf',[0.25 0.667 1]
MF8='PB':'trapmf',[0.7003 0.9667 1.4 4.6]

[Output1]
Name='output1'
Range=[-1 1]
NumMFs=7
```

```
MF1='NB':'trimf',[-1.333 -1 -0.6666]
MF2='NM':'trimf',[-1 -0.6666 -0.3334]
MF3='NS':'trimf',[-0.6666 -0.3334 0]
MF4='AZ':'trimf',[-0.3334 0 0.3334]
MF5='PS':'trimf',[0 0.3334 0.6666]
MF6='PM':'trimf',[0.3334 0.6666 1]
MF7='PB':'trimf',[0.6666 1 1.334]

[Rules]
1, 5 (1) : 1
2, 6 (1) : 1
3, 7 (1) : 1
4, 7 (1) : 1
5, 1 (1) : 1
6, 1 (1) : 1
7, 2 (1) : 1
8, 3 (1) : 1
```