PRECISE POSITION CONTROL OF MOBILE MANIPULATORS

by

Mehmet Nuri Önoğul

B.S., Mechanical Engineering, Doğuş University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Mechanical Engineering

Boğaziçi University

2025

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my beloved parents for their endless love, patience, and unwavering support throughout my life.

I am sincerely grateful to my thesis advisor, Assoc. Prof. Evren Samur, for his invaluable guidance, encouragement, and continuous support during the course of this study.

I would also like to extend my heartfelt thanks to Assoc. Prof. Vahan Kalenderoğlu, who has inspired and guided me since my undergraduate years and has always shared his knowledge and experience generously.

This thesis was in part supported by the Boğaziçi University Robotics and AI Laboratories Project (ROYAL), funded by the Presidency of Strategy and Budget of the Republic of Türkiye.

Finally, I would like to thank all those who contributed directly or indirectly to this work.

ABSTRACT

PRECISE POSITION CONTROL OF MOBILE MANIPULATORS

Mobile robots function as robotic systems which enable movement through locomotion to execute tasks within unstructured and dynamic environments. The main control challenge of mobile manipulation tasks remains the motion coordination between the mobile base and the manipulator arm because of the complexity that makes simple control architectures impractical. The use of Mecanum wheels on omnidirectional platforms introduces additional challenges because their high maneuverability comes with mechanical wheel slippage which creates substantial uncertainty in position estimation. The proposed research implements an adaptive PID control algorithm that adjusts its control gains based on the size of the tracking error. The research developed an absolute positioning and orientation estimation system that functions in real-time without external sensors to enable closed-loop position tracking. A mobile manipulator system consisting of a Mecanum-wheeled base and a 6-degree-of-freedom industrial robotic arm executed coordinated motion during the research. The evaluation of three trajectory types (trapezoidal, square, and circular) at four velocity levels from 100–400 mm/s was conducted through both simulated and physical experiments.

The adaptive control technique achieves precise trajectory tracking at the millimeter level during low-speed operations. The synchronized motion approach which enables base and manipulator movement together reduces total task duration by 42.5% compared to stop-and-go execution. The research proves that adaptive gain control through error measurements delivers better results and demonstrates synchronized motion improves mobile manipulation performance.

ÖZET

MOBİL MANİPÜLATÖRLERİN HASSAS KONUM KONTROLÜ

Mobil robotlar, lokomosyon kabiliyeti sayesinde yapısal olarak düzensiz ve dinamik ortamlarda görev icra edebilen robotik sistemler olarak işlev görür. Mobil manipülasyon görevlerindeki temel kontrol zorluğu, mobil taban ile manipülatör kol arasındaki hareket koordinasyonudur ve bu, basit kontrol mimarilerinin yetersiz kalmasına neden olacak kadar karmaşıktır. Mecanum tekerleklerin kullanıldığı omnidirectional (çok yönlü) platformlar, yüksek manevra kabiliyetine sahip olmalarına rağmen, mekanik teker kaymaları nedeniyle konum ölçümlerinde ciddi belirsizlikler oluşturarak ek problemler getirir. Bu çalışma, istenen yol hatasının büyüklüğüne göre kontrol kazançlarını ayarlayan bir adaptif PID kontrol algoritması önermektedir. Dış sensörlere ihtiyaç duymadan gerçek zamanlı çalışabilen, mutlak konum ve duruş açısı ölçümü yapan bir sistem geliştirilmiş ve kapalı çevrimli konum takibine olanak sağlanmıştır. Çalışmada, Mecanum tekerlekli bir mobil taban ve 6 serbestlik dereceli endüstriyel bir robot koldan oluşan bir mobil manipülatör sistemi ile koordineli hareket gerçekleştirilmiştir. Üç farklı yol tipi (trapez, kare ve dairesel) ve dört farklı hız seviyesi (100–400 mm/s) hem simülasyon hem de fiziksel deneylerle değerlendirilmiştir.

Adaptif kontrol tekniği, düşük hızlarda milimetre düzeyinde hassas yol takibi sağlamaktadır. Mobil taban ve manipülatörün eşzamanlı hareketine olanak tanıyan senkronize hareket yaklaşımı, görev süresini dur-kalk çalışmaya kıyasla %42.5 oranında azaltmaktadır. Araştırma, hata ölçümüne dayalı adaptif kazanç kontrolünün daha iyi sonuçlar verdiğini ve senkronize hareketin mobil manipülasyon performansını artırdığını göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xviii
LIST OF SYMBOLS . . . xix
LIST OF ACRONYMS/ABBREVIATIONS . . . xx
1. INTRODUCTION . . . 1
1.1. Background and Motivation . . . 1
1.2. Problem Statement . . . 2
1.3. Objectives of the Thesis . . . 2
1.4. Scope and Methodology . . . 3
1.5. Originality and Novelty . . . 3
1.6. Structure of the Thesis . . . 4
2. LITERATURE REVIEW . . . 5
3. 3D DESIGN, MANUFACTURING AND CONTROLLER DESIGN . . . 18
3.1. 3D Design of the Mobile Manipulator . . . 18
3.1.1. Mobile Platform . . . 19
3.1.2. Link 1 and Joint 1 . . . 21
3.1.3. Link 2 and Joint 2 . . . 23
3.1.4. Link 3 and Joint 3 . . . 25
3.1.5. Spherical Wrist and Joints . . . 26
3.2. Manufacturing of the Mobile Manipulator . . . 27
3.3. Mathematical Model of the Mobile Manipulator . . . 29
3.3.1. Forward and Inverse Kinematics of the Mecanum-Wheeled Mobile Base . . . 30
3.3.2. Forward and Inverse Kinematics of the 6 DOF Manipulator . . . 32
3.3.3. Velocity Matching . . . 43

3.4. Controller Design 44
3.4.1. Mecanum Wheeled Mobile Robot Controller 45
3.4.2. 6 DOF Manipulator Controller Design 47
4. SIMULATION AND PROTOTYPE SETUP 50
4.1. Simulation Setup 50
4.2. Prototype Setup 53
4.2.1. Custom Built Absolute Position and Pose Estimator 58
4.2.2. Sensor Calibration Processes 59
5. RESULTS AND DISCUSSION 63
5.1. Overview 63
5.2. Trajectory Tracking at 100 mm/s 63
5.3. Trajectory Tracking at 200 mm/s 67
5.4. Trajectory Tracking at 300 mm/s 71
5.5. Trajectory Tracking at 400 mm/s 75
5.6. Adaptive Gain Behavior and Local Error Analysis 79
5.7. Mobile Manipulator Regular Task Simulation 81
5.8. Mobile Manipulator Synchronized Task Simulation 84
5.9. Discussion 87
6. CONCLUSION AND FUTURE WORK 90
6.1. Conclusion 90
6.2. Contributions 91
6.3. Future Work 92
REFERENCES 94
APPENDIX A: Local Error and Adaptive PID Gain Plots 98
A.1. Trapezoidal Path 98
A.2. Square Path 104
A.3. Circular Path 110
APPENDIX B: COPYRIGHTS 117

LIST OF FIGURES

Figure 2.1. Zhang et al. [10] null-space task prioritization coordinate system placement. 5

Figure 2.2. Chen et al. [11] hierarchical and global path planner work flow scheme. 6

Figure 2.3. Pardi et al. [13] trajectory planing results © 2020 IEEE. 7

Figure 2.4. Sayar et al. [14] 9 dof mobile manipulator. 8

Figure 2.5. Sayar et al. [14] tracking a sinusoidal predefined trajectory in simulation environment. 9

Figure 2.6. Wang et al. [15] (a) trajectory of the mobile base (b) tracking error x of mobile base (c) tracking error y of mobile base © 2023 IEEE. 10

Figure 2.7. Nguyen et al. [16] trajectory tracking results. 11

Figure 2.8. Wan et al. [17] dual-arm mobile manipulator. 12

Figure 2.9. Hernandez et al. [18] high-level and low level control schemes. . . . 13

Figure 2.10. Yung et al. [18] error comparison of H2 and adaptive H2 control method. 14

Figure 2.11. Sun et al. [20] structure of TPANFTSM control system. 15

Figure 3.1. CAD model of the mobile manipulator. 19

Figure 3.2. CAD model of the mecanum base. 21

Figure 3.3. (a)CAD model of the mecanum base (b)belt drive bottom view. . 22

Figure 3.4. Transparent CAD modeling of the first manipulator link. 23

Figure 3.5. CAD modeling of the second manipulator link. 24

Figure 3.6. CAD modeling of the third manipulator link. 25

Figure 3.7. CAD modeling of the spherical wrist. 27

Figure 3.8. Top-down schematic of the mecanum-wheeled robot platform. . . . 30

Figure 3.9. Link parameters of the 6-DOF manipulator. 33

Figure 3.10. Coordinate frame assignment of the full 8-DOF mobile manipulator. 34

Figure 3.11. Illustration of the wrist center in our 6-DOF industrial manipulator. 38

Figure 3.12. Geometric approach schematic. 40

Figure 3.13. Proposed adaptive controller trajectory tracking block diagram. . 48

Figure 3.14. Manipulator PID controller block diagram. 49

Figure 4.1. Simulink model of the adaptive PID controller for mecanum base. 51

Figure 4.2. Simulink model of forward kinematics for the mobile manipulator. 52

Figure 4.3. Prototype mecanum platfrom. 54

Figure 4.4. Prototype of the mobile manipulator. 56

Figure 4.5. Mobile manipulator workspace visualization. 57

Figure 4.6. Mobile manipulator position and posture tracking device. 58

Figure 4.7. PMW3360 pixel shift demonstration. 59

Figure 4.8. 3D printed PMW3360 calibration part. 60

Figure 4.9. MPU6050 sensor. 62

Figure 5.1. Simulated trajectory and posture angle tracking on trapezoidal path at 100 mm/s. 64

Figure 5.2. Experimental trajectory and posture angle tracking on trapezoidal path at 100 mm/s. 65

Figure 5.3. Simulated trajectory and posture angle tracking on square path at 100 mm/s. 65

Figure 5.4. Experimental trajectory and posture angle tracking on square path at 100 mm/s. 66

Figure 5.5. Simulated trajectory and posture angle tracking on circular path at 100 mm/s. 66

Figure 5.6. Experimental trajectory and posture angle tracking on circular path at 100 mm/s. 67

Figure 5.7. Simulated trajectory and posture angle tracking on trapezoidal path at 200 mm/s. 68

Figure 5.8. Experimental trajectory and posture angle tracking on trapezoidal path at 200 mm/s. 69

Figure 5.9. Simulated trajectory and posture angle tracking on square path at 200 mm/s. 69

Figure 5.10. Experimental trajectory and posture angle tracking on square path at 200 mm/s. 70

Figure 5.11. Simulated trajectory and posture angle tracking on circular path at 200 mm/s. 70

Figure 5.12. Experimental trajectory and posture angle tracking on circular path at 200 mm/s. 71

Figure 5.13. Simulated trajectory and posture angle tracking on trapezoidal path at 300 mm/s. 72

Figure 5.14. Experimental trajectory and posture angle tracking on trapezoidal path at 300 mm/s. 73

Figure 5.15. Simulated trajectory and posture angle tracking on square path at 300 mm/s. 73

Figure 5.16. Experimental trajectory and posture angle tracking on square path at 300 mm/s. 74

Figure 5.17. Simulated trajectory and posture angle tracking on circular path at 300 mm/s. 74

Figure 5.18. Experimental trajectory and posture angle tracking on circular path at 300 mm/s. 75

Figure 5.19. Simulated trajectory and posture angle tracking on trapezoidal path at 400 mm/s. 76

Figure 5.20. Experimental trajectory and posture angle tracking on trapezoidal path at 400 mm/s. 77

Figure 5.21. Simulated trajectory and posture angle tracking on square path at 400 mm/s. 77

Figure 5.22. Experimental trajectory and posture angle tracking on square path at 400 mm/s. 78

Figure 5.23. Simulated trajectory and posture angle tracking on circular path at 400 mm/s. 78

Figure 5.24. Experimental trajectory and posture angle tracking on circular path at 400 mm/s. 79

Figure 5.25. Experimental local error in x and adaptive PID gain response for circular path at 100 mm/s. 80

Figure 5.26. Simulated local error in x and adaptive PID gain response for circular path at 100 mm/s. 81

Figure 5.27. Movement detection of the robot arm in the regular task configuration. 82

Figure 5.28. Base trajectory tracking and orientation error during regular task operation. 83

Figure 5.29. 3D trajectory of the tool frame in the regular task operation. . . . 84

Figure 5.30. Continuous movement detection of the manipulator during synchronized task execution. 85

Figure 5.31. 3D trajectory of the tool frame during synchronized task operation. 86

Figure 5.32. Base trajectory tracking and orientation error during synchronized task operation. 87

Figure A.1. Experimental local error in x and adaptive PID gain response on trapezoidal path at 100 mm/s. 98

Figure A.2. Simulated local error in x and adaptive PID gain response on trapezoidal path at 100 mm/s. 99

Figure A.3. Experimental local error in y and adaptive PID gain response on trapezoidal path at 100 mm/s. 99

Figure A.4. Simulated local error in y and adaptive PID gain response on trapezoidal path at 100 mm/s. 100

Figure A.5. Experimental local error in ϕ and adaptive PID gain response on trapezoidal path at 100 mm/s. 100

Figure A.6. Simulated local error in ϕ and adaptive PID gain response on trapezoidal path at 100 mm/s. 101

Figure A.7. Experimental local error in x and adaptive PID gain response on trapezoidal path at 200 mm/s. 101

Figure A.8. Simulated local error in x and adaptive PID gain response on trapezoidal path at 200 mm/s. 102

Figure A.9. Experimental local error in y and adaptive PID gain response on trapezoidal path at 200 mm/s. 102

Figure A.10. Simulated local error in y and adaptive PID gain response on trapezoidal path at 200 mm/s. 103

Figure A.11. Experimental local error in ϕ and adaptive PID gain response on trapezoidal path at 200 mm/s. 103

Figure A.12. Simulated local error in ϕ and adaptive PID gain response on trapezoidal path at 200 mm/s. 104

Figure A.13. Experimental local error in x and adaptive PID gain response on square path at 100 mm/s. 104

Figure A.14. Simulated local error in x and adaptive PID gain response on square path at 100 mm/s. 105

Figure A.15. Experimental local error in y and adaptive PID gain response on square path at 100 mm/s. 105

Figure A.16. Simulated local error in y and adaptive PID gain response on square path at 100 mm/s. 106

Figure A.17. Experimental local error in ϕ and adaptive PID gain response on square path at 100 mm/s. 106

Figure A.18. Simulated local error in ϕ and adaptive PID gain response on square path at 100 mm/s. 107

Figure A.19. Experimental local error in x and adaptive PID gain response on square path at 200 mm/s. 107

Figure A.20. Simulated local error in x and adaptive PID gain response on square path at 200 mm/s. 108

Figure A.21. Experimental local error in y and adaptive PID gain response on square path at 200 mm/s. 108

Figure A.22. Simulated local error in y and adaptive PID gain response on square path at 200 mm/s. 109

Figure A.23. Experimental local error in ϕ and adaptive PID gain response on square path at 200 mm/s. 109

Figure A.24. Simulated local error in ϕ and adaptive PID gain response on square path at 200 mm/s. 110

Figure A.25. Experimental local error in x and adaptive PID gain response on circular path at 100 mm/s. 110

Figure A.26. Simulated local error in x and adaptive PID gain response on circular path at 100 mm/s. 111

Figure A.27. Experimental local error in y and adaptive PID gain response on circular path at 100 mm/s. 111

Figure A.28. Simulated local error in y and adaptive PID gain response on circular path at 100 mm/s. 112

Figure A.29. Experimental local error in ϕ and adaptive PID gain response on circular path at 100 mm/s. 112

Figure A.30. Simulated local error in ϕ and adaptive PID gain response on circular path at 100 mm/s. 113

Figure A.31. Experimental local error in x and adaptive PID gain response on circular path at 200 mm/s. 113

Figure A.32. Simulated local error in x and adaptive PID gain response on circular path at 200 mm/s. 114

Figure A.33. Experimental local error in y and adaptive PID gain response on circular path at 200 mm/s. 114

Figure A.34. Simulated local error in y and adaptive PID gain response on circular path at 200 mm/s. 115

Figure A.35. Experimental local error in ϕ and adaptive PID gain response on circular path at 200 mm/s. 115

Figure A.36. Simulated local error in ϕ and adaptive PID gain response on circular path at 200 mm/s. 116

Figure B.1. Copyright permission for *path planning for mobile manipulator robots under non-holonomic and task constraints.* 117

Figure B.2. Copyright permission for *whole-body control of an autonomous mobile manipulator using model predictive control and adaptive fuzzy technique.* 118

Figure B.3. Copyright permission for *tajectory tracking using linear state feedback controller for a mecanum wheel omnidirectional* page 1. . . . 119

Figure B.4. Copyright permission for *tajectory tracking using linear state feedback controller for a mecanum wheel omnidirectional* page 2. . . . 120

Figure B.5. Copyright permission for *precise trajectory tracking of mecanum-wheeled omnidirectional mobile robots via a novel fixed-time sliding mode control approach* page 1. 121

Figure B.6. Copyright permission for *precise trajectory tracking of mecanum-wheeled omnidirectional mobile robots via a novel fixed-time sliding mode control approach* page 2. 122

LIST OF TABLES

Table 2.1. Comparison of control strategies for mecanum-wheeled mobile manipulators. 17

Table 3.1. Denavit-Hartenberg parameters. 35

Table 4.1. BOM for prototype mobile manipulator. 55

Table 5.1. Trajectory tracking errors at 100 mm/s for different path types. . . 67

Table 5.2. Trajectory tracking errors at 200 mm/s for different path types. . . 71

Table 5.3. Trajectory tracking errors at 300 mm/s for different path types. . . 75

Table 5.4. Trajectory tracking errors at 400 mm/s for different path types. . . 79

LIST OF SYMBOLS

e_x, e_y, e_ϕ	Local position and orientation errors
K_p, K_i, K_d	Proportional, Integral, and Derivative gains
L, W	Length and width of the mobile base [m]
p_{cur}	Current (measured) global position vector
p_{des}	Desired global position vector
r	Radius of the mecanum wheels [m]
R	Rotation or transformation matrix between frames
t	Time variable [s]
V_x, V_y	Linear velocities along x and y axes [m/s]
ω	Angular velocity of the robot base [rad/s]
X, Y	Global Cartesian coordinates [m]
T	Time vector used in plotting or indexing
ϕ	Posture orientation angle [rad]
θ	Joint angle or rotational coordinate [rad]

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
BOM	Bill of Materials
DMP	Digital Motion Processor
DOF	Degrees of Freedom
GBNN+H	Glasius Bioinspired Neural Network with Heuristics
I2C	Inter-Integrated Circuit
IK	Inverse Kinematics
IMU	Inertial Measurement Unit
MBPS	Megabits Per Second
MPC	Model Predictive Control
MWMR	Mecanum Wheeled Mobile Robot
NRF	Nordic Radio Frequency
PID	Proportional–Integral–Derivative
PTFE	Polytetrafluoroethylene
RL	Reinforcement Learning
RMSE	Root Mean Square Error
ROS	Robot Operating System
SMC	Sliding Mode Control
TPANFTSM	Two-Phase Attractor Nonsingular Fast Terminal Sliding Mode
UWB	Ultra-Wideband

1. INTRODUCTION

Mobile manipulators which unite robotic manipulator systems with mobile bases have gained popularity in industrial service and research robotics applications because of their extensive operational area and adaptability [1]. The combination of stationary robotic arms with mobile bases enables these systems to handle bigger and more active operational environments [2]. The omnidirectional movement capability makes mecanum-wheeled platforms highly appealing among different mobile base types [3]. Robots that use this design are known as mecanum robots because they maintain their posture angle while moving in any direction [4].

The mecanum-wheeled mobile robots maintain excellent kinematic properties but experience major positioning inaccuracies because of wheel slippage and inadequate surface traction. Real-world applications show that the manipulator arm achieves sub-millimeter precision but the base introduces centimeter-level drift and angular posture error which results in poor positioning accuracy [5]. The system experiences maximum slippage during lateral and rotational movements which makes odometry unreliable and produces increasing positional errors throughout time.

Most mobile manipulation systems implement stop-perform-move strategies to handle this limitation by having the platform stop at a target position before the manipulator performs its task [6]. The stabilization of the base through this method leads to longer operation times and decreased system efficiency. The main objective of this research is to enhance mobile platform positioning accuracy to allow precise manipulator movements during base movement.

1.1. Background and Motivation

The combination of advanced control algorithms with rigid mechanical design and precise sensors enables modern fixed base robotic manipulators to reach extremely

high accuracy levels. The localization accuracy of mobile bases remains restricted because mecanum wheels provide limited precision. The roller design of Mecanum wheels creates slippage which leads to unreliable odometry and drift during lateral and rotational movements.

The system accuracy reaches its maximum limit because of the difference between the manipulator arm precision and the mobile base precision. Most systems use the stop-perform-move strategy as their primary compensation method [7] [8]. The stop-perform-move strategy reduces operational efficiency as extending task duration [9]. This thesis aims to create a control system which enables highly accurate robotic manipulator operation during mobile base movement to shorten task duration and enhance system fluidity.

1.2. Problem Statement

Although robotic arms can be controlled with great precision, the mobile bases —especially those with Mecanum wheels— introduce significant errors in positioning due to inherent mechanical and control limitations. This mismatch between arm and base precision hinders the performance of mobile manipulators in high-accuracy tasks. The primary problem tackled in this thesis is: *How can the positioning accuracy of a Mecanum-wheeled mobile base of a mobile manipulator be improved sufficiently to enable high-precision manipulation without requiring the base to stop?*

1.3. Objectives of the Thesis

The objectives of this research are as follows:

- Develop a time-varying PID control strategy for the mecanum-wheeled mobile base of a mobile manipulator to minimize positioning errors.
- Implement a coordinated control scheme that combines this advanced controller for the base with a conventional PID controller for the robotic arm of the mobile

manipulator.

- Demonstrate that the proposed control approach enables accurate manipulation while the base of the mobile manipulator is in motion.
- Quantitatively evaluate the improvements in task completion time and overall system accuracy compared to the traditional stop-perform-move strategy.

1.4. Scope and Methodology

This thesis is limited to mobile manipulators operating in indoor environments with smooth, flat surfaces, and focuses on control strategies rather than mechanical or sensor hardware enhancements. The proposed control methods are evaluated using both simulations and physical experiments on a custom-built mobile manipulator.

The methodology involves:

- Modeling the kinematics of the mecanum-wheeled base and the robotic arm of a mobile manipulator.
- Designing and simulating the proposed control algorithms.
- Implementing the algorithms on a real robot using onboard sensors such as wheel magnetic encoders, IMUs, optical flow sensors and current meters.
- Performing comparative experiments to evaluate task success, positioning error, and efficiency.

1.5. Originality and Novelty

The literature shows various adaptive control strategies that implement time-varying gain scheduling based on system dynamics and tracking error trends yet our method presents an original approach by using tracking error magnitude in local coordinates to adjust PID gains. The linear first-degree equations modify gains through a direct relationship between error magnitude and integral gain reduction which accelerates convergence while preventing overshoot during error reduction. The research

introduces an innovative hardware solution which enables local position and posture estimation through internal systems without requiring external infrastructure. The device combines two sensors which operate as follows:

- PMW3360 optical camera receives ground texture data to performs planar displacement calculations through successive frame comparisons.
- The MPU-6050 IMU detects the posture angle through z-axis rotation.

The sensors operate together to provide absolute robot localization at high frequencies which enables self-contained and robust control operations despite environmental limitations. The following sections present a comprehensive explanation of the custom localization device development along with its operational principles and system integration details.

1.6. Structure of the Thesis

This thesis is organized as follows:

- Chapter 1: Introduction — Provides motivation, definition of the problem, novelty status and describes the structure of the thesis.
- Chapter 2: Literature Review — Summarizes related work in mobile manipulation, Mecanum kinematics, and control techniques.
- Chapter 3: Modeling and Controller Design — Develops the system models and presents the proposed control algorithms.
- Chapter 4: Implementation and Experimental Setup — Describes the hardware, software architecture, and experimental design.
- Chapter 5: Results and Discussion — Presents and analyzes the experimental results.
- Chapter 6: Conclusion and Future Work — Summarizes the contributions and outlines potential future directions.

2. LITERATURE REVIEW

Mobile manipulators represent a combination of two historically distinct branches of robotics: mobile platforms and robotic manipulators. While traditional manipulators have long achieved impressive precision within fixed workspaces, the challenge of extending this dexterity across a broader operational area led to the evolution of mobile manipulators. Their appeal lies in their ability to navigate and manipulate—however, the dual-control challenge of coordinating locomotion and manipulation creates significant complexity, especially for high-precision applications.

A central obstacle in achieving high positioning accuracy lies in the mobility subsystem. Most mobile manipulators rely on wheeled platforms, with Mecanum-wheeled robots offering omnidirectional movement. While this mobility enhances maneuverability in constrained spaces, it also introduces greater sensitivity to wheel slippage, odometry drift, and external disturbances, all of which impact the overall end-effector precision.

Zhang et al. [10] addressed the redundancy of mobile manipulators by proposing a null-space task prioritization control scheme. Their approach separates base and arm commands while considering task hierarchy as shown in Figure 2.1, allowing lower-priority actions (like base stabilization) to occur without interfering with manipulation. Through hybrid control involving dynamic feedback, the controller ensured precise tracking and collision avoidance under complex trajectories.

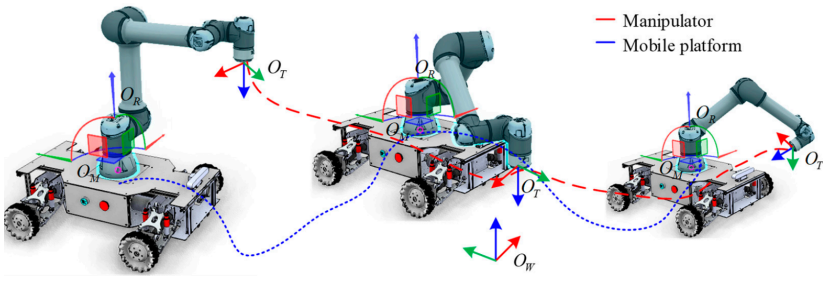

Figure 2.1. Zhang et al. [10] null-space task prioritization coordinate system placement.

To complement task prioritization, Chen et al. [11] proposed a hierarchical planner combining global path planning for the base and local task planning for the manipulator as shown in Figure 2.2. This division allowed dynamic arm reconfiguration during locomotion, which proved vital when navigating environments with obstacles or limited reachability. Their results on real-world mobile platforms show a significant reduction in task time when coordination is done hierarchically.

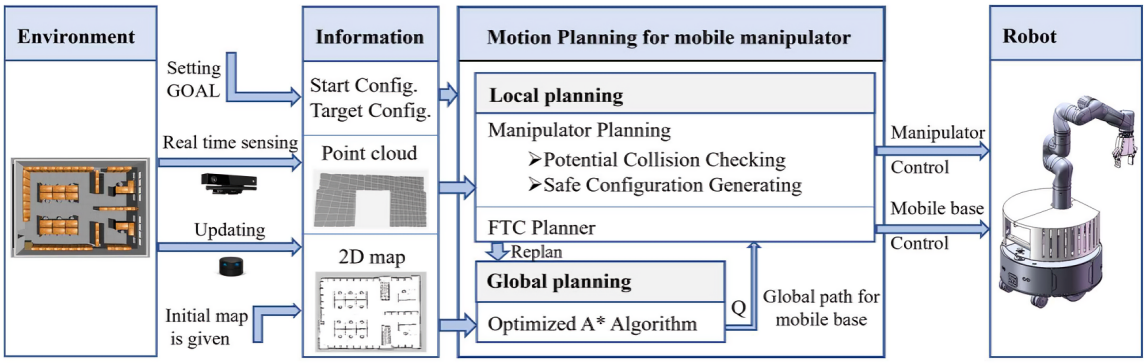

Figure 2.2. Chen et al. [11] hierarchical and global path planner work flow scheme.

Sleiman et al. [12] advanced the state of whole-body control with a unified MPC approach. Unlike decoupled or sequential planners, their system encoded the entire kinematic and dynamic model into a single cost function optimized over a prediction horizon. The result was a smooth and responsive system capable of handling disturbances such as object displacement or dynamic targets. This approach is particularly useful in warehouse automation and healthcare logistics, where unpredictable changes are common. Their real-time feasibility is demonstrated through ROS2 implementation on a physical robot.

In contexts where base manipulation must respect contact constraints—such as surface cutting or inspection—Pardi et al. [13] developed a constrained Jacobian framework that embedded both non-holonomic base constraints and manipulability optimization into trajectory planning. This ensured the manipulator maintained desirable configurations throughout tasks while adhering to surface-following requirements as shown in Figure 2.3.

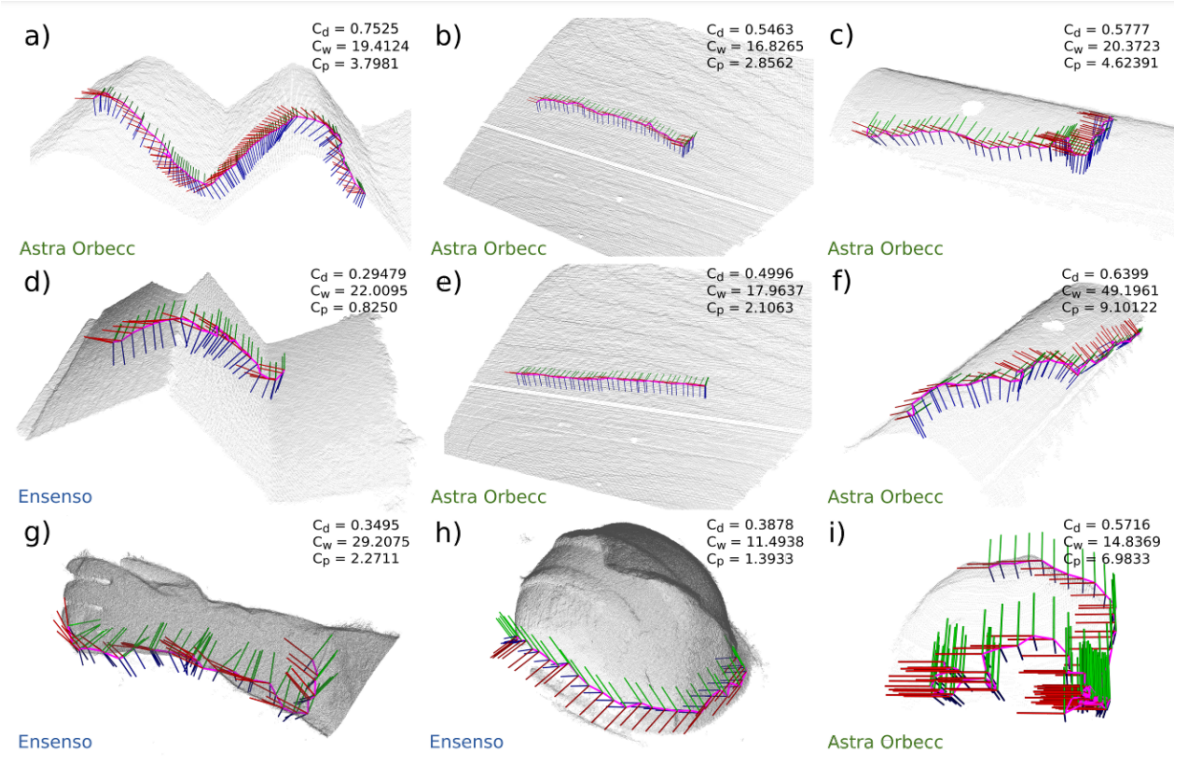

Figure 2.3. Pardi et al. [13] trajectory planing results © 2020 IEEE.

Sayar et al. [14] introduced a real-time capable hierarchical control architecture using recurrent neural networks. Their approach formulates a quadratic programming problem that optimizes joint velocities under redundancy, obstacle constraints, and actuation limits. Applied to a 9-DOF mobile manipulator shown in Figure 2.4, the controller achieved continuous manipulation under dynamically changing workspaces as shown in Figure 2.5. This study highlights the importance of integrating machine learning-based optimizers into real-time robot control, especially for highly redundant systems.

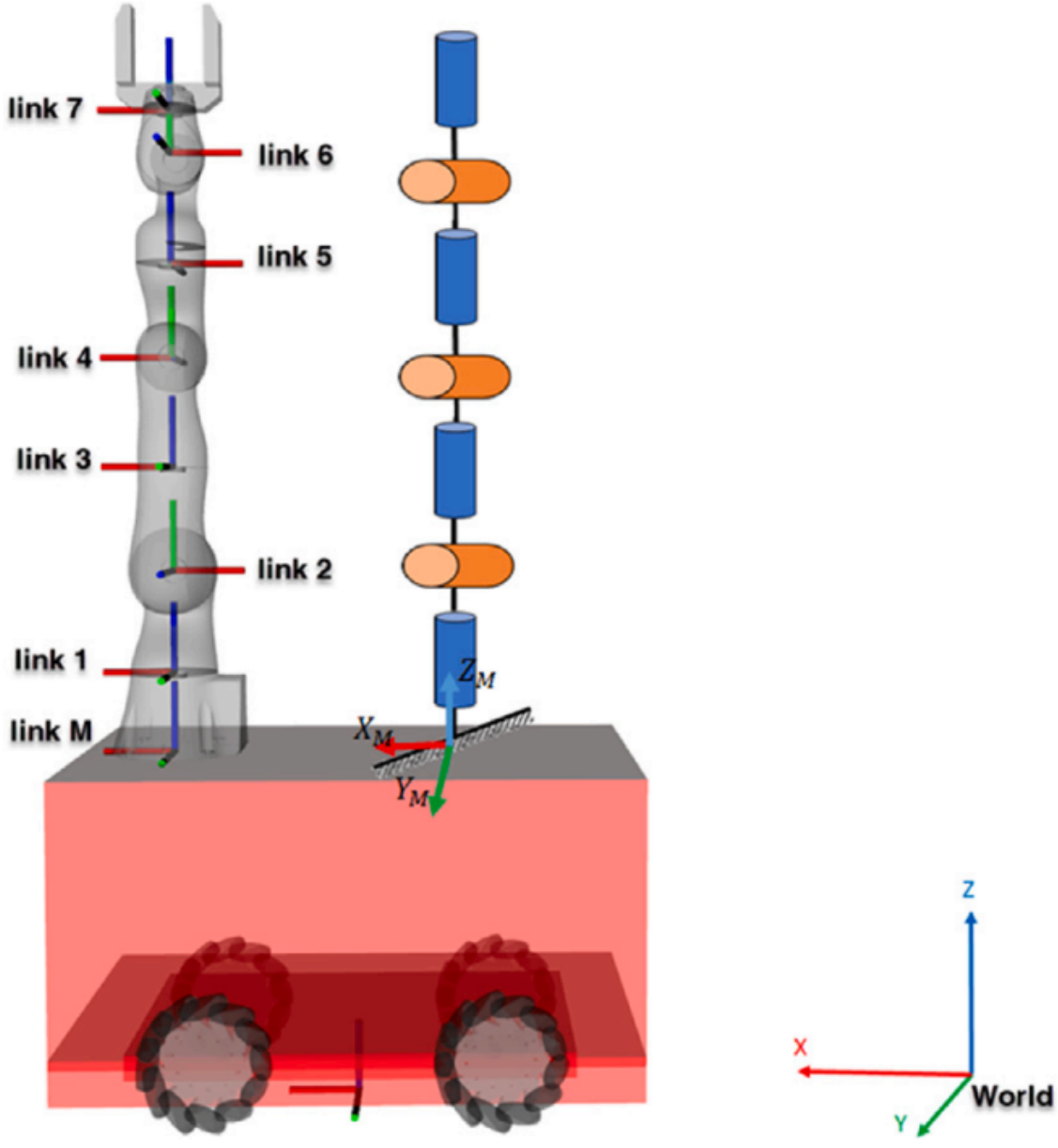

Figure 2.4. Sayar et al. [14] 9 dof mobile manipulator.

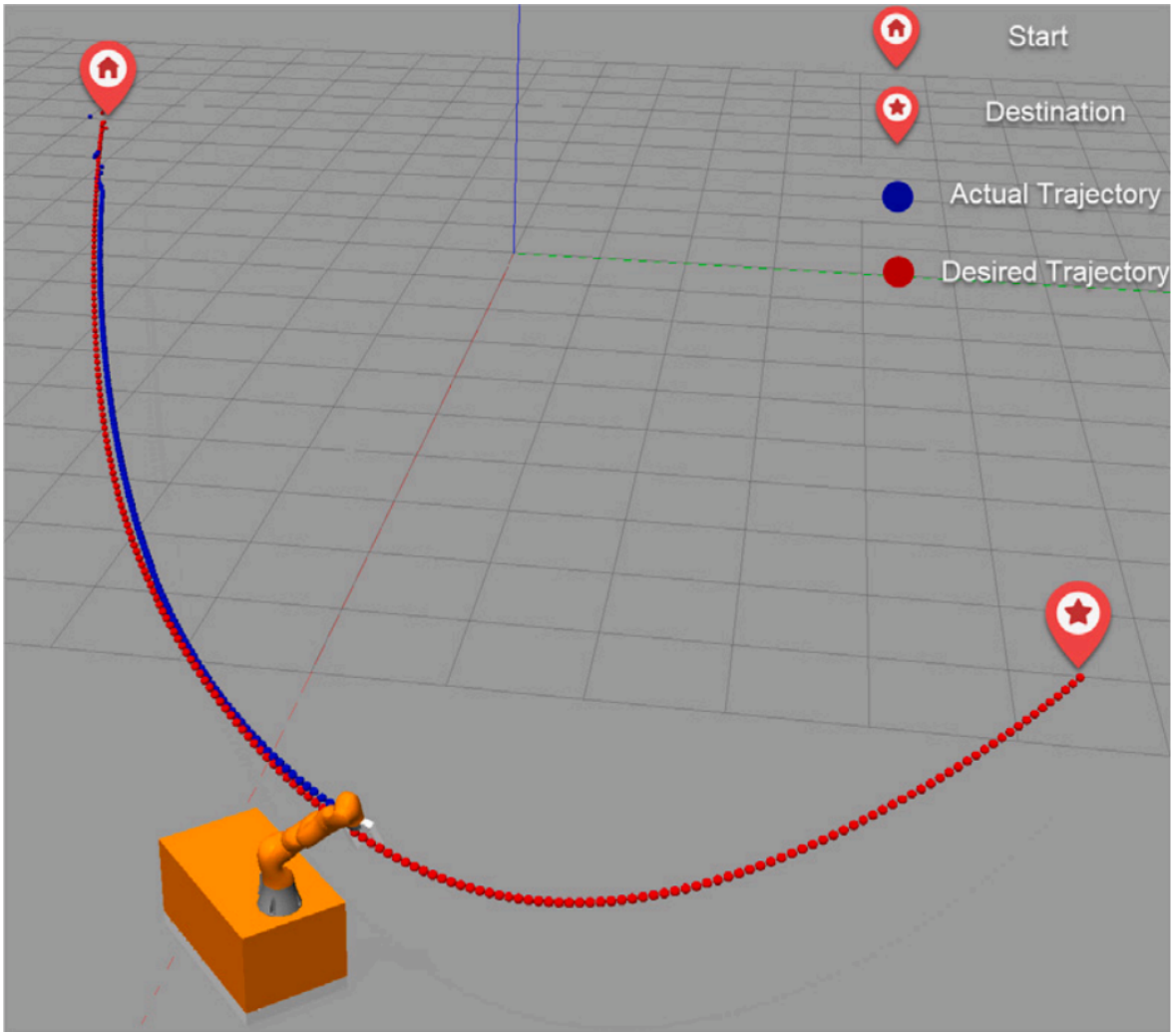

Figure 2.5. Sayar et al. [14] tracking a sinusoidal predefined trajectory in simulation environment.

Another dimension of enhancing precision control lies in the improvement of feedback systems. Wang et al. [15] tackled model uncertainty by combining model predictive control with fuzzy logic approximation. Their adaptive fuzzy model learned residual dynamics and compensated for modeling errors online. As a result, the controller could maintain stable and accurate performance even with unknown disturbances. Figure 2.6 shows trajectory tracking error results of traditional fuzzy logic controller and proposed method. This method is especially beneficial in environments where parameter variations are frequent (e.g., payload changes or wheel wear).

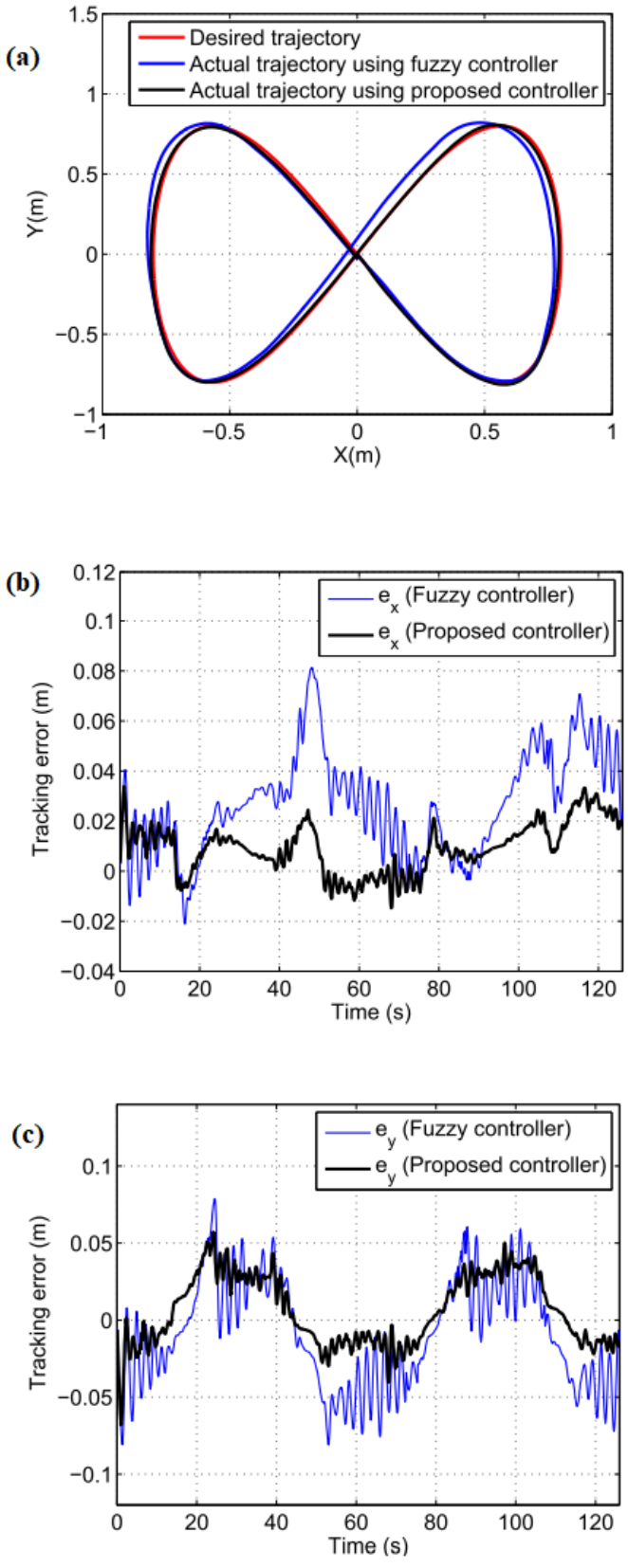

Figure 2.6. Wang et al. [15] (a) trajectory of the mobile base (b) tracking error x of mobile base (c) tracking error y of mobile base © 2023 IEEE.

In a simpler yet effective trajectory tracking strategy, Nguyen et al. [16] employed a linear state-feedback controller with pole-placement for mecanum kinematics. While not as adaptive, the benefit lies in its low computational burden and ease of tuning, making it favorable for budget-constrained or tightly controlled industrial scenarios. Figure 2.7 shows trajectory tracking results of the proposed linear state-feedback controller method.

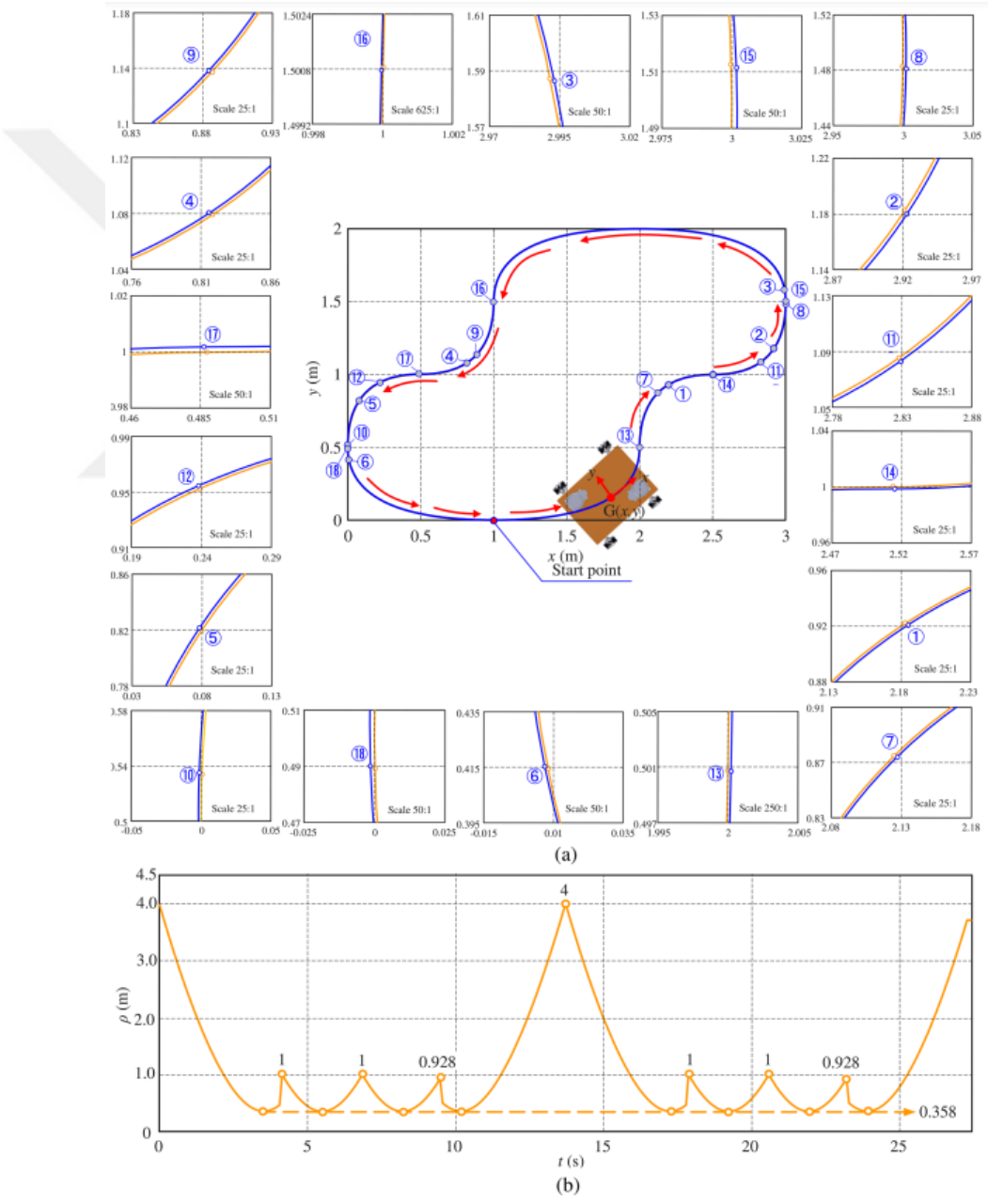

Figure 2.7. Nguyen et al. [16] trajectory tracking results.

Sensor integration plays a crucial role in closing the loop for precise control. Wan et al. [17] explored the GBNN+H (Glasius Bioinspired Neural Network with Heuristics) for complete surface coverage. Though the application was surface cleaning in hospitals, the coverage optimization principles are generalizable to manipulation tasks requiring exhaustive exploration or inspection. Their dual-arm mobile manipulator shown in Figure 2.8 adapted to irregular floor plans and dynamic obstacles, maintaining near-complete surface coverage with low computational cost.

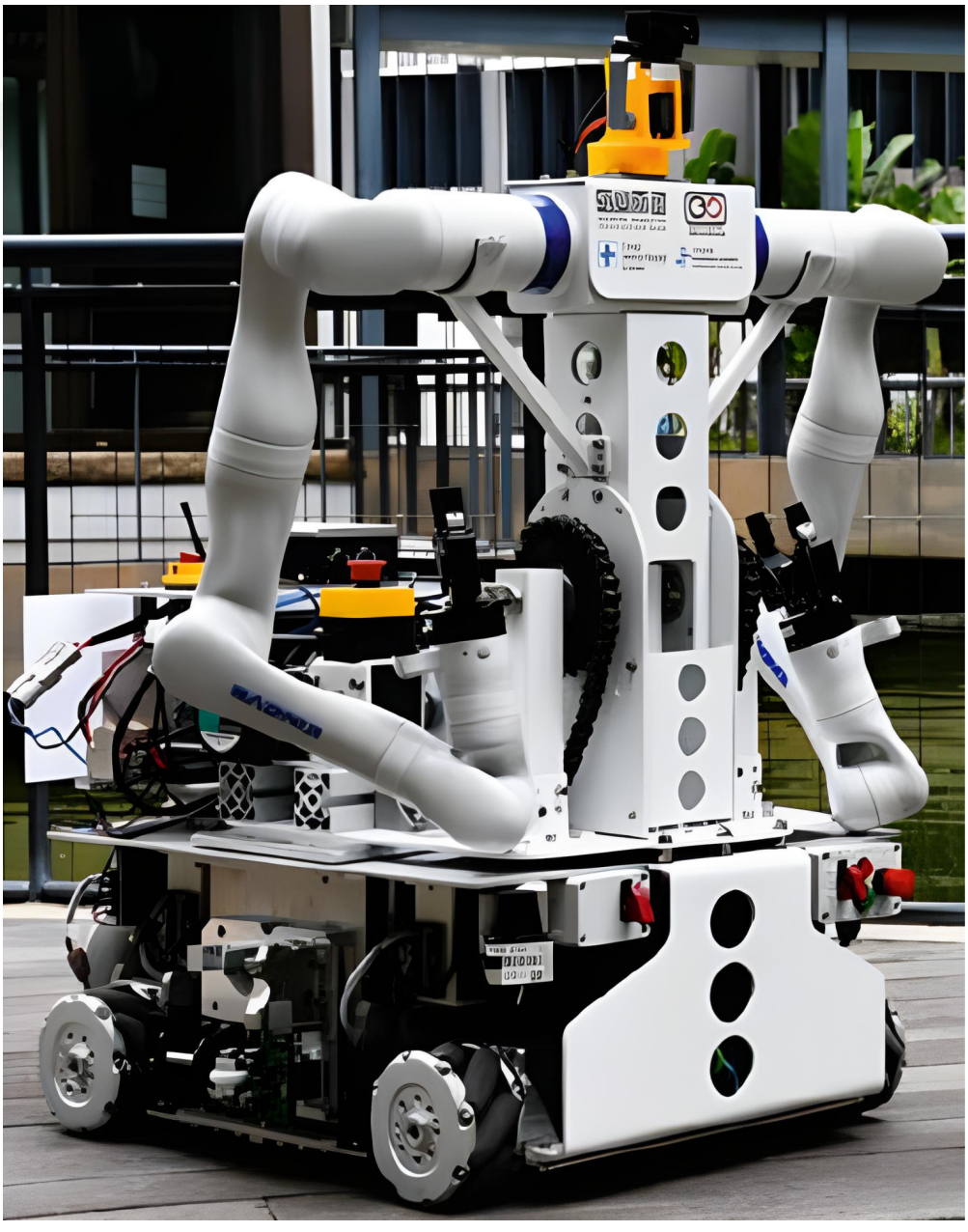

Figure 2.8. Wan et al. [17] dual-arm mobile manipulator.

Recent studies continue to highlight novel control approaches for addressing the unique challenges of Mecanum-wheeled mobile robots. Ortiz Hernández and Almeida [18] proposed a hierarchical kinematic control method using both high-level and low-level control loops, where the high-level controller handles global pose tracking and the low-level one manages wheel speed. Figure 2.9 show both high-level and low-level control scheme. Their experimental validation using 120 test scenarios showed that combining PID with quasi-sliding mode significantly improves robustness while maintaining ease of implementation. The method treats the system as a black box, focusing on input-output behaviors, making it practical for hardware-limited real-time control systems.

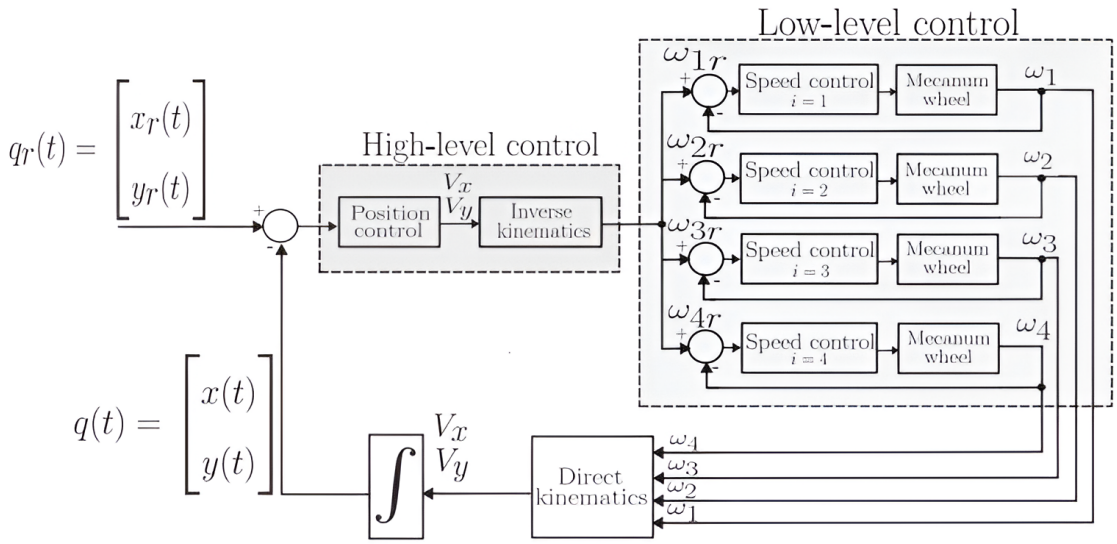

Figure 2.9. Hernandez et al. [18] high-level and low level control schemes.

Yung-Hsiang Chen [19] developed a nonlinear adaptive H2 optimal control method to enhance the tracking accuracy of MWMRs under model uncertainties as shown in Figure 2.10. By analytically deriving an adaptive law that compensates for parameter deviations in mass and inertia, the control system achieved significant improvement in trajectory tracking accuracy and energy efficiency. The proposed scheme showed marked superiority over traditional H2 controllers in both simulation and physical trials, particularly when operating under variable payload conditions.

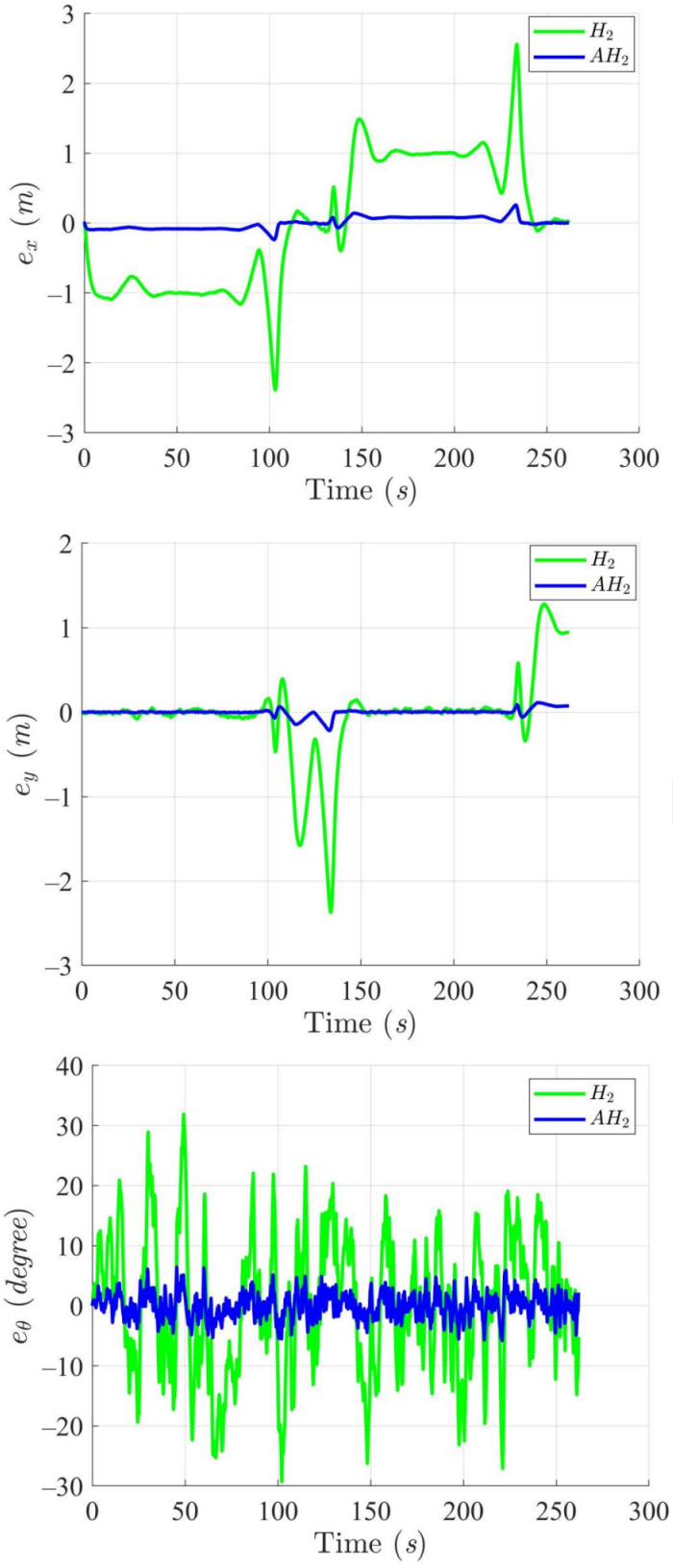

Figure 2.10. Yung et al. [18] error comparison of H2 and adaptive H2 control method.

Sun et al. [20] introduced a fixed-time nonsingular terminal sliding mode control (TPANFTSM) tailored for MWMRs. Figure 2.11 shows the structure of the control system. This method innovatively incorporates a two-phase attractor structure in the sliding surface, ensuring finite-time convergence and high tracking precision without singularities. Their experimental trials under multiple trajectory shapes demonstrated not only reduced error bounds but also increased robustness to chattering and modeling discrepancies. The TPANFTSM controller emerged as a leading-edge candidate among recent SMC strategies for mobile manipulators.

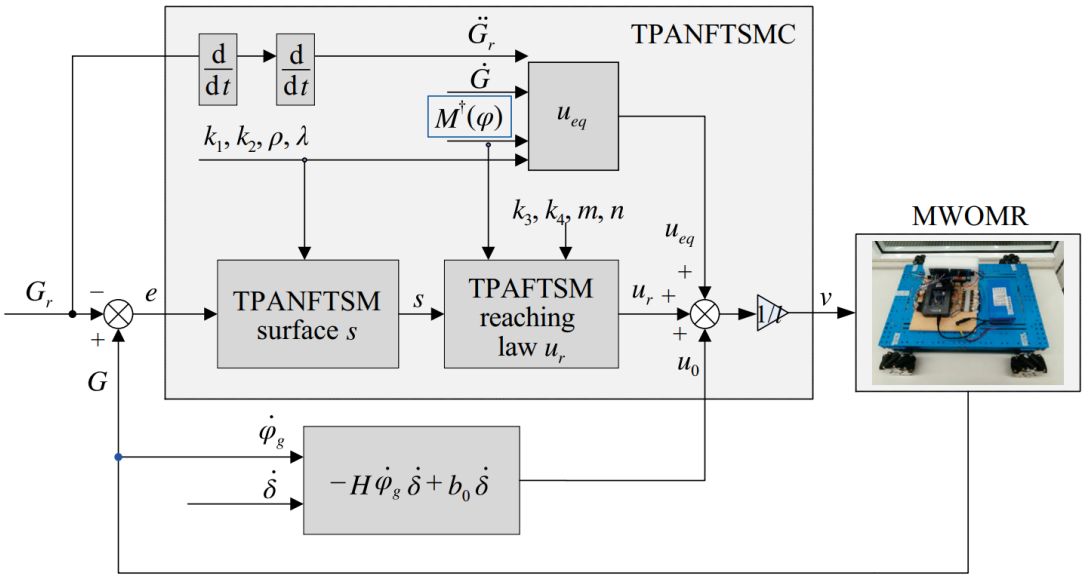

Figure 2.11. Sun et al. [20] structure of TPANFTSM control system.

Collectively, the literature reveals a shift toward integrated control architectures that merge traditional control theory (e.g., PID, MPC) with modern learning methods (e.g., neural networks, fuzzy logic). The challenges of dynamic environments, mechanical compliance, and uncertain kinematics necessitate such hybrid solutions. Especially in Mecanum-wheeled mobile manipulators, where non-holonomic effects and ground slip can distort feedback, the role of adaptive control and robust sensor fusion becomes indispensable. Future directions appear to be moving toward real-time optimization, task-aware perception, and on-the-move manipulation, where control decisions are made in milliseconds while preserving long-term accuracy.

Table 2.1 presents a summary of recent control strategies for Mecanum-wheeled mobile manipulators by showing their fundamental concepts alongside their advantages and possible drawbacks. Our method stands apart from other approaches because it uses minimal computational resources and direct state feedback to achieve high-speed control on low-resource platforms. We proposed a lightweight adaptive PID controller to enable high-speed on-the-move operation on low-resource microcontroller architectures. The development of a dedicated hardware module provided direct planar displacement measurement with 2.1 μm resolution in both x and y directions instead of traditional velocity output integration methods or external machine vision that limits the workspace of the mobile manipulator. The measurement system allows posture estimation including orientation without requiring external tracking systems or infrastructure which enables untethered deployment without spatial constraints. The system achieves fast response times and high positioning accuracy through direct measurement and adaptive control law which eliminates feedback control loop delays and data transfer problems. The system demonstrates fast response times and high positioning accuracy which solves one major challenge for mobile robotic platforms to achieve robust manipulation during motion.

Table 2.1. Comparison of control strategies for mecanum-wheeled mobile manipulators.

Control Strategy	**Key Features**	**Advantages / Limitations**
Null-Space Hybrid Control [10]	Task prioritization between base and manipulator using redundancy	Effective redundancy use; sensitivity to model changes
Hierarchical Motion Planning [11]	Global-local planner separation with real-time adaptability	Efficient for structured environments; may lack dynamic responsiveness
Unified MPC [12]	Whole-body optimization under hybrid dynamics	Handles complex coupling well; requires high computation power
Adaptive Fuzzy MPC [15]	Combines fuzzy logic with MPC for uncertainty adaptation	Robust under uncertainty; tuning complexity
Linear State Feedback [16]	Pole-placement design for Mecanum platforms	Fast response; limited to structured, predictable tasks
RNN-based Hierarchical QP [14]	Neural-network-assisted QP optimization with redundancy handling	High flexibility and constraint handling; resource-intensive
Hierarchical Kinematic PID+QSMC [18]	Dual-layer kinematic control with quasi-sliding mode feedback	Real-time feasible; black-box model restricts internal adaptation
Nonlinear Adaptive H2 Optimal Control [19]	Online parameter adaptation compensating mass/inertia uncertainties	High precision and energy efficiency; parameter estimation complexity
Fixed-Time TPANFTSM Control [20]	Two-phase sliding surface for fixed-time convergence without singularity	Fast convergence and strong robustness; requires careful gain design

3. 3D DESIGN, MANUFACTURING AND CONTROLLER DESIGN

In order to test and validate the experimental studies and the control algorithms developed throughout this thesis, a custom mobile manipulator was designed and built. The chapter describes the complete 3D design, manufacturing, and controller design procedures for the mobile manipulator developed in this research. The mechanical design process used SolidWorks software to create a design that prioritized manufacturing feasibility. The selection of components focused on local market availability to achieve quick prototyping and economical fabrication.

The control architecture implements a dual-stage approach. The Mecanum-wheeled mobile base receives control signals from a time-varying (adaptive) PID controller which modifies its gains to enhance trajectory tracking performance when dealing with wheel slip and sensor drift issues. The control method has another name which is variable-gain PID method. The 6-DOF robotic arm installed on the mobile base operates with two independent PID controllers which handle torque control and position accuracy separately for each joint. The modular control approach enables both base and manipulator to move accurately while remaining decoupled which supports effective task execution when the system performs simultaneous locomotion and manipulation.

3.1. 3D Design of the Mobile Manipulator

The 3D modeling of the mobile manipulator was carried out using SolidWorks software, utilizing off-the-shelf bearings, belts, grippers, and pulleys. A SolidWorks render of the manipulator is shown in Figure 3.1. A total of 122 parts were designed, of which 53 could be directly purchased from the industrial market. The remaining components were custom-designed to fit the specific requirements of our industrial robotic arm architecture and Mecanum-wheeled mobile platform.

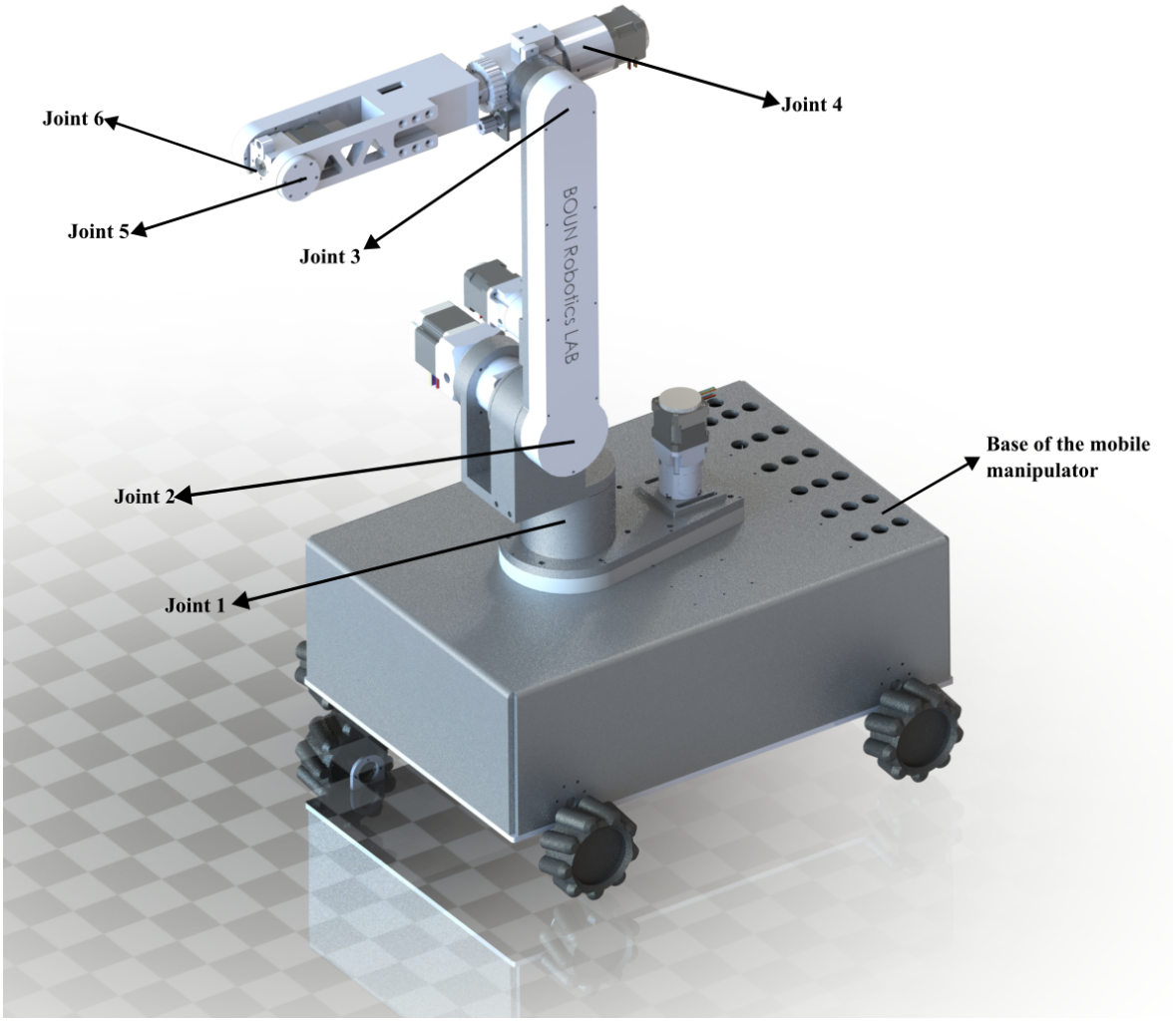

Figure 3.1. CAD model of the mobile manipulator.

3.1.1. Mobile Platform

The mecuanum wheel mobile robot represents an important solution to provide omnidirectional mobility in structured or constrained environments. The mecanum wheel design enables holonomic motion because it allows robots to move forward, backward, to the left, to the right and diagonally without changing their orientation. The ability of mecanum wheels to maintain orientation stability during sideways motion proves beneficial for mobile manipulators that need to position themselves precisely

next to or under target objects. The ability to separate translation and rotation movements through kinematic analysis improves motion planning and enables straightforward control approaches that work well when robotic arm operations occur during base movement.

The implementation of Mecanum wheels generates multiple obstacles that mainly stem from mechanical complexity together with reduced traction performance. The angled passive rollers on each wheel make mecanum platforms prone to wheel slippage when operating on smooth or low-friction surfaces. The system becomes unreliable during lateral or rotational motions because wheel slippage produces inaccurate odometry readings and drift. The mecanum wheel design requires more energy for operation and produces less load capacity than standard wheeled systems because of its mechanical construction. The existing system requires sophisticated control systems as well as sensor fusion methods to maintain precise navigation and trajectory tracking. The mecanum-based design remains the preferred choice when space requirements and omnidirectional movement stand as critical factors in applications.

The research involved creating a customized mobile base which used mecanum wheels for precise manipulation tasks during movement. SolidWorks software has been used to create platform designs that emphasized production simplicity and robotic arm compatibility. The system utilized standard motor components and encoder units together with custom-built elements to maximize operational effectiveness while maintaining system reliability. The key off-the-shelf components and custom-fabricated elements utilized in the system are detailed in the *Prototype Setup* subsection. The rendered design image of the final product is presented in Figure 3.2 to show how the mobile base achieves compact functionality. This platform design functions as a base to implement adaptive PID-based motion control algorithms in the following sections which address the dynamic uncertainties from mecanum wheels.

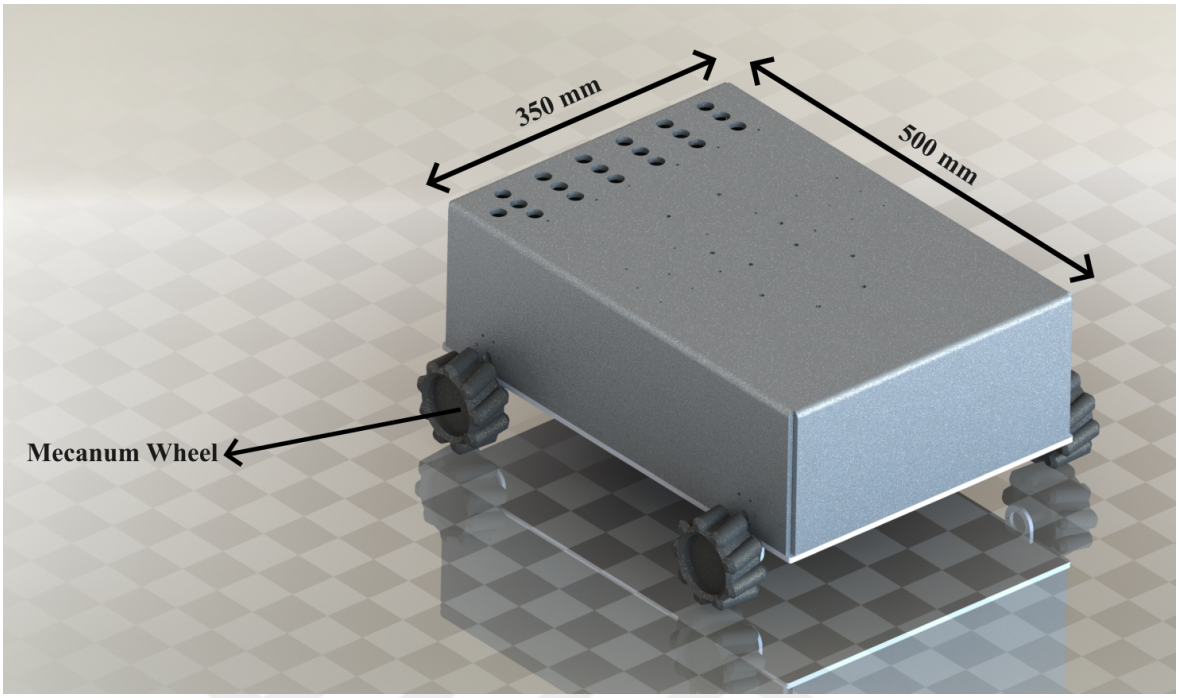

Figure 3.2. CAD model of the mecanum base.

3.1.2. Link 1 and Joint 1

A 6-DOF industrial manipulator with a spherical wrist consists of three positioning joints followed by a wrist with three intersecting rotational axes which allows full orientation control of the end-effector within its workspace. The first joint and link of the 6-DOF manipulator serve as the base elements of the entire arm structure which experience the most severe bending moments when in operation. The design of joint 1 focused on precision and structural robustness to achieve both rigidity and long-term durability. The isometric rendered view of the joint assembly appears in subfigure (a) of Figure 3.3 while subfigure (b) displays the timing-belt-driven transmission mechanism from a bottom perspective. The belt and pulley system enables joint 1 to rotate while providing backdrivability and lower backlash compared to traditional gear systems.

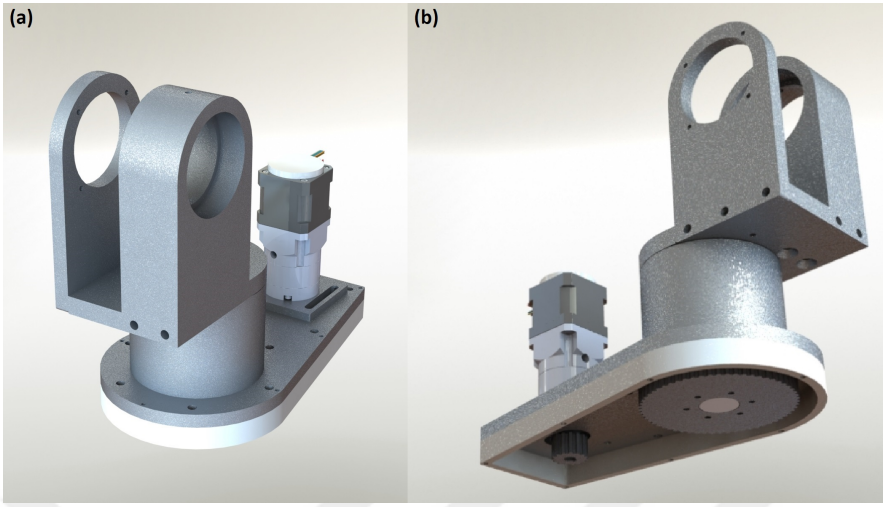

Figure 3.3. (a)CAD model of the mecanum base (b)belt drive bottom view.

Joint 1 uses two tapered roller bearings that oppose each other along with a custom tensioning system to handle substantial mechanical loads and maintain component alignment during stress. The specific bearing arrangement provides improved radial and axial load resistance while maintaining the concentric position of rotating elements. The transparent rendering in Figure 3.4 shows the bearing layout and inner assembly details of the system. The contact surfaces and bearing seats received precise manufacturing tolerances and the moving components maintained a 0.2 mm clearance to reduce friction. The structure underwent optimization for CNC machining from 7075-grade aluminum to achieve the optimal balance between weight, stiffness and manufacturability. The motor driving joint 2 mounts to link 1 through a bearing housing which supports its rotational axis.

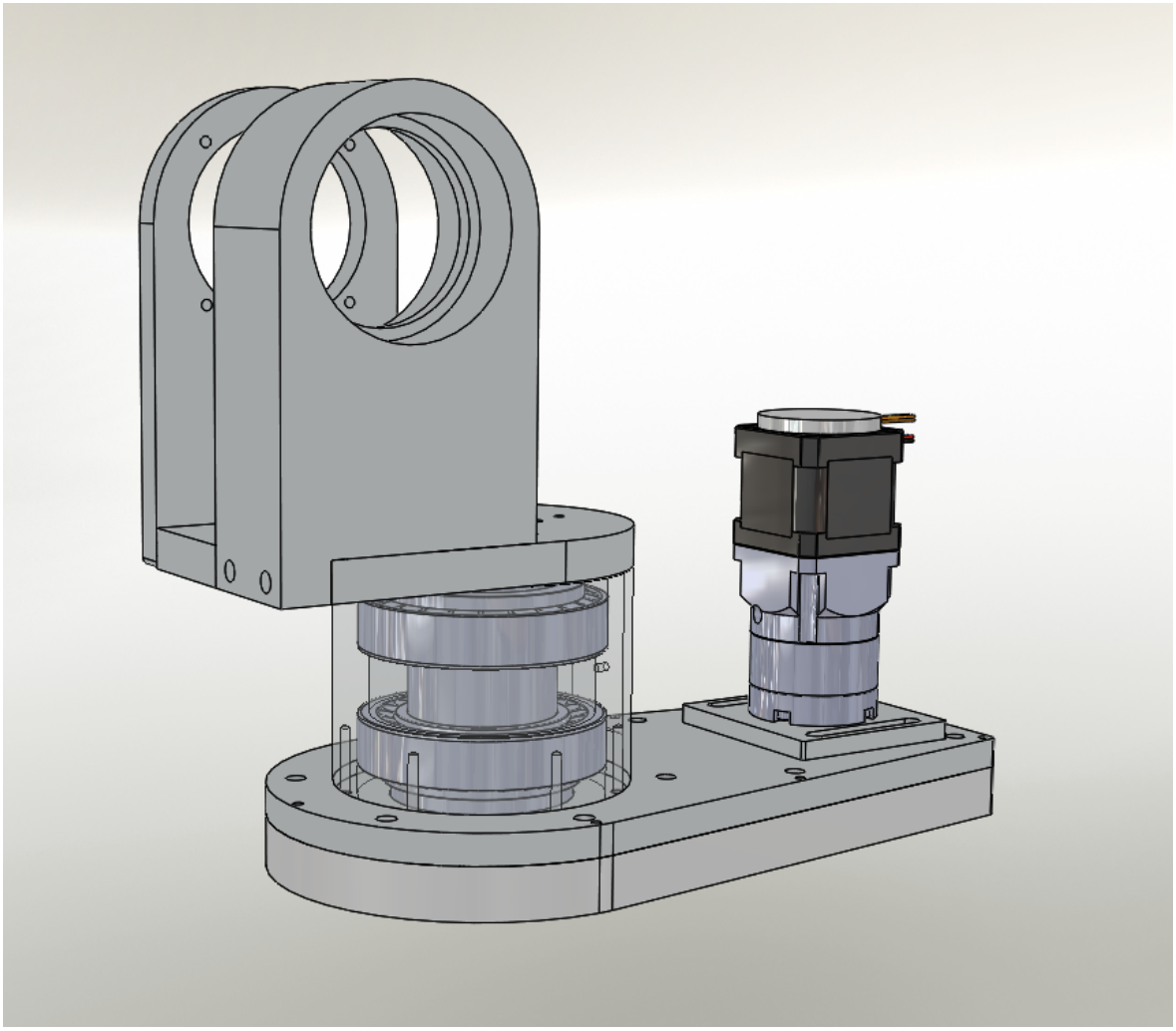

Figure 3.4. Transparent CAD modeling of the first manipulator link.

3.1.3. Link 2 and Joint 2

The design philosophy from the first joint was applied to link 2 and joint 2 for developing structures that maintain stiffness and precise alignment under high dynamic loads. The main rotational axis of vertical articulation joint 2 faces substantial bending and torsional moments. A double tapered roller bearing configuration was implemented to provide high radial and axial load support because of the need to minimize deflection and ensure long-term mechanical stability. The internal bearing layout and timing belt that drives joint 3 are easily observable in Figure 3.5. The figure demonstrates how the motor for joint 3 actuation is integrated and placed near the link's central axis.

Joint 3 motor received placement near Link 2's rotation center to reduce parasitic torque effects while maintaining manipulator compactness. The engineers selected this specific location to stop the creation of long moment arms which would decrease control performance and create excessive stress on previous joints. The Joint 3 actuation system uses a timing belt transmission with 1:1 drive ratio and a 5M-pitch fiber-reinforced belt to achieve smooth torque delivery while minimizing backlash. The solution provides both smooth torque delivery and enables motor decoupling from the joint's rotational axis which helps maintain a slim mechanical profile. The render in Figure 3.5 displays both pulley layout and bearing housings to show the complete mechanical design of the compact system.

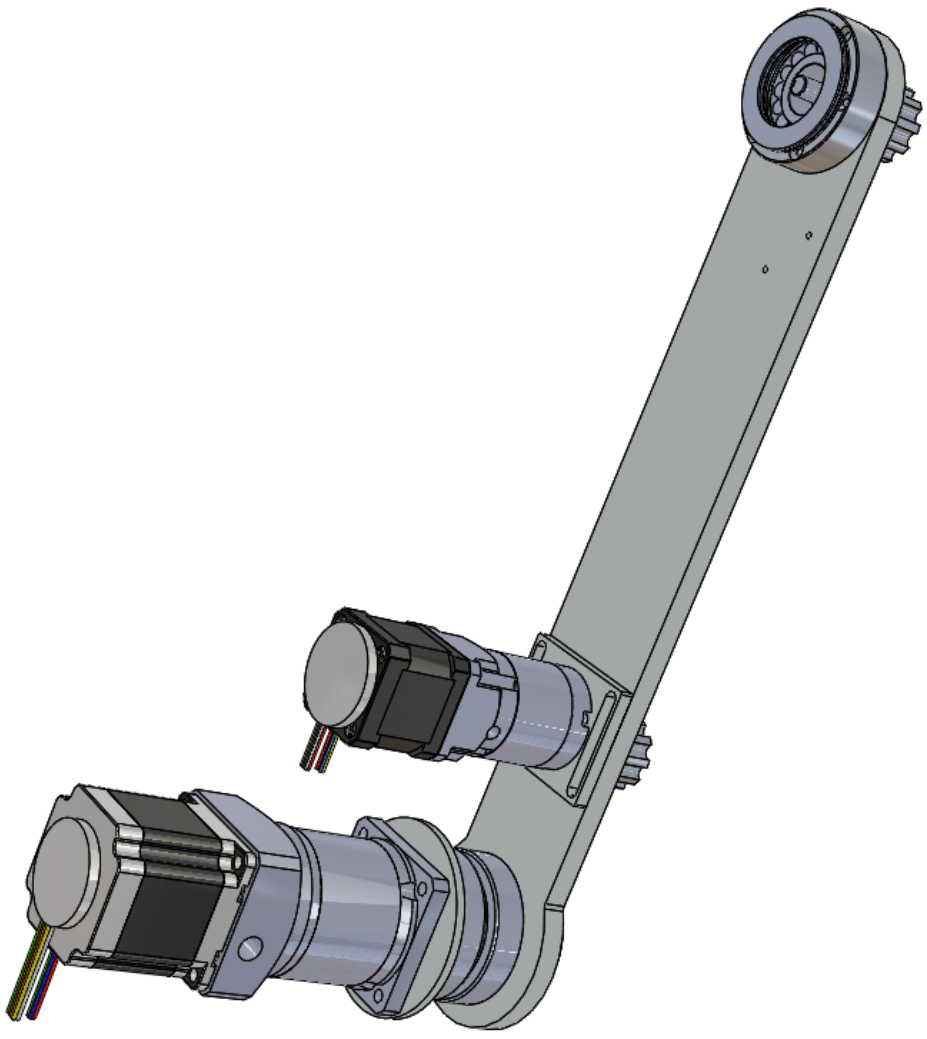

Figure 3.5. CAD modeling of the second manipulator link.

3.1.4. Link 3 and Joint 3

The design of the third joint and link of the manipulator followed the established principles for structure and manufacturing. The 7075-grade aluminum link 3 was optimized for CNC manufacturing to achieve strength and machinability and weight optimization for combined loads from both upstream and downstream joints. The Figure 3.6 shows joint 3 with its rotary bearing elements and structural support for joint 4 actuation components. The design achieves dual functionality through separate bearing assemblies for each joint that enable free and interference-free rotation of both axes.

The design employed needle bearings and base thrust bearings to decouple the rotations of joint 3 and joint 4. The configuration enables joint 4 to move independently from joint 3 rotation which is necessary for compound movement tasks. The motor for joint 4 actuation is mounted on link 3 and its output is transmitted through a small backlash-reduced mechanism which is not depicted in full detail. The bearing arrangement together with the mechanical framework can be seen in the rendered model Figure 3.6 which demonstrates how multiple degrees of freedom fit inside a small mechanical space. The design provides both modularity and reliability because real-time motion control requires dynamic decoupling.

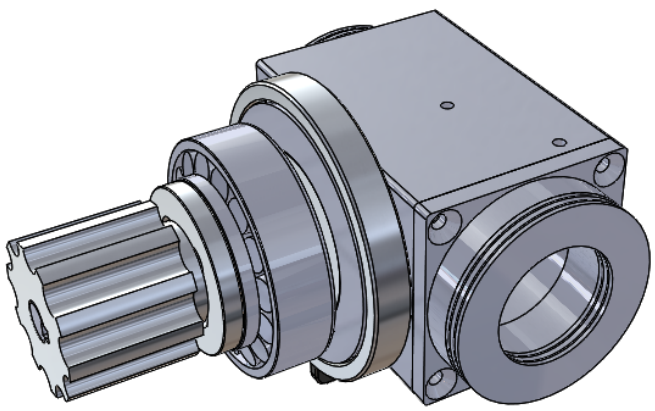

Figure 3.6. CAD modeling of the third manipulator link.

3.1.5. Spherical Wrist and Joints

Joints 4 through 6 form the final stage of the 6-DOF manipulator which creates a spherical wrist with an R–R–R configuration as shown in Figure 3.7. The three joints work independently and in line to grant the end-effector complete orientation freedom across the three cartesian axes. The design complexity of these components prevents separate visual representation so the wrist assembly functions as one unified mechanical system. The red line in the figure shows the rotational center of the spherical wrist which represents the point where the three joint axes meet.

Link 4 (L4) component functions with the needle and thrust bearings embedded in link 3 (L3) as a pure rotational component. The precise operation of joint 4 depends on this setup which creates both a rigid transmission path and eliminates backlash for the subsequent joints. The joint 4 motor located on link 3 provides torque to link 4 which starts the spherical wrist's first rotational axis. The joint 5 motor mounts directly onto link 4 while sharing rotation with it. The joint rotation maintains both the decoupling of joint 5 and the lightweight and balanced design structure.

The implementation of a 1:1 ratio timing belt mechanism with a 5M fiber-reinforced belt serves to drive joint 5. The torque from the joint 5 motor reaches the rotating joint 5 axis through a hollow aluminum tube which contains both a trapezoidal lead screw and a flanged brass nut. The design uses this configuration to meet two essential requirements by moving the spherical wrist's center of mass toward joint 3 rotation axis which reduces the motor torque requirements and enables smaller motor selection while providing self-locking functionality in case of power loss to stop uncontrolled joint movement.

The manipulator terminates with joint 6 which functions as a motor output shaft that provides direct rotation. In the figure, the terminal component link 6 connects directly to the end-effector interface through rigid connections.

The wrist assembly structure consisting of link 4 and rotating tubes and housing brackets uses 7075-grade aluminum for manufacturing because it provides high strength with lightweight properties. Brass material was specifically chosen for Joint 5's sliding nut and bushing assembly because it provides superior wear resistance and low friction when interacting with steel guides. The system features a dedicated belt tensioning mechanism for joint 5 which maintains preload while preventing backlash when the system performs frequent bidirectional motion. The compact design of this spherical wrist structure provides accurate end-effector orientation through its simple mechanical design and robust control features.

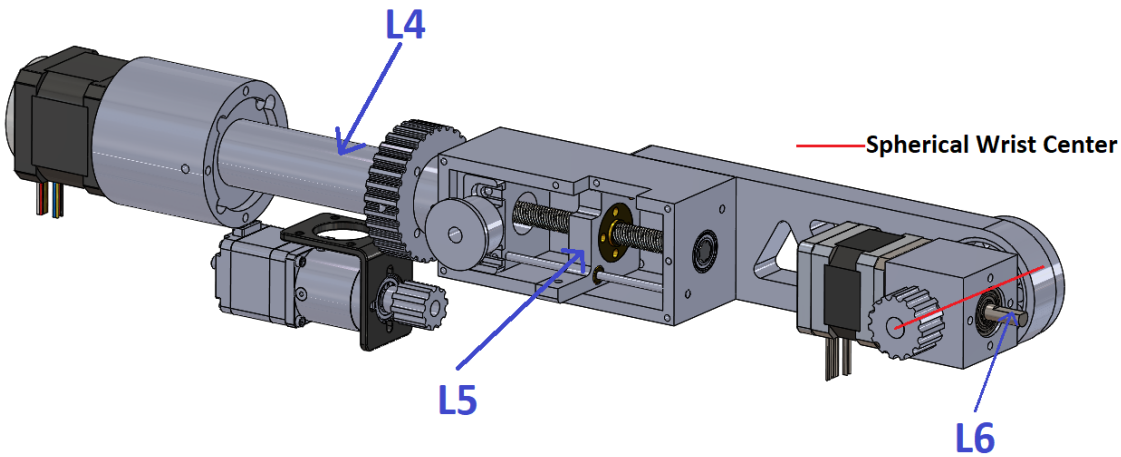

Figure 3.7. CAD modeling of the spherical wrist.

3.2. Manufacturing of the Mobile Manipulator

The production of the mobile manipulator followed a hybrid strategy that combined internal manufacturing capabilities with external precision machining services. We have used in-house 3-axis CNC milling machines to manufacture the majority of functional and structural components. The production included three-dimensional shapes like base mounts together with joint housings and intermediate structural brackets which were designed for simplicity in manufacturing and fixture operations.

The production of rotational and cylindrically symmetric components required CNC turning operations for shafts and couplers and bearing seats. The parts needed CNC lathe operations that exceeded our internal capabilities so we subcontracted them to a foreign manufacturer who specialized in precision turning. The manufacturing process required 5-axis CNC milling for restricted parts that needed compound surface geometries or multiple angle machining. The production of these parts required international outsourcing because it ensured both high precision and strict surface quality standards could be achieved.

Several components required non-traditional manufacturing methods, such as Wire Electrical Discharge Machining (EDM) and Graphite EDM. The EDM process was used to create internal profiles and intricate keyways and sharp inside corners that milling machines alone could not produce. A Chinese vendor specializing in cost-effective high-tolerance micromachining performed all fabrication tasks.

The production of auxiliary plastic components including cable holders and sensor mounts and protective covers utilized Fused Deposition Modeling (FDM) 3D printers located in-house. The design process focused on making the parts lightweight while optimizing print orientation and support reduction and filament flow consistency. Standard mechanical components such as timing belts, pulleys, bearings, bushings, bolts, stepper motors, and encoders were procured from commercial suppliers. The unavailability of closed-loop stepper motors with integrated encoders in local markets forced the purchase of these items from overseas suppliers but other off-the-shelf components came from industrial suppliers in Türkiye.

The mobile base component used 6063 aluminum plates measuring 6 mm in thickness. The panels received their shape through CAD-drawn laser cutting followed by CNC press brake bending to create the required enclosure geometries. After bending the panels, TIG welding was used to weld together the structural corners and seams. Manual grinding and polishing of junctions followed the welding process to achieve surface smoothness and dimensional consistency. Sheet metal processing combined

with high-strength aluminum alloy production allowed the creation of a strong yet lightweight base frame for the Mecanum wheel platform.

The manufacturing approach successfully combined internal capabilities with external expertise to achieve a balance between cost effectiveness and both delivery speed and product quality. All fabrication steps received strict monitoring to maintain alignment with CAD models while special attention focused on tolerance stack-ups at bearing interfaces and actuator mounting points and belt alignment.

3.3. Mathematical Model of the Mobile Manipulator

The mathematical modeling of the mobile manipulator provides analytical support for motion planning and control design. The system modeling process divides into two sections because it contains a mecanum-wheeled omnidirectional base and a 6-DOF industrial robotic arm together. The first part of the modeling process develops kinematic equations for the mecanum base to show its ability to translate and rotate in a plane. The control algorithms benefit from improved accuracy through the inclusion of non-holonomic constraints and wheel-to-ground interactions and slippage effects. The wheel Jacobian-based formulation enables the derivation of forward and inverse kinematics to translate between cartesian velocities of individual wheels and their local cartesian velocities.

The second part of the model uses Denavit–Hartenberg (DH) convention to describe link transformations and joint relations for the manipulator arm. The model contains both forward and inverse kinematics functions which enable the conversion between end-effector pose and joint space configurations. The floating-base manipulator concept is used to integrate the mobile base with the manipulator because the system operates as a single unit. The integrated method becomes vital for coordinated motion control because both base and arm need simultaneous movement during dynamic operations. The mathematical model developed here allows real-time control law implementation and provides the foundation for the adaptive and decoupled PID

controllers that appear later in this thesis.

3.3.1. Forward and Inverse Kinematics of the Mecanum-Wheeled Mobile Base

The mathematical relationship between wheel angular velocities and robot cartesian velocity in the global frame is established by deriving forward and inverse kinematics of the mecanum-wheeled mobile base. The mobile platform contains four mecanum wheels which are installed at the corners of its rectangular chassis. The wheels have a $\pm 45^{\circ}$ angle which enables omnidirectional movement through rotational motion conversion into lateral or diagonal translation. The robot can move in any direction and rotate at the same time because of this design which enables excellent navigation in restricted spaces as shown in Figure 3.8.

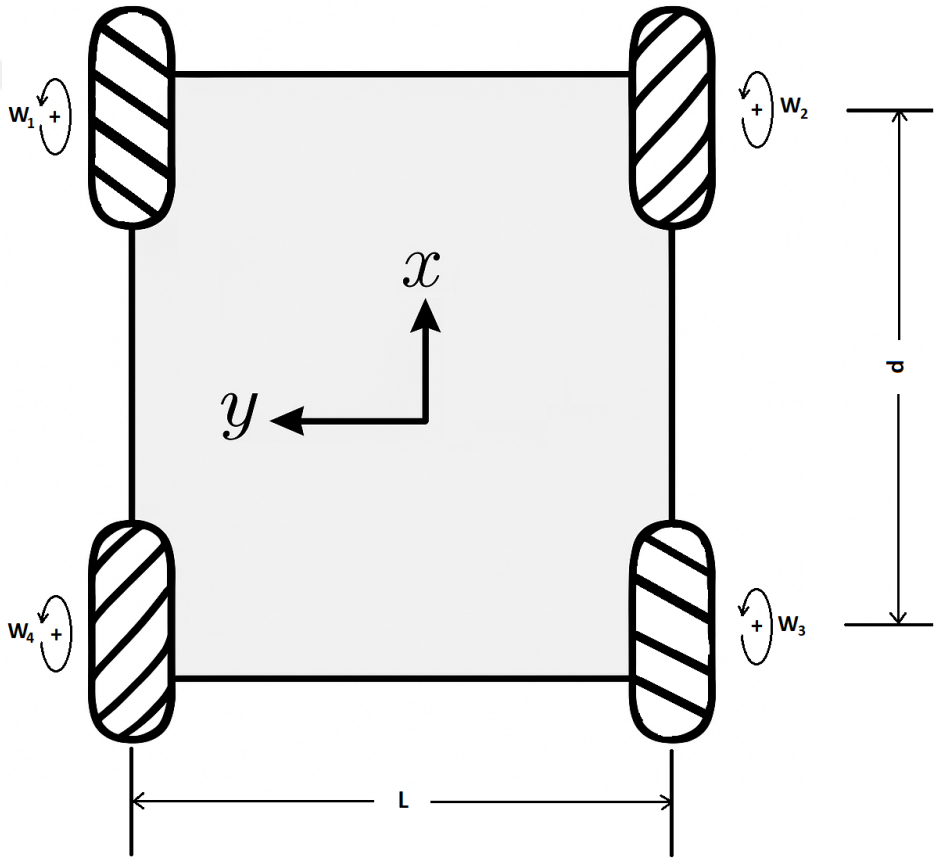

Figure 3.8. Top-down schematic of the mecanum-wheeled robot platform.

The kinematic modeling approach in this study uses the methodology from [21] but with different wheel numbering than the original reference. The wheel indices follow the sequence 1–2–3–4 in a clockwise direction from the front-left wheel. All matrix formulations and velocity transformations in this section take into account the different wheel numbering system.

The forward kinematic formulation given in Equation 3.1 depends on identical wheel geometry and symmetrical wheel placement relative to the robot center [22]. The wheel radius is denoted by r and the wheel axis distances from center are denoted by L for lateral direction and d for longitudinal direction. Using these parameters, a matrix J is constructed to relate the wheel speeds $\dot{\theta}_i$ to the robot's body velocities $\dot{q} = [\dot{x}, \dot{y}, \dot{\phi}]^T$ in the global frame via a rotation matrix $Q(\phi)$:

$$\omega = JQ^T(\phi)\dot{q}, \tag{3.1}$$

The matrix J is defined as

$$J = \frac{1}{r}\begin{bmatrix} 1 & -1 & \frac{-(L+d)}{2} \\ 1 & 1 & \frac{(L+d)}{2} \\ 1 & -1 & \frac{(L+d)}{2} \\ 1 & 1 & \frac{-(L+d)}{2} \end{bmatrix}. \tag{3.2}$$

The rotation matrix $Q(\phi)$ transforms velocity vectors from the local to the global frame

$$Q(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.3}$$

Transformation between $\dot{q}$ in the global frame and $\dot{q}_R$ in the local frame is defined as

$$\dot{q} = Q(\phi)\dot{q}_R. \tag{3.4}$$

The required wheel speeds for platform velocity can be calculated through the inverse kinematics process by solving Equation 3.1. The Moore–Penrose pseudoinverse method computes the inverse of non-square matrix J. This method solves overdetermined systems by finding the least-squares solution that minimizes errors in the system:

$$\dot{q} = Q(\phi)(J^T J)^{-1} J^T \omega. \tag{3.5}$$

The mathematical framework described here allows precise and efficient transformations between actuator commands and cartesian motion which serves as the basis for control algorithms in subsequent sections.

3.3.2. Forward and Inverse Kinematics of the 6 DOF Manipulator

The 6-DOF industrial manipulator requires accurate position and orientation determination of its end-effector relative to the robot base through kinematic modeling. The section develops both forward and inverse kinematics to achieve precise control of the arm during complex manipulation tasks. The forward kinematics calculate end-effector positions from known joint angles through the Denavit–Hartenberg (DH) convention and the inverse kinematics determine joint configurations for reaching specific end-effector positions and orientations. These models serve as essential components for trajectory tracking and control law development and mobile base motion integration.

The kinematic formulation receives support from two illustrative figures. The first figure in Figure 3.9 demonstrates the link parameters which serve as the basis for modeling the manipulator through the Denavit–Hartenberg (DH) convention. The transformation between adjacent coordinate frames depends on the link parameters which include length and offset and twist and joint type. The second figure in Figure 3.10 demonstrates the complete coordinate frame assignments for the entire mobile manipulator system. The system has eight degrees of freedom where the first two represent prismatic joints for mobile base translation in x and y directions and the last six represent revolute joints like industrial 6-DOF robotic arms. The unified coordi-

nate framework depicted in the figure enables the development of forward and inverse kinematics for the complete kinematic chain.

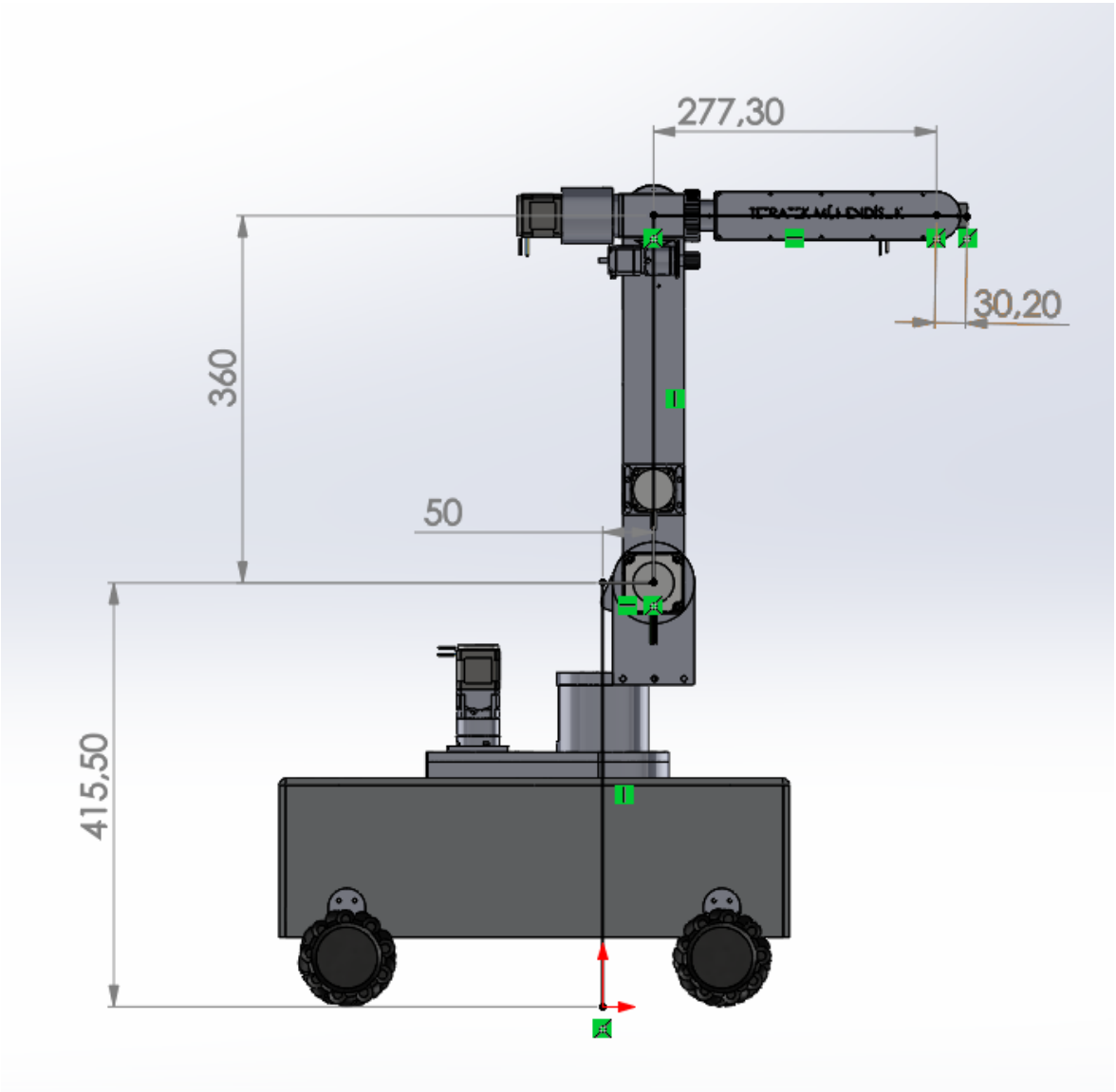

Figure 3.9. Link parameters of the 6-DOF manipulator.

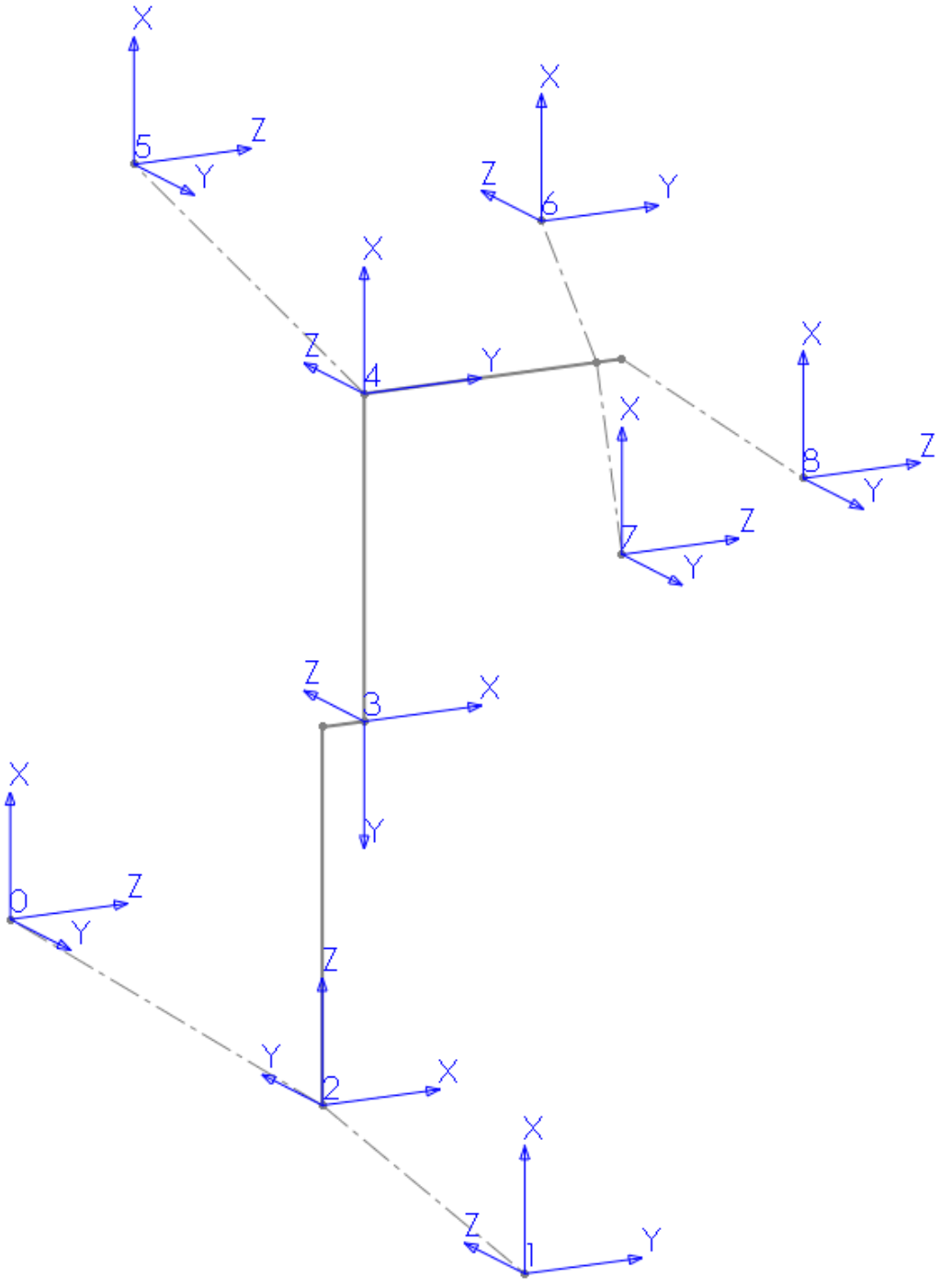

Figure 3.10. Coordinate frame assignment of the full 8-DOF mobile manipulator.

The detailed Denavit–Hartenberg (DH) parameters used in the kinematic modeling of the 6-DOF manipulator are listed in Table 3.1. Each row corresponds to one joint in the kinematic chain and includes the necessary parameters for constructing the

homogeneous transformation matrices between successive links. Notation TL in the below table represents the tool length which is the end effector. We have calculated

Table 3.1. Denavit-Hartenberg parameters.

Links	d(mm)	θ(rad)	r(mm)	α(rad)
1	d_x	0	0	1.57
2	d_y	1.57	50	1.57
3	415.5	θ_3	50	-1.57
4	0	θ_4-1.57	360	0
5	0	θ_5	0	-1.57
6	277.3	θ_6	0	1.57
7	0	θ_7	0	-1.57
8	30.2+TL	θ_8	0	0

transformation matrices using the Denavit-Hartenberg matrix given as

$$T_n^{n-1} = \begin{bmatrix} c\theta_n & -s\theta_n c\alpha_n & s\theta_n s\alpha_n & r_n c\theta_n \\ s\theta_n & c\theta_n c\alpha_n & -c\theta_n s\alpha_n & r_n s\theta_n \\ 0 & s\alpha_n & c\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

Equation 3.7, 3.8 , 3.9 , 3.10, 3.11, 3.12, 3.13, 3.14 shows the Denavit-Hartenberg matrices calculated for each joint.

$$T_1^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_x \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.7}$$

$$T_2^1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.8}$$

$$T_3^2 = \begin{bmatrix} c\theta_3 & 0 & -s\theta_3 & 50c\theta_3 \\ s\theta_3 & 0 & c\theta_3 & 50s\theta_3 \\ 0 & -1 & 0 & 415.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.9}$$

$$T_4^3 = \begin{bmatrix} c(\theta_4 - 90) & -s(\theta_4 - 90) & 0 & 360c(\theta_4 - 90) \\ s(\theta_4 - 90) & c(\theta_4 - 90) & 0 & 360s(\theta_4 - 90) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.10}$$

$$T_5^4 = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 & 0 \\ s\theta_5 & 0 & c\theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.11}$$

$$T_6^5 = \begin{bmatrix} c\theta_6 & 0 & s\theta_6 & 0 \\ s\theta_6 & 0 & -c\theta_6 & 0 \\ 0 & 1 & 0 & 277.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.12}$$

$$T_7^6 = \begin{bmatrix} c\theta_7 & 0 & -s\theta_7 & 0 \\ s\theta_7 & 0 & c\theta_7 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.13}$$

$$T_8^7 = \begin{bmatrix} c\theta_8 & -s\theta_8 & 0 & 0 \\ s\theta_8 & c\theta_8 & 0 & 0 \\ 0 & 0 & 1 & 30.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.14}$$

Equation 3.15 have been used to find the transformation between the end effector of the manipulator and the ground. Also this equation can be used to extract position and rotation matrices.

$$T_8^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T_7^6 T_8^7, \tag{3.15}$$

Most industrial robotic arms use their first three joints to establish end-effector position while their last three joints control end-effector orientation. The architectural design of the system makes forward and inverse kinematics calculations easier because it separates motion translation from rotation. A spherical wrist configuration represents the final three joints because it offers roll-pitch-yaw rotational freedom which maintains the wrist center position. The robot achieves end-effector pose goals by first placing the wrist center at the intersection of the three wrist joint axes before using rotational movements about those axes to adjust orientation shown in Figure 3.11.

The decoupling method in kinematic modeling provides an organized solution for the inverse kinematics problem. The procedure consists of two essential operational phases. The first step in the process determines the wrist center position through a calculation that involves subtracting the orientation-dependent offset (usually the final link length) from the end-effector position. The position-based inverse kinematics problem can be solved independently for the first three joints after the end-effector position has been isolated from other contributions. The required rotations for the spherical wrist joints become calculable by determining the difference between the desired end-effector orientation and the orientation produced by the first three joints. The stepwise solution method enhances both computational efficiency and numerical stability when

solving the inverse kinematics problems of advanced industrial manipulators.

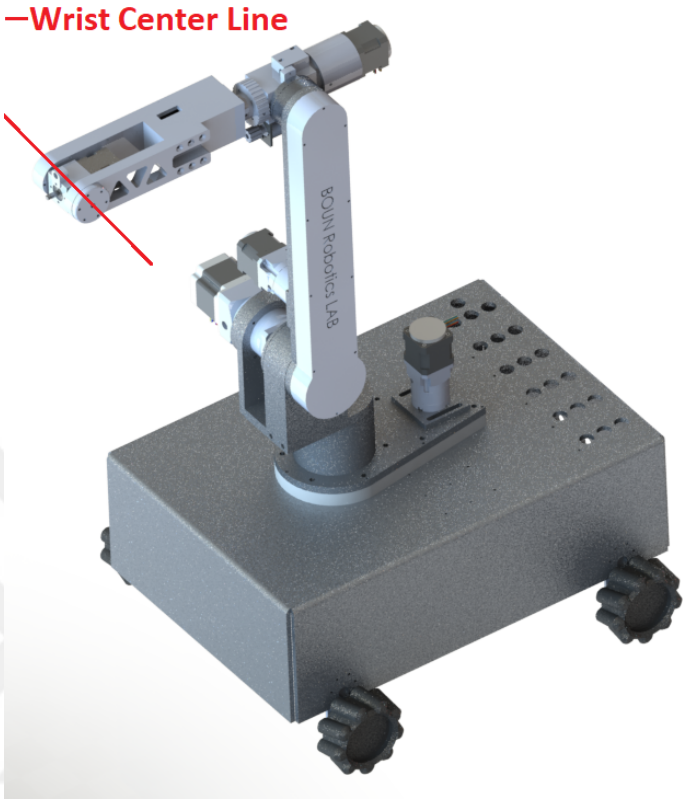

Figure 3.11. Illustration of the wrist center in our 6-DOF industrial manipulator.

The structure of Equation 3.15 is given below, where r represents the elements of the rotation matrix and p denotes the components of the translation vector.

$$T = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \tag{3.16}$$

Forward kinematics Equations 3.17, 3.18, 3.19 can be used to extract the individual rotation angle for each axis. Equation 3.20, 3.21, 3.22 shows the translation of the end effector.

$$R_X(\psi) = \text{atan2}\left(\frac{T_8^0(3,2)}{\cos\phi}, \frac{T_8^0(3,3)}{\cos\phi}\right), \tag{3.17}$$

$$R_Y(\phi) = \text{atan2}\left(-T_8^0(3,1),\ \sqrt{T_8^0(1,1)^2 + T_8^0(2,1)^2}\right), \tag{3.18}$$

$$R_Z(\theta) = \text{atan2}\left(\frac{T_8^0(2,1)}{\cos\phi},\ \frac{T_8^0(1,1)}{\cos\phi}\right), \tag{3.19}$$

$$T_x = T_8^0(1,4), \tag{3.20}$$

$$T_y = T_8^0(2,4), \tag{3.21}$$

$$T_z = T_8^0(3,4), \tag{3.22}$$

Inverse kinematics of the manipulator can be found using the transformation matrix between 8. and 2. coordinate system. Since the movement of the carrier and the manipulator will be synchronized, it is better to evaluate both inverse kinematics separately. Transforming the movement of the carrier to the coordinate system 2 makes the inverse kinematics more manageable. Equation 3.23 shows the transformation matrix of the manipulator. The first 3 joints of the manipulator determine the wrist position and the last 3 joints determine the rotation of the end effector. The wrist center is located at the center of coordinated system 6 and coordinate system 7 shown in Figure 3.11 and Figure 3.10. In inverse kinematics, the position of the end effector and the rotation angles are known. We seek the joint variables as an outcome.

$$T_8^2 = T_3^2 T_4^3 T_5^4 T_6^5 T_7^6 T_8^7, \tag{3.23}$$

Equation 3.23 shows the desired position and rotation vector with respect to coordinate system 2. Subscription e added short for end effector in Equation 3.24.

$$\begin{bmatrix} X_e & Y_e & Z_e & \psi_e & \phi_e & \theta_e \end{bmatrix}^T. \tag{3.24}$$

Substituting Equation 3.24 into Equation 3.23 gives the following result

$$T_8^2 = \begin{bmatrix} c\theta_e c\phi_e & c\theta_e s\phi_e s\psi_e - s\theta_e c\psi_e & c\theta_e s\phi_e c\psi_e + s\theta_e s\psi_e & X_e \\ s\theta_e c\phi_e & s\theta_e s\phi_e s\psi_e + c\theta_e c\psi_e & s\theta_e s\phi_e c\psi_e - c\theta_e s\psi_e & Y_e \\ -s\phi_e & c\phi_e s\psi_e & c\phi_e c\psi_e & Z_e \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.25)$$

The wrist transformation matrix can be found using Equation 3.14 when $\theta_8 = 0$, 3.25 and 3.26. As mentioned above, T_7^2 is the center of the wrist and θ_3, θ_4 and θ_5 determine the translation of the center of the wrist.

$$T_7^2 = T_8^2[T_8^7]^{-1}, \quad (3.26)$$

The geometric approach has been used to calculate the first tree joint angles of the manipulator. Geometric approach schematic has been shown in Figure 3.12. In our case, $a_3 = 0$ as there is no offset in out design. Equation 3.27 shows calculated θ_3 angle for the desired position of the manipulator using Equation 3.26.

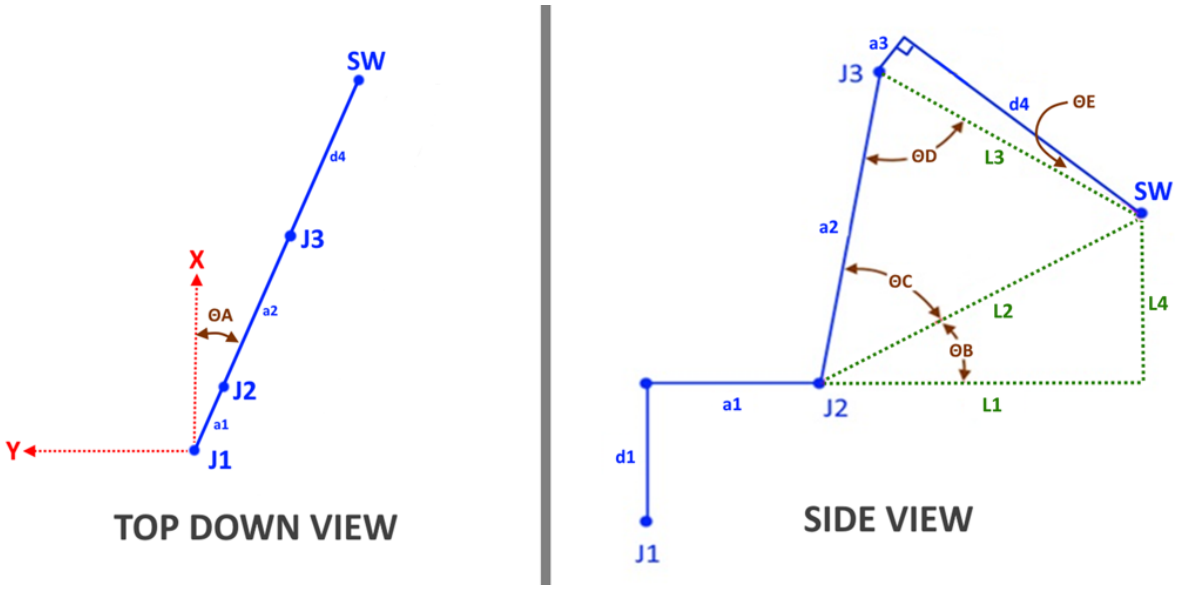

Figure 3.12. Geometric approach schematic.

$$\theta_3(\theta_A) = atan(\frac{T_7^2(2,4)}{T_7^2(1,4)}). \tag{3.27}$$

Figure 3.12 side view shows the manipulator at $\theta_3 = 0$ position. For simplicity, we have projected wrist position to same plane using Equation 3.28 and 3.29.

$$[T_7^2(1,4)]' = T_7^2(1,4)\cos\theta_3 - T_7^2(2,4)\sin\theta_3, \tag{3.28}$$

$$[T_7^2(2,4)]' = T_7^2(2,4)\cos\theta_3 + T_7^2(1,4)\sin\theta_3. \tag{3.29}$$

We have calculated the geometric parameters shown in Figure 3.12 using 3.28, 3.29 and Table 3.1 and found the following expressions

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ \theta B \\ \theta C \\ \theta D \end{bmatrix} = \begin{bmatrix} |[T_7^2(1,4)]' - 50| \\ \sqrt{L_1^2 + L_4^2} \\ 277.3 \\ T_7^2(3,4) - 415.5 \\ atan(\frac{L_1}{L_4}) \\ acos\frac{360^2 + L_2^2 - L_3^2}{2L_2 360} \\ acos\frac{360^2 + L_3^2 - L_2^2}{2L_3 360} \end{bmatrix}. \tag{3.30}$$

Equation 3.30 shows geometric parameters to solve the inverse kinematic problem for θ_4 and θ_5. Solution to inverse kinematic θ_4 and θ_5 can be found using Equation 3.31 and 3.32

$$\theta_4 = \begin{cases} \theta C - \theta B, & \text{if } [T_7^2(1,4)]' > 50 \; and \; L_4 > 0 \\ \theta C - \theta B - 180, & \text{if } [T_7^2(1,4)]' > 50 \; and \; L_4 \leq 0 \\ \theta C + \theta B, & \text{if } [T_7^2(1,4)]' \leq 50 \end{cases}, \tag{3.31}$$

$$\theta_5 = \theta D. \tag{3.32}$$

We can calculate T_8^2 using desired position and orientation values using Equation 3.25. We can also calculate T_3^8, T_4^3 and T_5^4 by substituting Equation 3.27, 3.31 and 3.32 into Equation 3.9, 3.10 and 3.11 respectively to find T_5^2. We can calculate wrist rotation matrix using homogeneous transformation matrices as shown in Equation 3.33. Top left 3x3 matrix of the T_8^5 matrix defines rotation of the spherical wrist. Thus, θ_6, θ_7 and θ_8 can be extracted from Equation 3.33 using Equation 3.34, 3.35 and 3.36.

$$T_8^5 = [T_5^2]^T \, T_8^2 = [T_5^2]^T \, T_5^2 \, T_8^5, \tag{3.33}$$

$$\theta_7 = \text{atan2}\left(T_8^5(3,3), \mp\sqrt{1 - T_8^5(3,3)^2} \right), \tag{3.34}$$

$$\theta_6 = \begin{cases} \text{atan2}(T_8^5(1,3), T_8^5(2,3)), & \text{if } \theta_7 > 0 \\ \text{atan2}(-T_8^5(1,3), -T_8^5(2,3)), & \text{if } \theta_7 \leq 0 \end{cases}, \tag{3.35}$$

$$\theta_8 = \begin{cases} \text{atan2}(-T_8^5(3,1), T_8^5(3,2)), & \text{if } \theta_7 > 0 \\ \text{atan2}(T_8^5(3,1), -T_8^5(3,2)), & \text{if } \theta_7 \leq 0 \end{cases}, \tag{3.36}$$

Synchronized motion of the carrier and the manipulator requires velocity matching. Thus, velocity kinematics for the mobile manipulator has been studied. Equation 3.37 shows how the related link i Jacobian matrix column have been calculated. Equation 3.38 shows full form of the jacobian matrix for 8 DOF system.

$$J_i = \begin{bmatrix} z_{i-1}^0 \times (o_n^0 - o_{i-1}^0) \\ z_{i-1}^0 \end{bmatrix}, \tag{3.37}$$

$$J = \begin{bmatrix} J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 & J_8 \end{bmatrix}, \tag{3.38}$$

Equation 3.39 shows how to calculate z_i^0 unit vectors.

$$z_{i-1}^0 = R_{i-1}^0 k, \tag{3.39}$$

Equation 3.40 shows how to calculate o_i^0 coordinates of the origins of the coordinate systems shown in Figure 3.10.

$$o_i^0 = \begin{bmatrix} T_i^0(1,4) \\ T_i^0(2,4) \\ T_i^0(3,4) \end{bmatrix}, \tag{3.40}$$

The first two columns of the Jacobian matrix represent the velocity of the carrier. Since it moves omnidirectional, the last three rows of the Jacobian matrix become all zeros. Columns 3-8 represent the velocity properties of the manipulator. Equation 3.41 shows the relation between the joint variable change rate and the linear velocity of the end effector and the angular velocity.

$$\begin{bmatrix} v_x & v_y & v_z & w_x & w_y & w_z \end{bmatrix}^T = J \begin{bmatrix} \dot{d}_x & \dot{d}_y & \dot{\theta}_3 & \dot{\theta}_4 & \dot{\theta}_5 & \dot{\theta}_6 & \dot{\theta}_7 & \dot{\theta}_8 \end{bmatrix}^T, \tag{3.41}$$

3.3.3. Velocity Matching

In mobile manipulators, ensuring velocity matching between the carrier platform and the end effector of the manipulator is crucial to efficient and precise task execution while in motion. When the carrier moves, the manipulator must dynamically adjust its velocity to compensate for the platform's movement, preventing unintended positional errors and ensuring smooth trajectory tracking. This synchronization is especially important in tasks such as pick-and-place, plasma cutting or collaborative operations, where misalignment due to velocity mismatches can lead to failed positioning of the

end effector, increased cycle time, or even damage to objects. Proper coordination between the carrier and the end effector improves stability, reduces the computational load on motion planning, and improves overall task efficiency, allowing for seamless interaction with the environment without unnecessary stops. Velocity values of the carrier platform and end effector have been calculated using Equation 3.42 and 3.43 respectively. Superscription "c" represents carrier and "e" represents end effector in Equation 3.42 and 3.43.

$$\begin{bmatrix} v_x^c & v_y^c & v_z^c & w_x^c & w_y^c & w_z^c \end{bmatrix}^T = J \begin{bmatrix} \dot{d}_x & \dot{d}_y & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{3.42}$$

$$\begin{bmatrix} v_x^e & v_y^e & v_z^e & w_x^e & w_y^e & w_z^e \end{bmatrix}^T = J \begin{bmatrix} 0 & 0 & \dot{\theta}_3 & \dot{\theta}_4 & \dot{\theta}_5 & \dot{\theta}_6 & \dot{\theta}_7 & \dot{\theta}_8 \end{bmatrix}^T. \tag{3.43}$$

For the end effector to remain motionless, path planning should obey the criteria defined in Equation 3.44. Meeting this criteria also synchronize the position of the end-effector and the carrier.

$$\begin{bmatrix} v_x^e & v_y^e & v_z^e \end{bmatrix}^T = - \begin{bmatrix} v_x^c & v_y^c & v_z^c \end{bmatrix}^T. \tag{3.44}$$

3.4. Controller Design

The chapter explains the control architecture of the mobile manipulator system which includes both the mobile base and the robotic arm. The controller design achieves precise and stable motion in real-time operations by handling system nonlinearities together with disturbances and model uncertainties. The mecanum-wheeled mobile base employs adaptive PID control known as time-varying PID to improve tracking performance despite slippage and uneven surface interactions. The 6-DOF manipulator operates with independent PID controllers for each joint which require separate tuning for torque and position regulation to produce responsive yet smooth arm movements. The control framework functions through independent modules which enable position

and orientation control while preserving system robustness and scalability. The following sections explain the mathematical approach and implementation steps and tuning procedures for developing these controllers.

3.4.1. Mecanum Wheeled Mobile Robot Controller

The control strategy for the mecanum-wheeled mobile base is presented in this section. The controller design uses a kinematic error model to describe the nonlinear relationship between desired and actual trajectories in the global coordinate frame. The nonlinear error model is first derived in terms of position and orientation errors to facilitate controller synthesis. The resulting expression is then linearized around the operating point using a first-order Taylor expansion, enabling the application of classical control techniques. An adaptive PID control law is proposed to minimize tracking error by dynamically adjusting its gains based on real-time error evolution, building upon the linearized model. This time-varying control approach improves robustness against disturbances and unmodeled dynamics such as wheel slippage, ensuring smooth and accurate navigation. Let $q_d = [x_d(t)\ y_d(t)\ \phi_d(t)]^T$ be the desired trajectory of the mecanum platform defined in global coordinate system. The vector $e = q_d(t) - q(t) = [e_x(t)\ e_y(t)\ e_\phi(t)]^T$ describes the position and posture error of the mecanum wheeled mobile platform. Since the calculations for the mecanum robot requires locally defined vector, we need to transform e from global coordinate system to local coordinate system as e_R given as

$$e_R = Q^T(\phi)e. \tag{3.45}$$

The derivative of Equation 3.45 provides information about velocity properties thus wheel speed calculations can be done using following expression

$$\dot{e}_R = \begin{bmatrix} -sin(e_\phi) & cos(e_\phi) & 0 \\ -cos(e_\phi) & -sin(e_\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \Omega\, Q(e_\phi)\, e_R + Q^T(e_\phi)\,(\dot{q}_d - \dot{q}). \tag{3.46}$$

Rearranging Equation 3.46 gives the following result

$$\dot{e}_R = \begin{bmatrix} 0 & \Omega(t) & 0 \\ -\Omega(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e_R + Q^T(e_\phi)\,(\dot{q}_d - \dot{q}). \tag{3.47}$$

Equation 3.47 shows the nonlinear error model of the mobile platform. Linearize this model around $cos(e_\phi) \approx 1$ and $sin(e_\phi) \approx e_\phi$ to get the following linear error model in the local coordinate system.

$$\dot{e}_R = \begin{bmatrix} 0 & \Omega(t) & V_y^d(t) \\ -\Omega(t) & 0 & -V_x^d(t) \\ 0 & 0 & 0 \end{bmatrix} e_R + I_{3x3} \begin{bmatrix} V_x^d - V_x^R \\ V_y^d - V_y^R \\ \Omega^d - \Omega^R \end{bmatrix}, \tag{3.48}$$

Equation 3.48 can be simplified as following expression

$$\dot{e}_R = A e_R + I_{3x3}\, u. \tag{3.49}$$

The control law to reference trajectory is designed as

$$\begin{bmatrix} u_x \\ u_y \\ u_\Omega \end{bmatrix} = \begin{bmatrix} K_P^x e_x^R + K_I^x \int e_x^R\, dt \; + K_D^x \frac{d(e_x^R)}{dt} \\ K_P^y e_y^R + K_I^y \int e_y^R\, dt \; + K_D^y \frac{d(e_y^R)}{dt} \\ K_P^\phi e_\phi^R + K_I^\phi \int e_\phi^R\, dt \; + K_D^\phi \frac{d(e_\phi^R)}{dt} \end{bmatrix}. \tag{3.50}$$

PID control coefficients calculated dynamically dependent on the local error of relevant variable such as $i = x, y, \phi$.

$$\begin{bmatrix} K_P^i \\ K_I^i \\ K_D^i \end{bmatrix} = \begin{bmatrix} a_i^1 + a_i^2 |e_i^R| \\ b_i^1 - b_i^2 |e_i^R| \\ c_i^1 + c_i^2 |e_i^R| \end{bmatrix}, \tag{3.51}$$

The model provides a strong response to trajectory error changes through coefficient adjustments based on absolute error values within local definitions. The main objective

of this controller is to achieve the shortest possible time for reaching the reference trajectory. Figure 3.13 shows the proposed trajectory controller block diagram.

3.4.2. 6 DOF Manipulator Controller Design

The industrial manipulator utilized in this research features a configuration known as RRR–RRR, with all revolute joints. The initial three joints control the end-effector position in Cartesian coordinates, while the final three joints establish a spherical wrist to manage the orientation of the end-effector. Such structural division allows for a control strategy where translational and rotational motions can be managed separately. In this part of the discussion, we use a joint-space PID control structure, where every joint is controlled separately based on position feedback. Error calculation for the joint variables is done by using Equation 3.52, where d denotes the desired value and m denotes the measured value.

$$e_{\theta_i} = \theta_i^d - \theta_i^m, \tag{3.52}$$

In this study, we have focused on the positioning accuracy of the mobile manipulator. For this reason, we have calculated the change rate of the error model defined in Equation 3.53, so that we can compensate for the joint variable speed error as well, to achieve speed matching between the mobile platform and the manipulator.

$$\dot{e}_{\theta_i} = \dot{\theta}_i^d - \dot{\theta}_i^m, \tag{3.53}$$

This simple yet effective method proves itself by performing very accurately, which will be discussed in the results section. Figure 3.14 shows the structure of the controller for each joint variable.

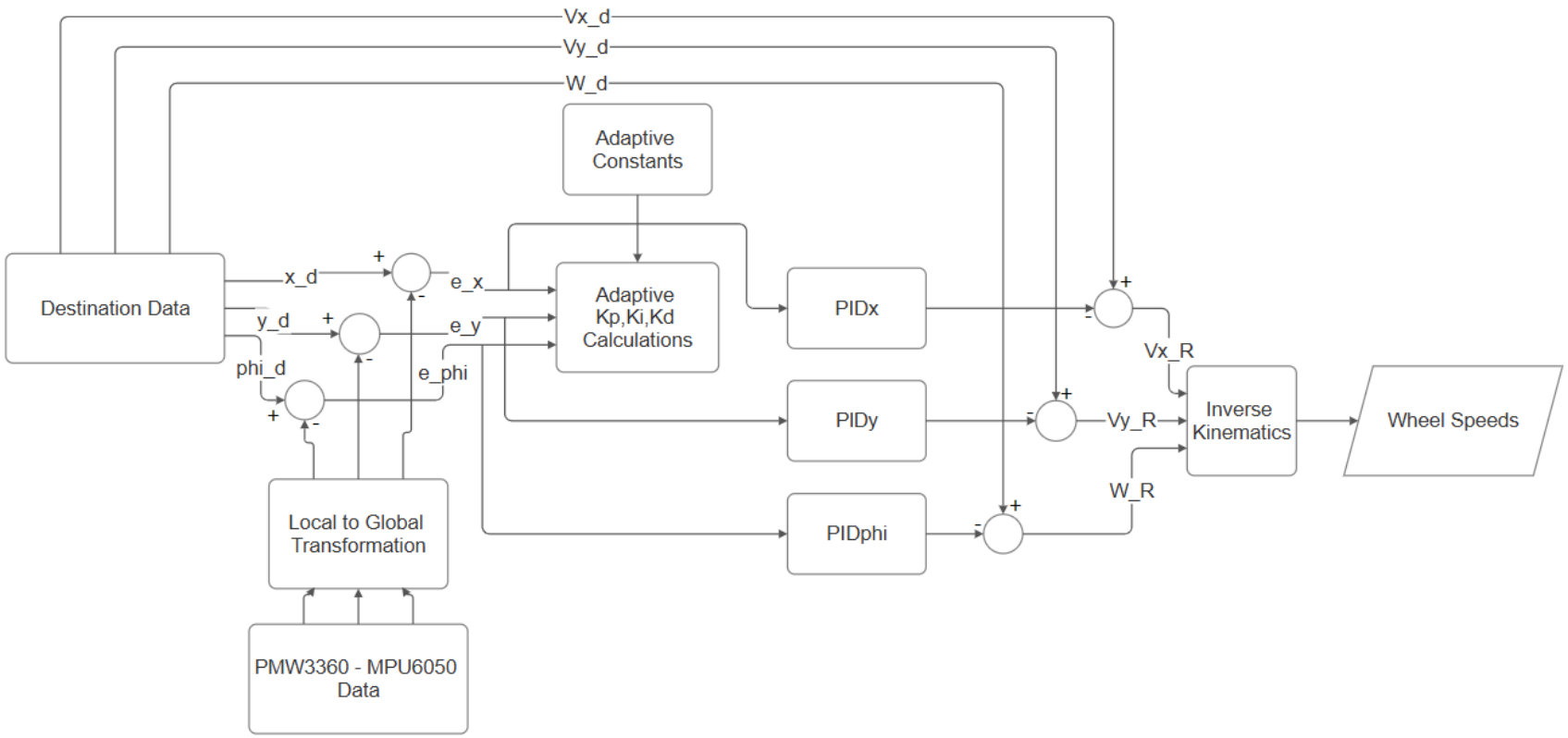

Figure 3.13. Proposed adaptive controller trajectory tracking block diagram.

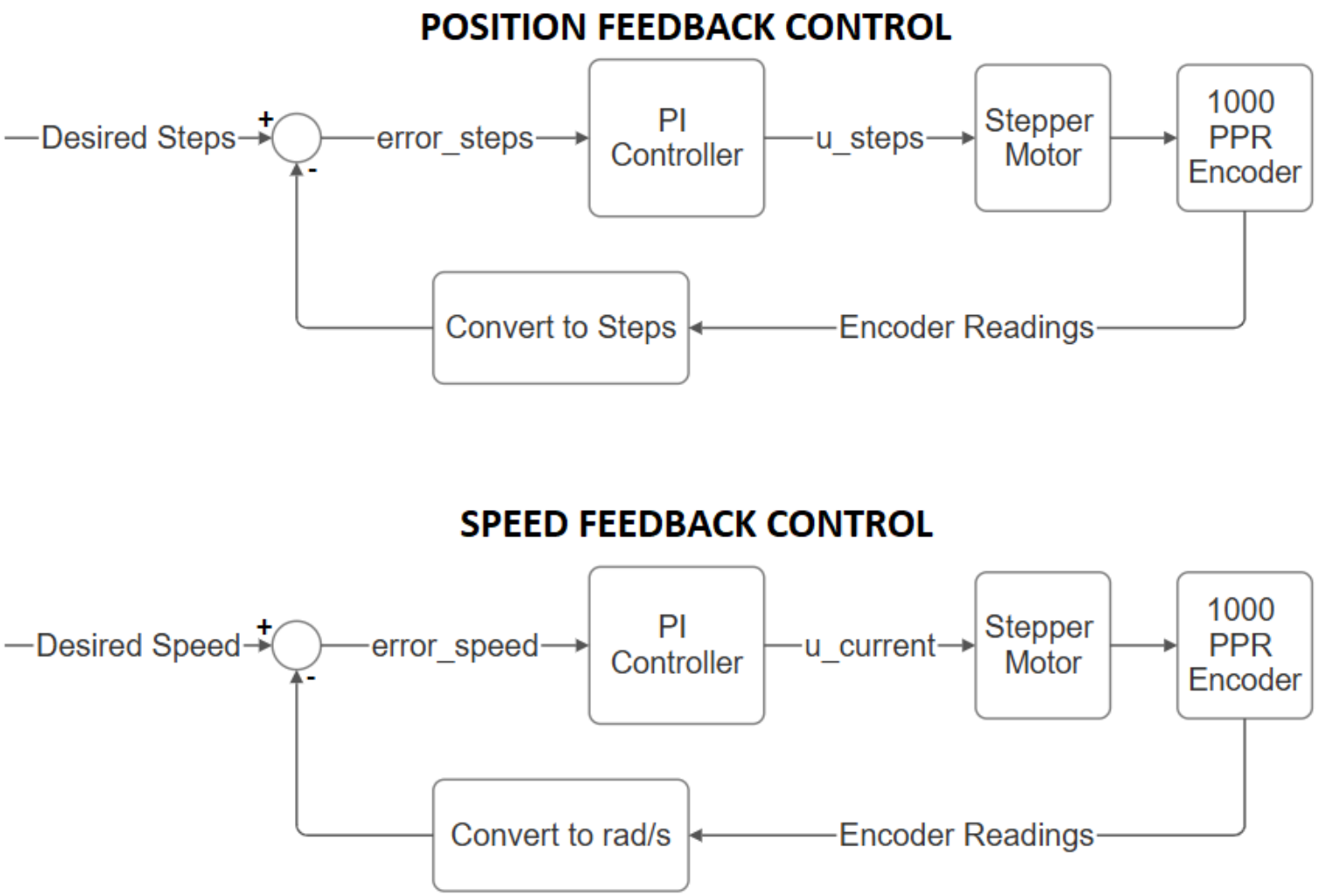

Figure 3.14. Manipulator PID controller block diagram.

4. SIMULATION AND PROTOTYPE SETUP

This chapter presents the complete setup developed for both the physical implementation and the simulation environment of the mobile manipulator system.

4.1. Simulation Setup

MATLAB Simulink and RoboDK provided simulation environments which verified mathematical modeling and controller design before deployment into the real world. Simulink models duplicated the kinematic motion of the mecanum-wheeled base together with the forward kinematics of the full 8-DOF mobile manipulator system. Through simulation the PID parameters of the base velocity controller received multiple iterations of tuning and the combined kinematic transformations between mobile base and industrial manipulator underwent validation tests.

A simulation environment with real-time base motion control functions through an adaptive PID controller runs alongside a pose calculation module for end-effectors based on joint variables. The simulation models match real-world behavior through the addition of randomly generated measurement noise and feedback disturbances in simulated sensor inputs. The controller undergoes thorough testing under unpredictable and imperfect operational conditions through this method. The algorithm becomes better equipped to handle realistic deployment scenarios through the inclusion of stochastic elements that simulate sensor drift and latency.

The two MATLAB Simulink models are shown in Figures 4.1 and 4.2 as representative screenshots. The initial screenshot displays the adaptive PID control system for Mecanum base movement which consists of reference tracking and feedback error calculation and motor speed output. The figure presents the complete forward kinematics structure which computes the end-effector pose of the 8-DOF mobile manipulator through the integration of base and arm systems.

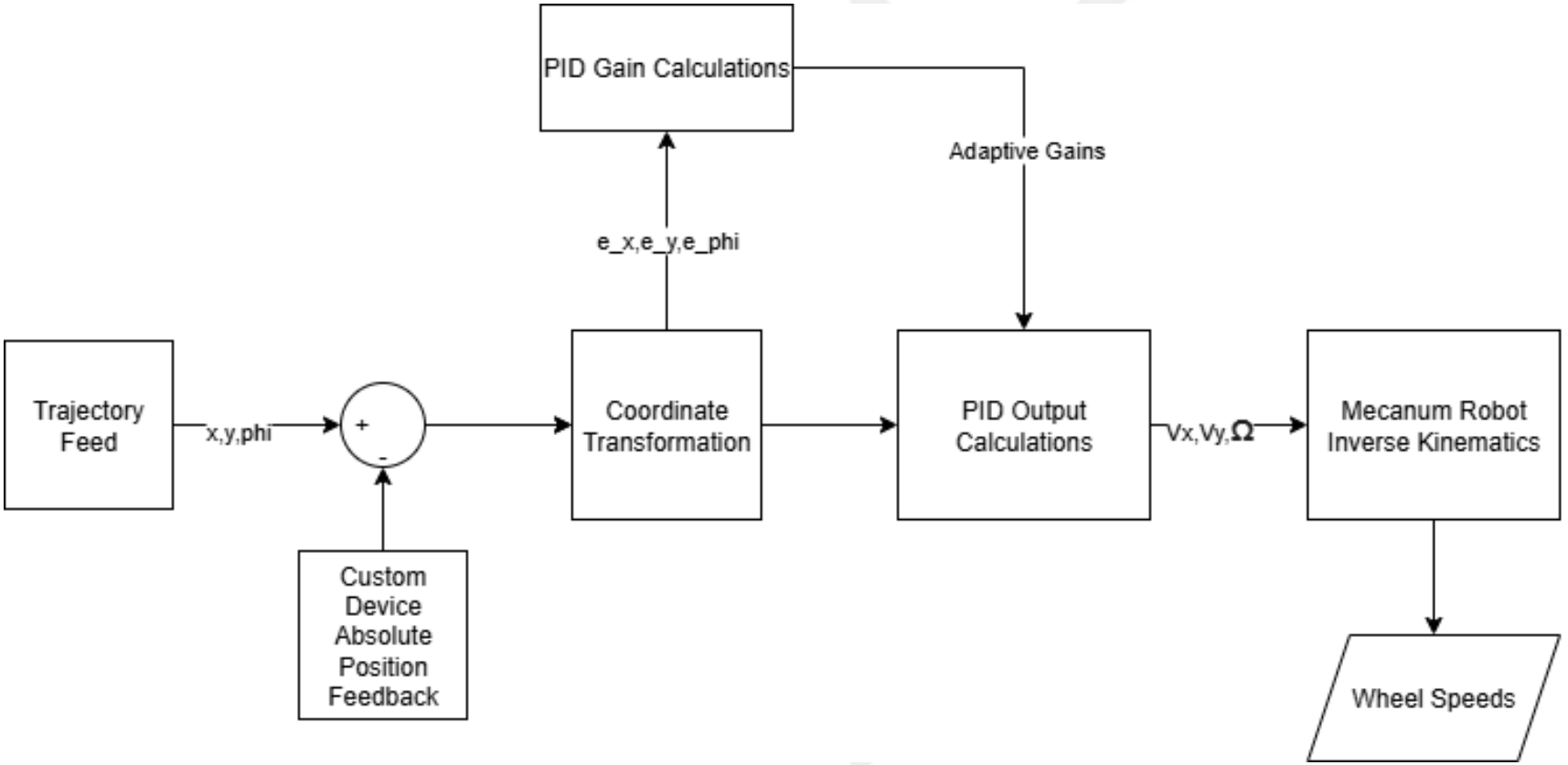

Figure 4.1. Simulink model of the adaptive PID controller for mecanum base.

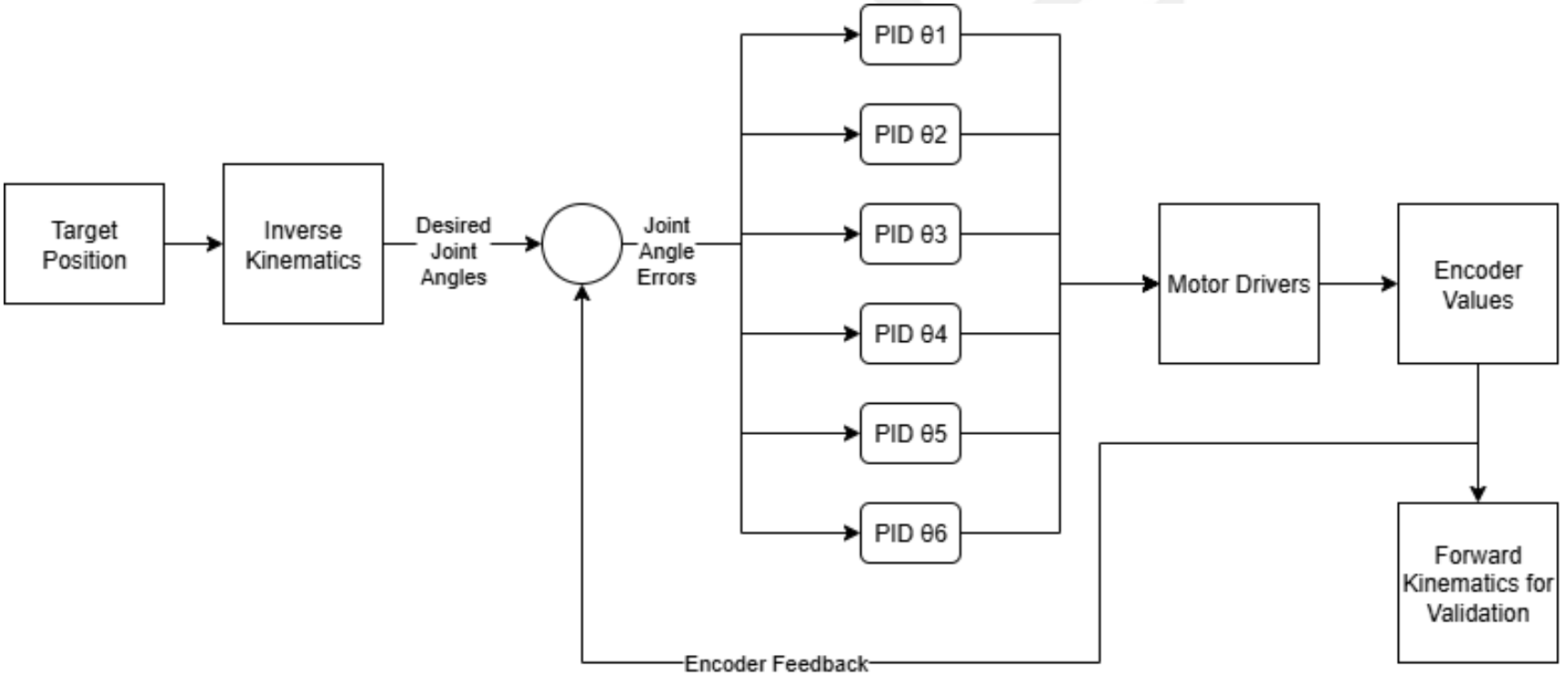

Figure 4.2. Simulink model of forward kinematics for the mobile manipulator.

The 3D visualization platform RoboDK served as a simulation tool to test how the manipulator could reach its workspace boundaries and what motions it could perform. The platform offers users an easy-to-use interface to view robot joint positions and test various target locations within the robot workspace.

The simulation environment functions as both a development tool and validation instrument for testing control methods before hardware deployment. The identical responses between simulated results and actual experiments confirm that the system will deliver reliable precise motion control operations.

4.2. Prototype Setup

The prototype development of the mobile manipulator system demonstrates its mechanical and electronic subsystem development in this section. The hardware components and sensors work together with embedded systems to support the implementation. System reliability and experimental repeatability and modularity receive special attention in the selection of components.

The adaptive PID algorithm performance evaluation for mobile base operation requires a dedicated test platform illustrated in Figure 4.3. The experimental setup employs four NEMA 23 stepper motors with 2.2 Nm torque output that stepper motor drivers control individually. The system uses a 16.8V 2.5Ah battery pack to supply power which supports continuous mobile testing. NRF24L01 transceivers function as wireless data transfer devices between the robot and control interface while providing low-latency data transfer. The control logic operates on an Arduino Mega while the chassis utilizes 20x20 mm aluminum extrusion sigma profiles for its construction to achieve both light weight and rigid structure.

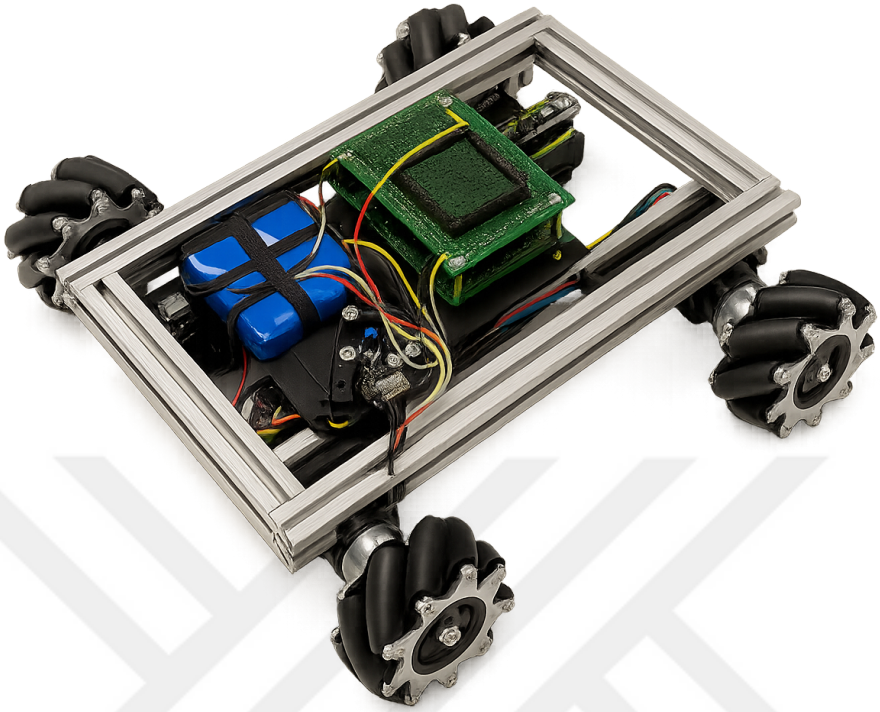

Figure 4.3. Prototype mecanum platfrom.

Prototype mobile manipulator features a 6-DOF robotic arm which operates as a lightweight precision manipulator shown in Figure 4.4. The first joint operates through a NEMA 17 motor which drives a 10:1 planetary gearbox and a 60:15 belt-pulley system. The NEMA 23 motor drives the second joint through a 50:1 planetary gearbox. The third joint uses a NEMA 17 motor which operates through a 50:1 gearbox and 1:1 pulley system. The fourth joint receives its motion from a NEMA 11 motor which uses a 19:1 gearbox. The fifth axis functions through a NEMA 17 motor operated by a ball screw (8 mm lead) which drives a linear carriage linked to the joint via a belt. The sixth and final joint operates through a NEMA 14 motor with a 10:1 gearbox that directly connects to the end-effector interface. The mechanical components undergo T6 heat treatment for strength enhancement after fabrication from 7075-grade aluminum. The system employs SKF high-precision bearings for rotary support applications while brass bushings serve locations that need sliding contact without rolling elements. Complete bill of materials for the prototype mobile manipulator given in Table 4.1

Table 4.1. BOM for prototype mobile manipulator.

Component Description
Mecanum-wheeled mobile base frame AL7075
Custom designed AL7075 part (x98)
Custom designed 3D printed part (x18)
Mecanum wheel (x4)
Nema 23 2.2 NM stepper motors with encoders (x5)
MKS SERVO57D closed-loop motor drivers (x4)
Nema 17 1.1 NM stepper motors with encoders (x3)
Nema 14 1.1 NM stepper motors with encoders (x1)
Nema 11 1.1 NM stepper motors with encoders (x1)
Nema 23 1:50 planetary reducer (x1)
Nema 17 1:50 planetary reducer (x1)
Nema 17 1:10 planetary reducer (x2)
Nema 17 1:8 planetary reducer (x2)
Nema 14 1:10 planetary reducer (x1)
Nema 11 1:19 planetary reducer (x1)
Closed loop stepper motor drivers (x6)
Custom-designed local positioning sensor module
Arduino Mega (x1)
Arduino Nano (x1)
Custom build signal processing PCB (x1)
Teensy 4.1 terminal block (x1)
Teensy 4.1 microcontroller for real-time control
End-effector gripper
RS485 communication interface board
Mechanical limit switch (x4)
Inductive limit switch (x2)

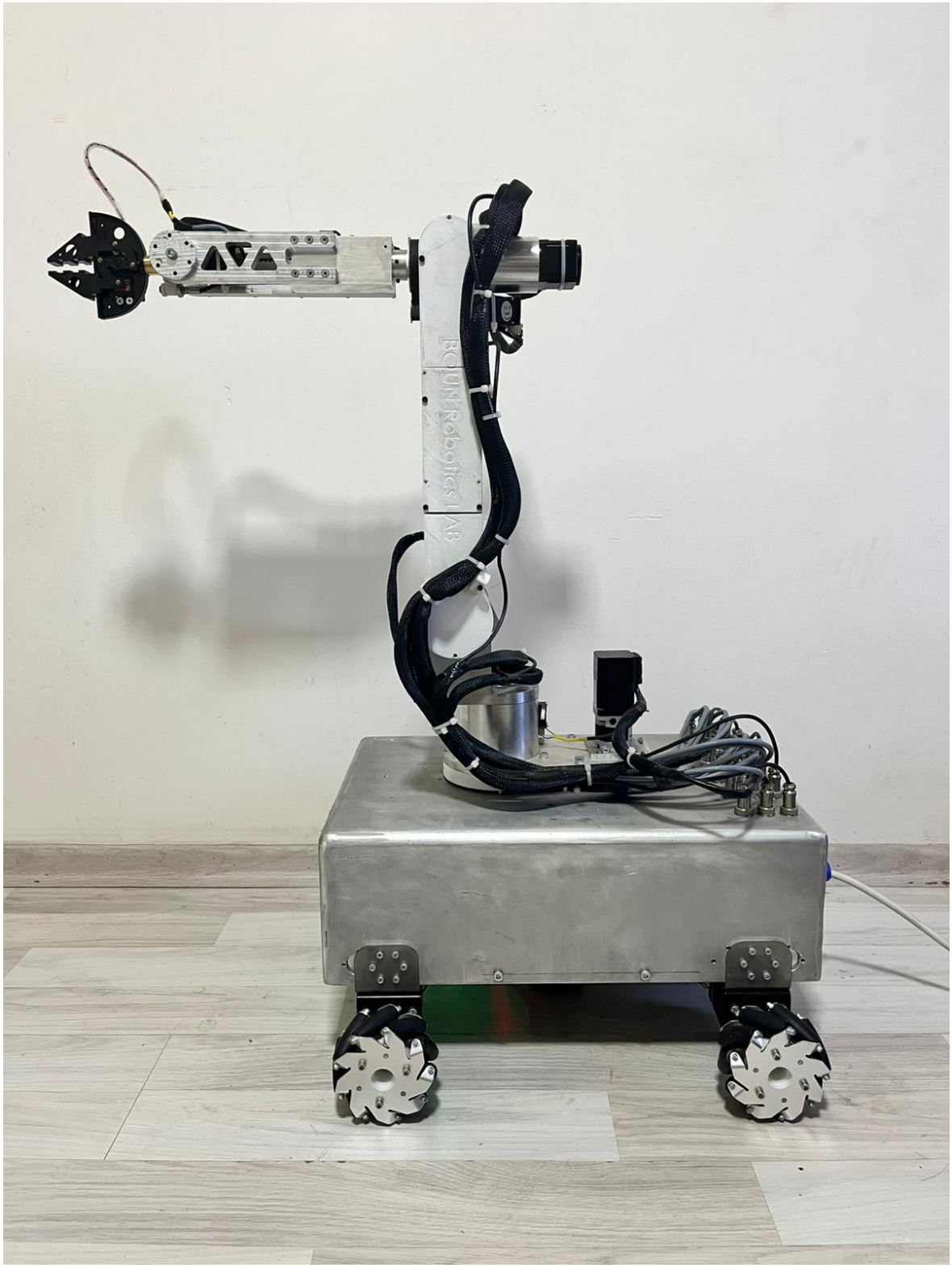

Figure 4.4. Prototype of the mobile manipulator.

The workspace boundaries of the mounted manipulator on its mobile base are shown in Figure 4.5. The visual presentation shows the reachable area of the robot when mobility combines with manipulation functions. The mobile base provides extensive operational territory that enhances the robotic arm's capabilities for tasks needing both relocation and restricted space access.

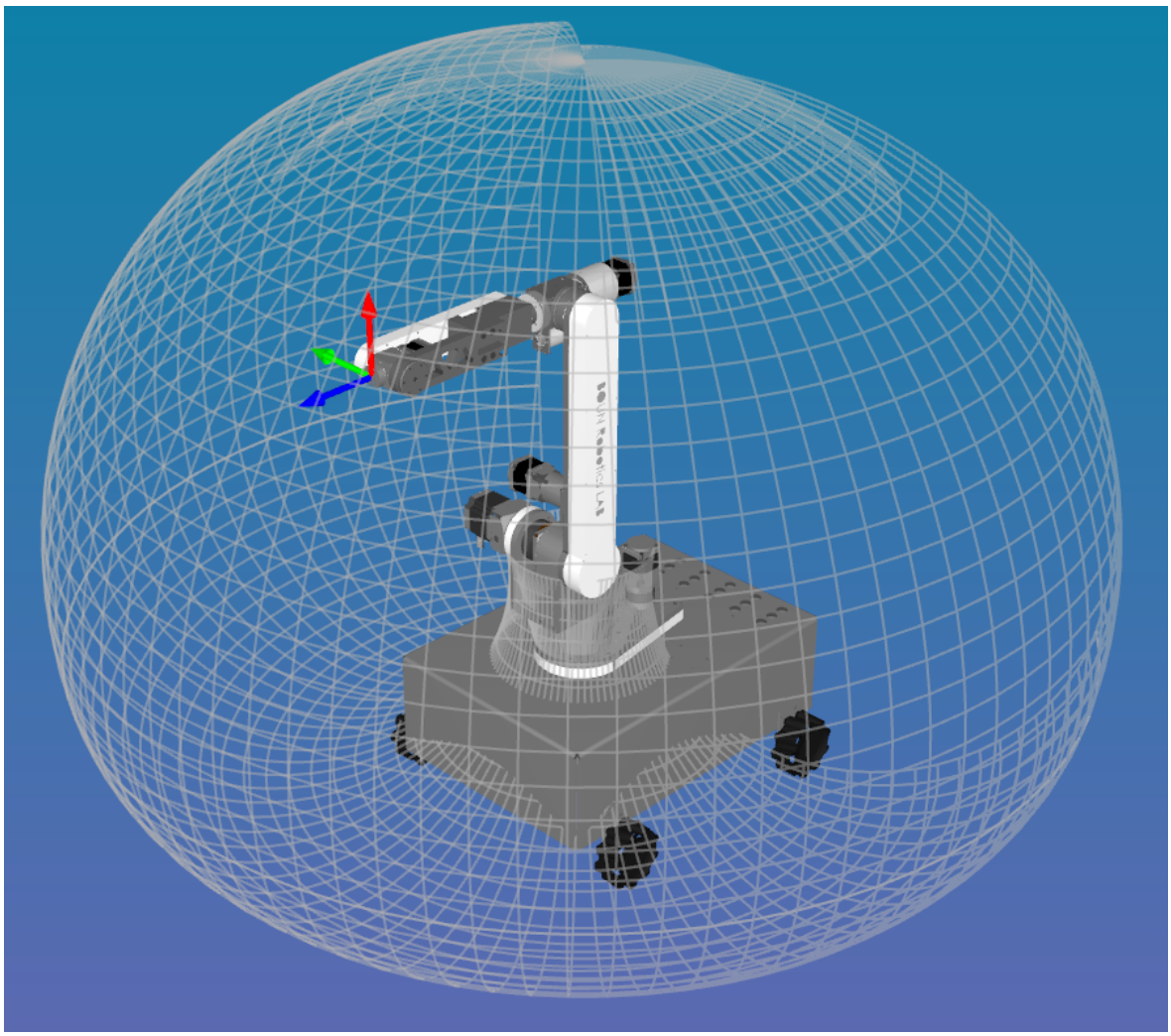

Figure 4.5. Mobile manipulator workspace visualization.

The integrated system consisting of precise actuation together with adaptive control and advanced sensing capabilities serves as a flexible system for robotic behavior examination under real-world conditions.

4.2.1. Custom Built Absolute Position and Pose Estimator

The mobile platform receives its real-time pose and position estimation through a custom sensor fusion module that incorporates a PMW3360 optical flow sensor and an MPU-6050 inertial measurement unit shown in Figure 4.6. The real-time fusion of inertial and optical data through this arrangement allows absolute positioning. The sensor module positioned at the mobile manipulator base center point facing downward 0.5 mm above the ground floor supported by PTFE gliding pads which are commonly used in high-performance mousepads. The particular sensor positioning creates an effective system which delivers accurate feedback regarding both translational and angular movement with minimal drift.

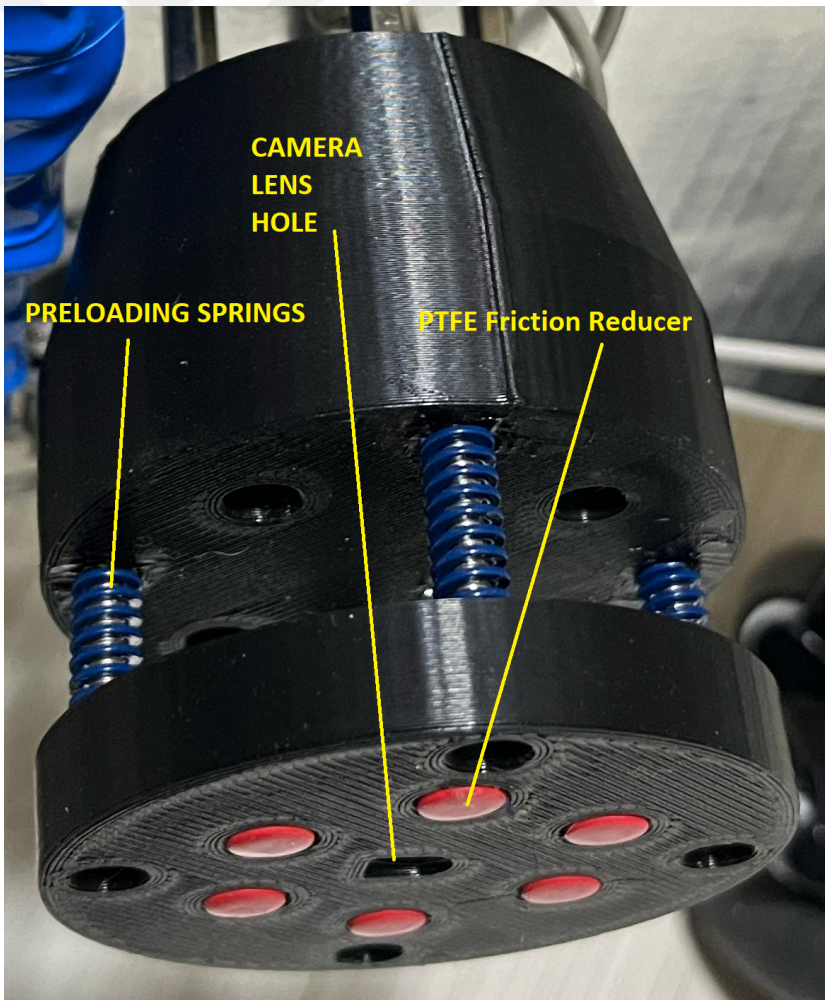

Figure 4.6. Mobile manipulator position and posture tracking device.

The PMW3360 optical sensor captures floor texture visual data at 12000 by 12000 pixels resolution through raw operation which it stores in 32-bit variables as multi-layered frames at 10 Mbps data transfer speed. The system detects surface texture pixel-level displacements through continuous frame comparison to determine translational shifts. The datasheet indicates that detected pixel shifts result in 2.1 micrometer displacements which the system reports through delta x or delta y values in motion directions. The mobile base maintains continuous real-time position tracking through absolute position measurements starting from (0, 0). The custom sensor module includes an MPU-6050 unit that provides filtered angular feedback through I2C communication at 400 kHz because of its onboard digital motion processor. The robot's central point yaw angle estimation becomes possible through local measurements. The work achieves its core scientific contribution by providing local pose estimation at high spatial resolution and fast update rates without needing external infrastructure. The system's low dependence on external data transmission layers produces minimal latency which prevents typical control problems caused by communication delays.

4.2.2. Sensor Calibration Processes

The PMW3360 optical flow sensor allows users to obtain raw monochrome image frames which are encoded as 32-bit variables. The device provides a burst mode operation which allows users to read thirty consecutive frames during a single readout sequence. We analyze each frame to detect pattern similarities between successive images and compute pixel movement in both x and y directions as shown in Figure 4.7. The motion estimation process tracks how specific features move between consecutive images to determine spatial displacement.

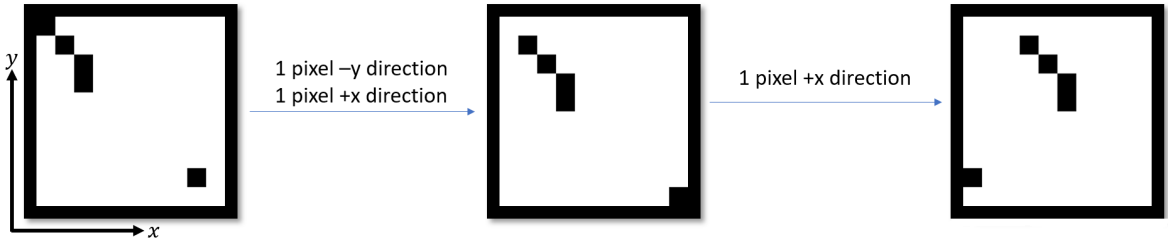

Figure 4.7. PMW3360 pixel shift demonstration.

We performed a specific calibration process to establish the physical measurement value of pixel displacement data. The sensor underwent testing on a platform that moved along a single axis shown in Figure 4.8. We performed multiple trials to determine the relationship between pixel increments and actual displacement which led to a physical step size measurement of $2.1\mu m$ per pixel count.

Figure 4.8. 3D printed PMW3360 calibration part.

We conducted additional tests to confirm the calibration factor's universality by moving the sensor across different textured and reflective surfaces including polished aluminum, ceramic tiles, matte adhesive foil, wood and laminated flooring. The sensor moved over set distances across different surfaces while we measured both pixel shifts and actual path lengths. The experiments showed that the sensor operated at $2.1\mu m$ resolution and maintained its calibration across different real-world operating scenarios.

The MPU6050 inertial measurement unit (IMU) contains both a three-axis gy-

roscope and a three-axis accelerometer in a single package to measure angular velocity and linear acceleration simultaneously. Most low-cost MEMS-based sensors including the MPU6050 experience sensor-specific biases and offsets that occur because of manufacturing processes. The device produces permanent offset readings that appear as non-zero values when the device remains motionless. The inherent biases in sensors accumulate during operation which causes major drift in orientation calculation.

The robot performs automatic offset calibration at its start-up stage as a solution to this issue. The sensor remains stationary throughout ten seconds during which the system records continuous gyroscope and accelerometer data streams. Each sensor channel maintains its offset value through calculation of mean values over the specified time period. The sensor uses these offset values as correction factors to zero its output when it is stationary for future measurements. This procedure ensures sensor-specific calibration because bias values exist differently across different sensors which enhances both measurement accuracy and long-term stability. After finishing the auto-calibration procedure the robot starts performing its operational tasks while providing dependable orientation and acceleration data.

The MPU6050 sensor shown in Figure 4.9 features an embedded Digital Motion Processor (DMP) which generates axis rotation angle outputs. The dedicated hardware component known as Digital Motion Processor (DMP) runs sensor fusion algorithms at its core which minimizes the processing requirements of the host microcontroller. The DMP system employs an internal version of Kalman filtering framework (which can be expressed through proprietary complementary/Kalman hybrid) to unite gyroscope and accelerometer readings. The orientation estimate remains reliable and stable in real time because this process reduces accelerometer noise in the short term and gyroscope drift in the long term.

Figure 4.9. MPU6050 sensor.

The sensor generates quaternions and Euler angles along with gravity compensated acceleration through the DMP without requiring complex sensor fusion programming from the user. The feature offers significant advantages to mobile robotic applications because it generates consistent orientation estimates with minimal latency when processing resources are limited. The DMP-based fused data stream served to improve heading reliability which enabled robust integration with the optical flow sensor for position estimation and trajectory tracking.

5. RESULTS AND DISCUSSION

5.1. Overview

The proposed adaptive PID controller achieves simulation and experimental results on a mecanum-wheeled mobile manipulator platform which this chapter demonstrates. The evaluation includes three path types: trapezoidal, square and circular which are tested at four different speeds of 100 mm/s, 200 mm/s, 300 mm/s and 400 mm/s. The analysis includes 24 trajectory tracking figures together with 36 adaptive gain response plots. Actual position of the mobile manipulator base is calculated using internal custom build motion tracking device. There were no external measurement except starting and finishing position deviation.

5.2. Trajectory Tracking at 100 mm/s

The performance of the mobile manipulator in trajectory tracking is shown in Figures 5.1, 5.2, 5.3, 5.4, 5.5, and 5.6 for trapezoidal, square and circular path geometries at a constant reference velocity of 100 mm/s. The figures show both the reference and actual trajectories as well as the posture angle evolution over time. The performance was quantitatively evaluated through root mean square error (RMSE), maximum path deviation and posture angle deviation for each case. The summary of the metrics is presented in Table 5.1 where rows represent different path types and columns represent simulation and experimental results for each error category.

The trapezoidal path showed the lowest RMSE value of 0.12 mm in the simulation data indicating high accuracy in tracking in idealized conditions. The minimum observed maximum deviation was 0.64 mm, which occurred in the circular path simulation, indicating smooth trajectory adherence. The best overall path-following performance in the experiment was on the square path, where an RMSE of 0.83 mm and a maximum deviation of 4.0 mm were recorded. The square path experiment showed

strong alignment with the desired trajectory even with the presence of sensor noise and real-world disturbances, confirming the robustness of the proposed control and localization framework. The results show that the system maintains high positioning accuracy across varied geometries and validates the effectiveness of the approach in both simulated and real-world settings.

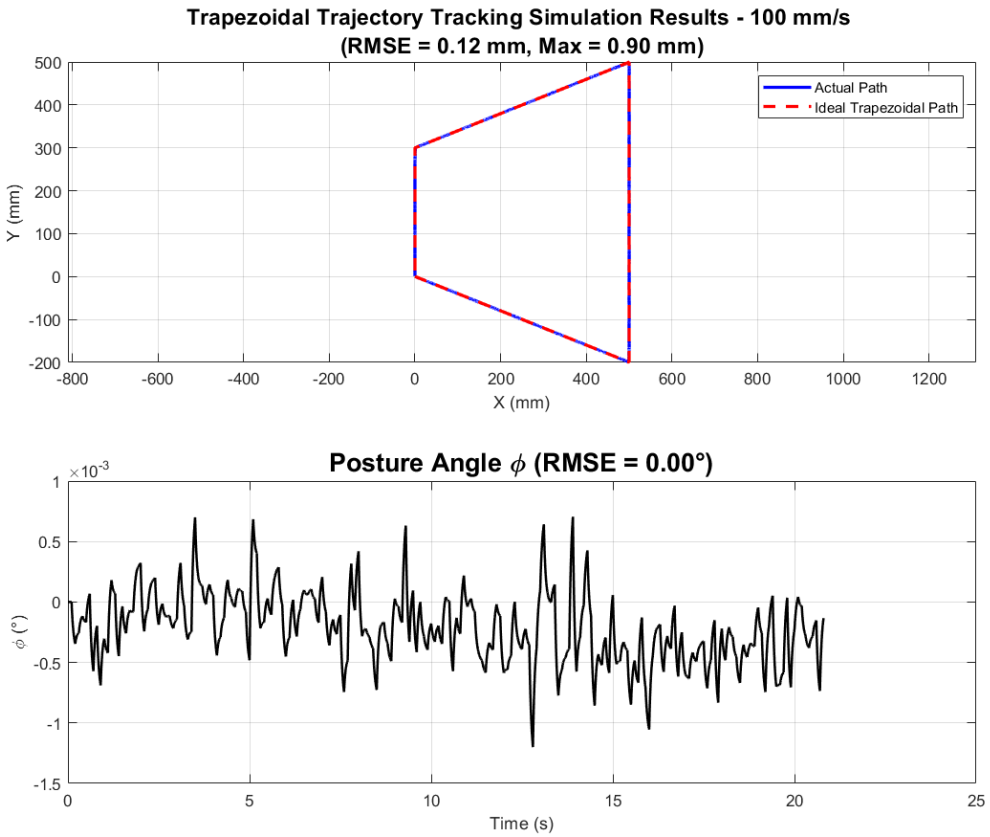

Figure 5.1. Simulated trajectory and posture angle tracking on trapezoidal path at 100 mm/s.

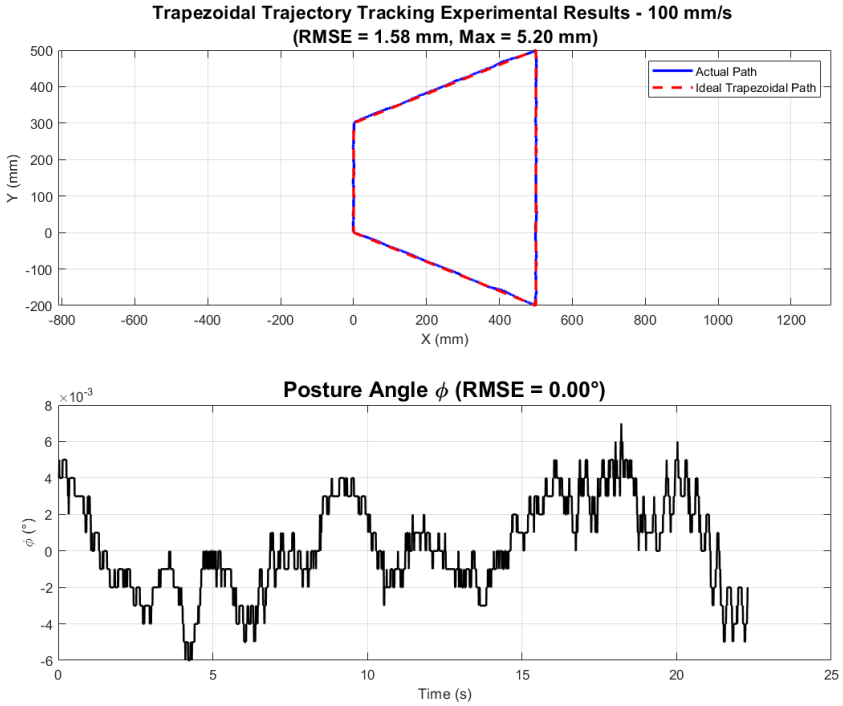

Figure 5.2. Experimental trajectory and posture angle tracking on trapezoidal path at 100 mm/s.

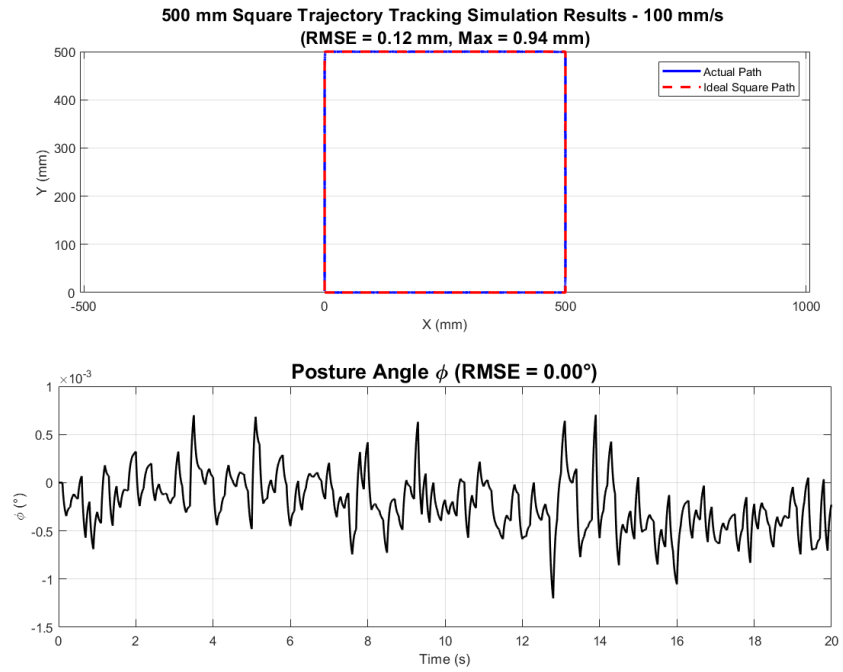

Figure 5.3. Simulated trajectory and posture angle tracking on square path at 100 mm/s.

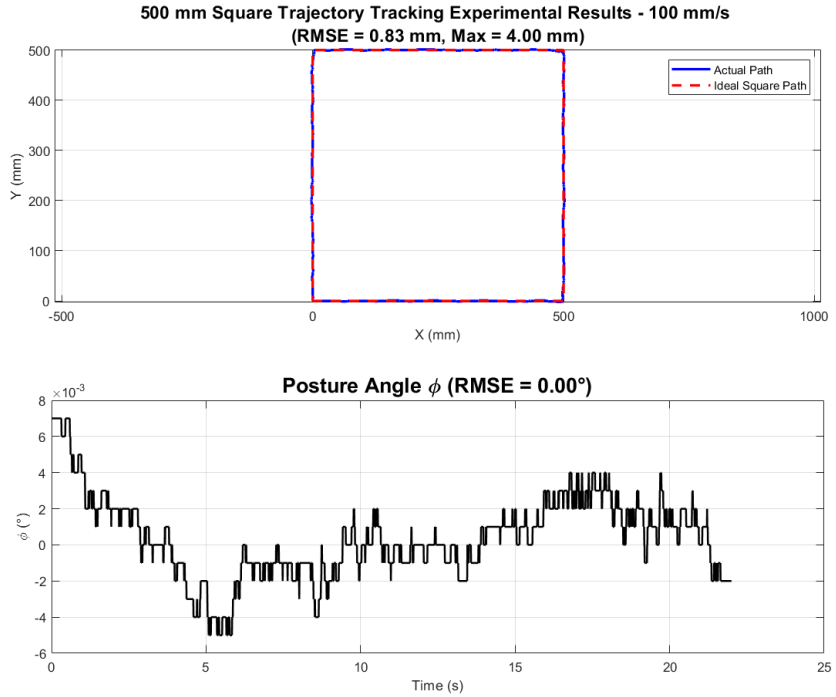

Figure 5.4. Experimental trajectory and posture angle tracking on square path at 100 mm/s.

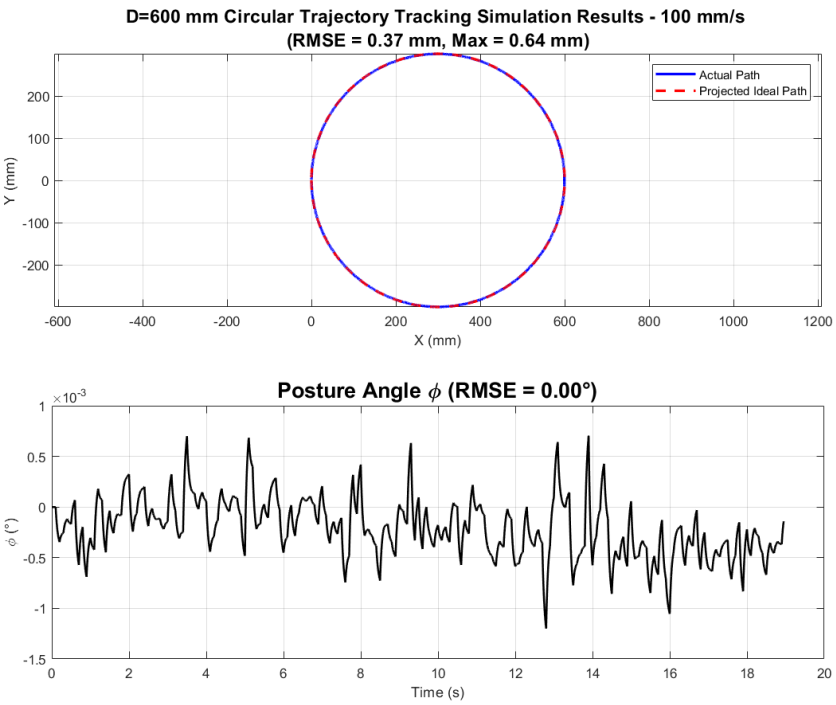

Figure 5.5. Simulated trajectory and posture angle tracking on circular path at 100 mm/s.

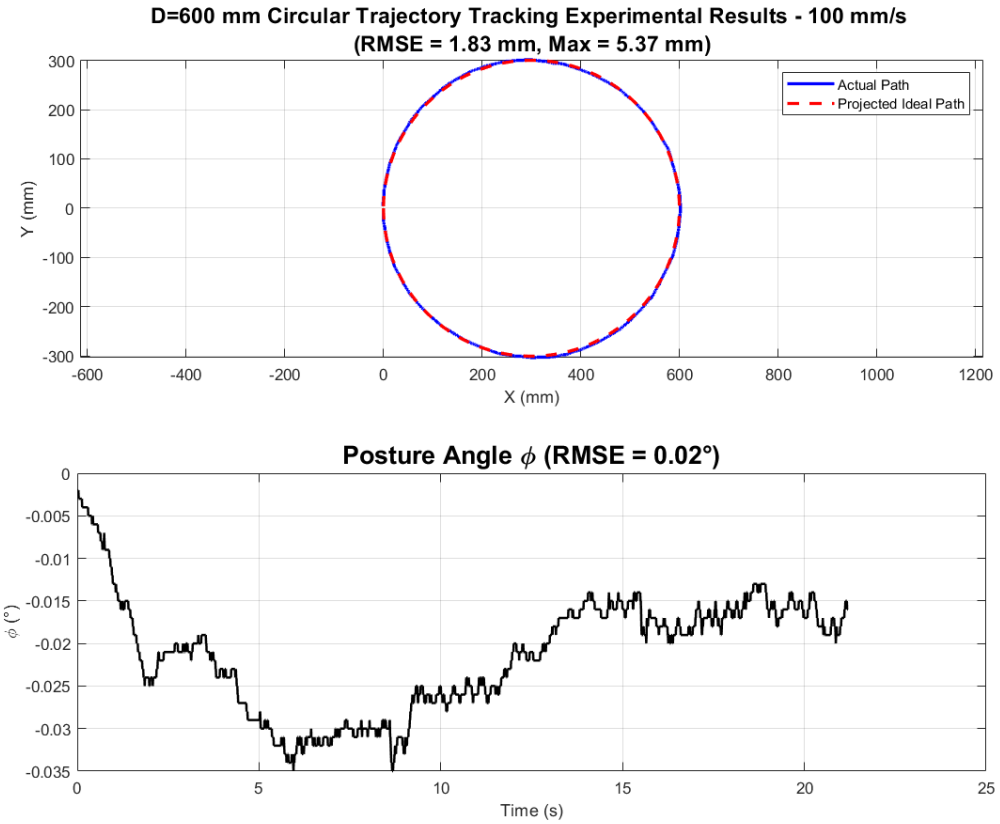

Figure 5.6. Experimental trajectory and posture angle tracking on circular path at 100 mm/s.

Table 5.1. Trajectory tracking errors at 100 mm/s for different path types.

Path Type	**Simulation**			**Experiment**		
	RMSE (mm)	Max Error (mm)	Posture Error	RMSE (mm)	Max Error (mm)	Posture Error
Trapezoid	0.12	0.90	0.00	1.58	5.20	0.00
Square	0.12	0.94	0.00	0.83	4.00	0.00
Circular	0.37	0.64	0.00	1.83	5.37	0.02

5.3. Trajectory Tracking at 200 mm/s

The performance of mobile manipulator trajectory tracking at 200 mm/s reference velocity appears in Figures 5.7, 5.8, 5.9, 5.10, 5.11, and 5.12 for trapezoidal, square

and circular paths. The system's dynamic behavior under increased speed conditions is shown through both positional tracking and heading orientation profiles in each figure. The best RMSE among simulation results was 0.11 mm on the trapezoidal path according to Table 5.2 which groups results by path geometry and data type (simulation or experiment). The circular path showed the smallest maximum path deviation under simulation conditions because its maximum error remained below 0.60 mm. The experimental trials showed that the circular path produced the lowest RMSE at 1.67 mm and the square path achieved the best maximum error performance with 4.0 mm. The proposed control architecture demonstrates reliable performance at higher speeds and path geometry remains essential for both absolute error and motion tracking consistency.

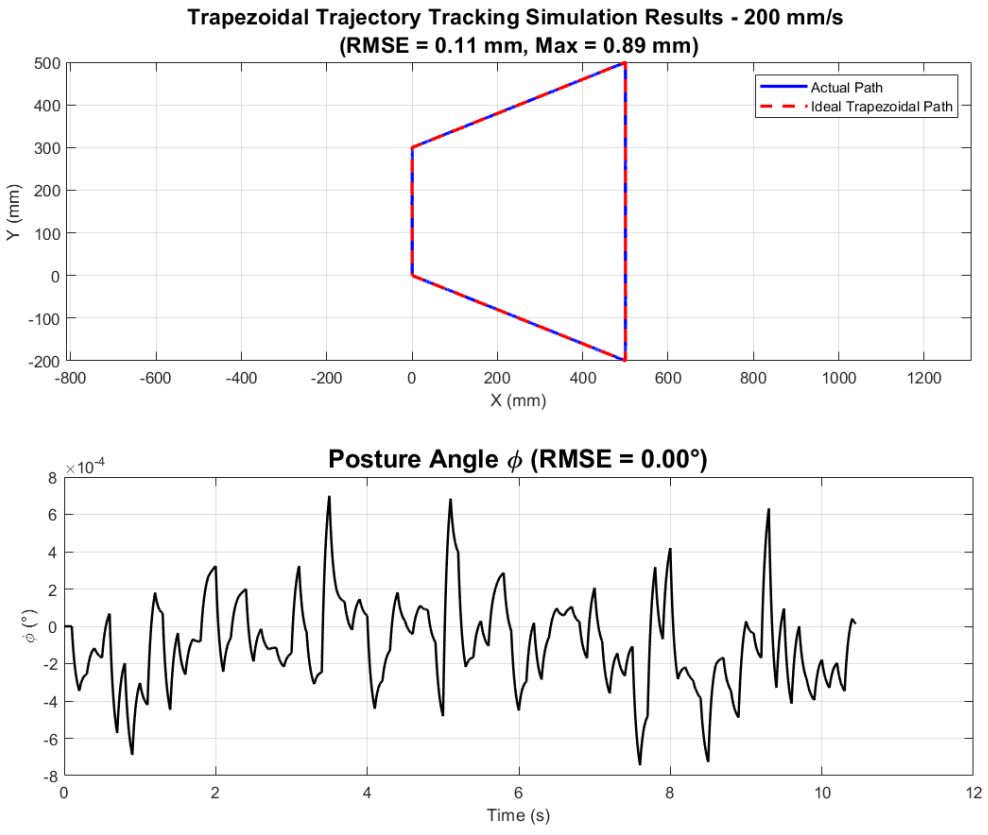

Figure 5.7. Simulated trajectory and posture angle tracking on trapezoidal path at 200 mm/s.

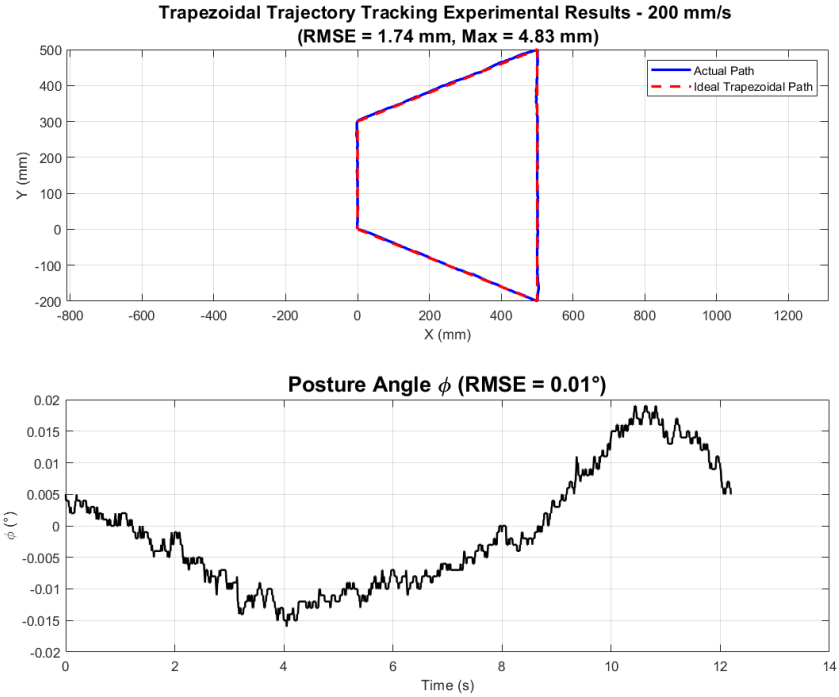

Figure 5.8. Experimental trajectory and posture angle tracking on trapezoidal path at 200 mm/s.

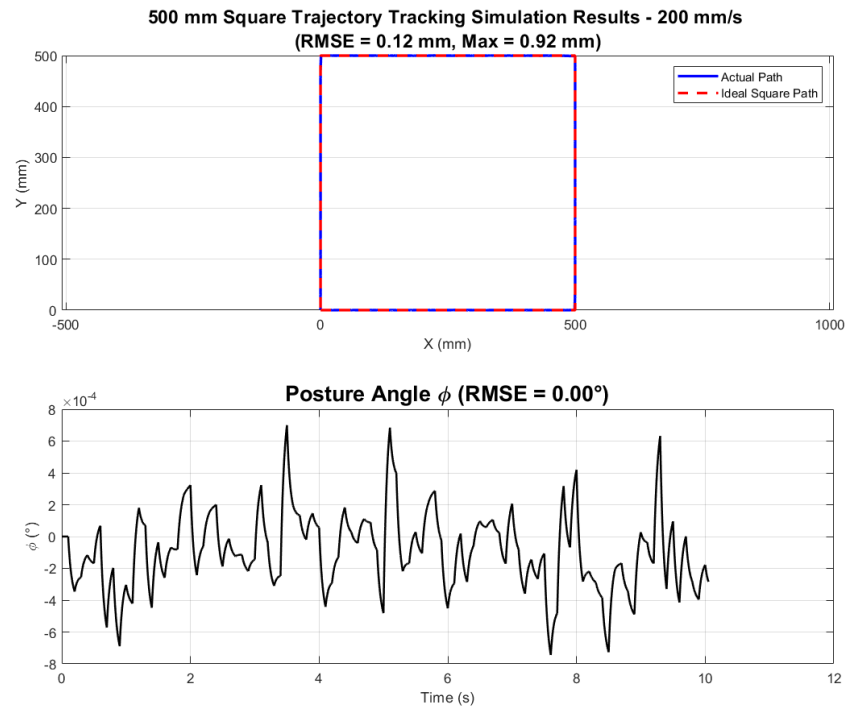

Figure 5.9. Simulated trajectory and posture angle tracking on square path at 200 mm/s.

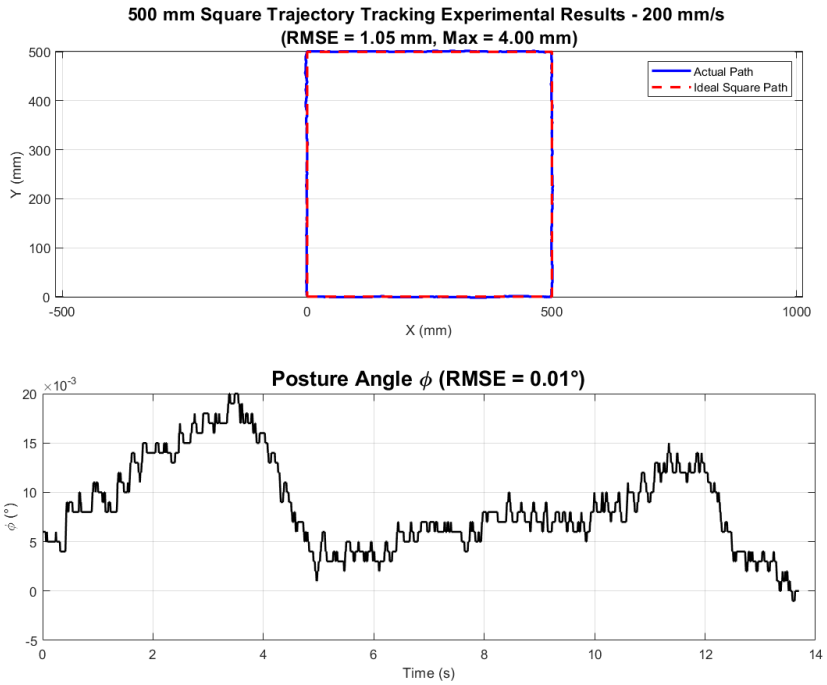

Figure 5.10. Experimental trajectory and posture angle tracking on square path at 200 mm/s.

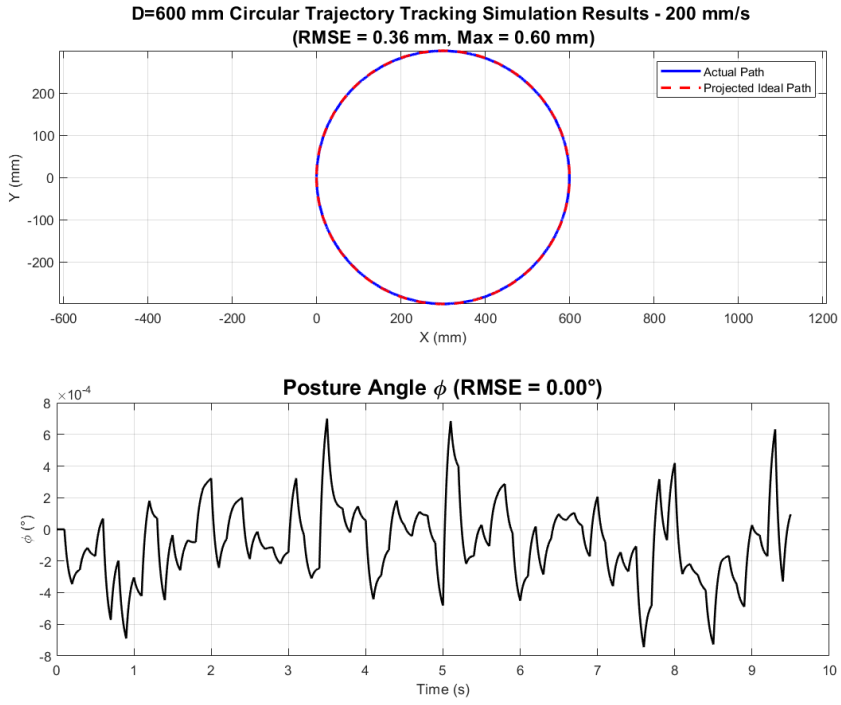

Figure 5.11. Simulated trajectory and posture angle tracking on circular path at 200 mm/s.

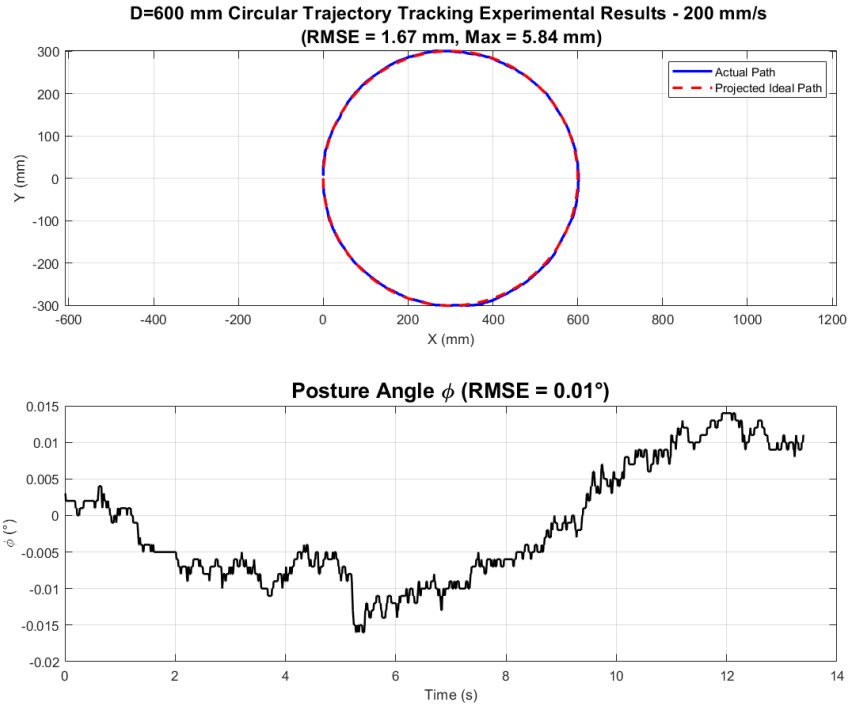

Figure 5.12. Experimental trajectory and posture angle tracking on circular path at 200 mm/s.

Table 5.2. Trajectory tracking errors at 200 mm/s for different path types.

Path Type	**Simulation**			**Experiment**		
	RMSE (mm)	Max Error (mm)	Posture Error	RMSE (mm)	Max Error (mm)	Posture Error
Trapezoid	0.11	0.89	0.00	1.74	4.83	0.01
Square	0.12	0.92	0.00	1.05	4.00	0.01
Circular	0.36	0.60	0.00	1.67	5.84	0.01

5.4. Trajectory Tracking at 300 mm/s

The mobile manipulator demonstrates its trajectory tracking performance in Figures 5.13, 5.14, 5.15, 5.16, 5.17, and 5.18 when moving at 300 mm/s across trapezoidal, square, and circular path types. The system demonstrates its ability to operate under high dynamic conditions through these figures which show both accurate tracking and

stable posture maintenance. The tracking performance metrics are summarized in Table 5.3 which presents results based on path geometry and test type. The trapezoidal path produced the minimum RMSE value of 0.10 mm during simulation testing and the circular path achieved the smallest maximum path deviation of 0.59 mm. The experimental results showed the square path achieved the minimum RMSE of 2.23 mm while the circular path produced the smallest maximum error of 8.08 mm. The control system maintained stable and responsive operation at increased velocity while keeping performance degradation within acceptable limits due to physical constraints and real-world experimental uncertainties.

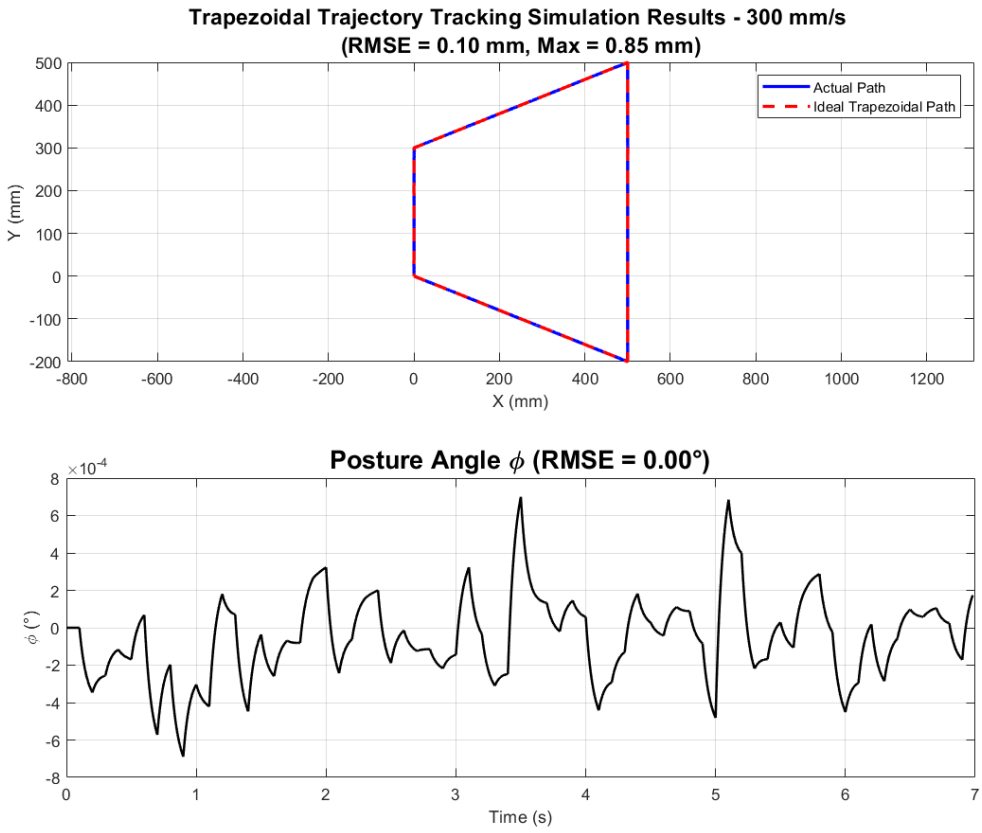

Figure 5.13. Simulated trajectory and posture angle tracking on trapezoidal path at 300 mm/s.

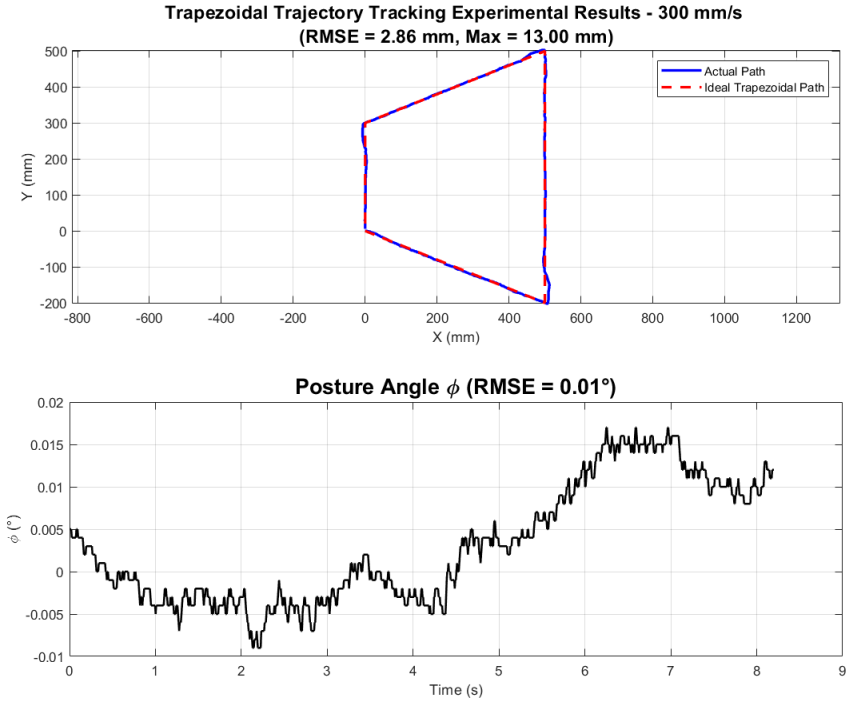

Figure 5.14. Experimental trajectory and posture angle tracking on trapezoidal path at 300 mm/s.

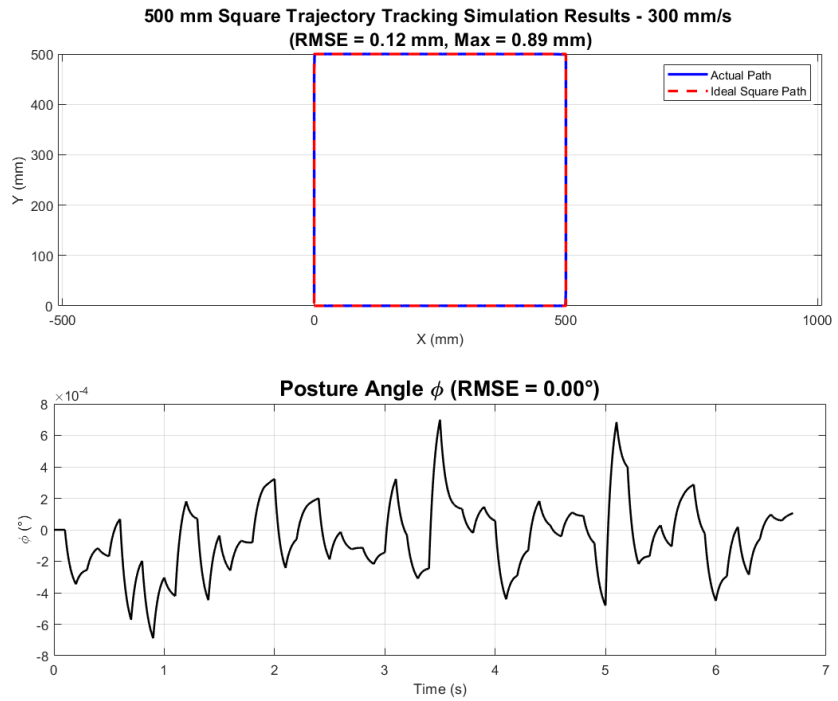

Figure 5.15. Simulated trajectory and posture angle tracking on square path at 300 mm/s.

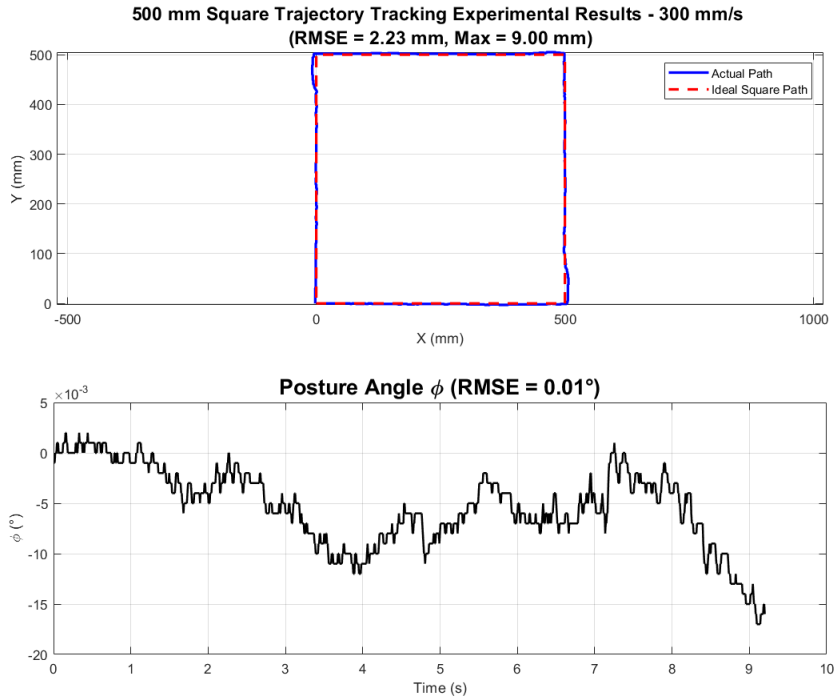

Figure 5.16. Experimental trajectory and posture angle tracking on square path at 300 mm/s.

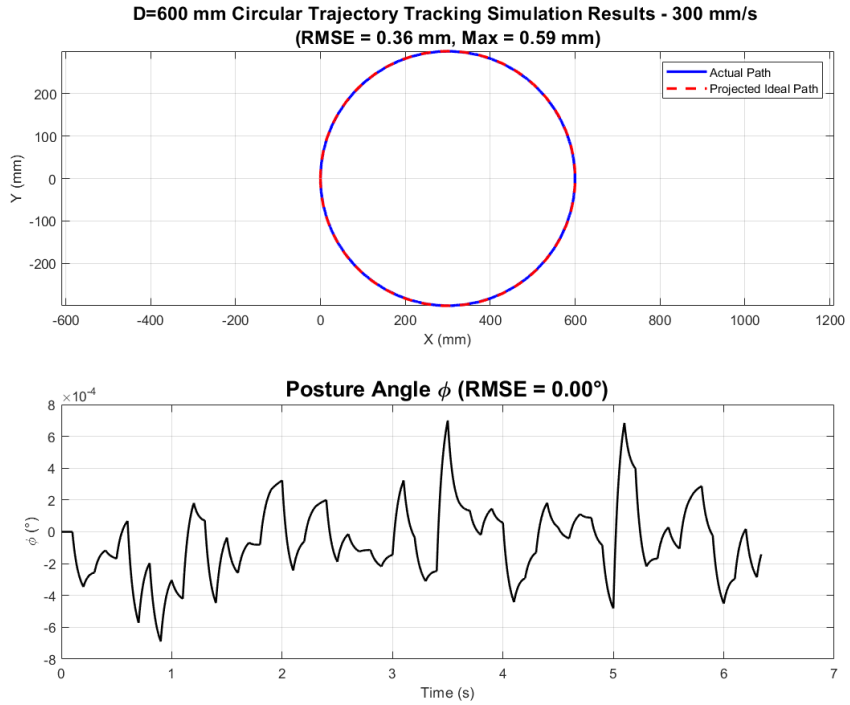

Figure 5.17. Simulated trajectory and posture angle tracking on circular path at 300 mm/s.

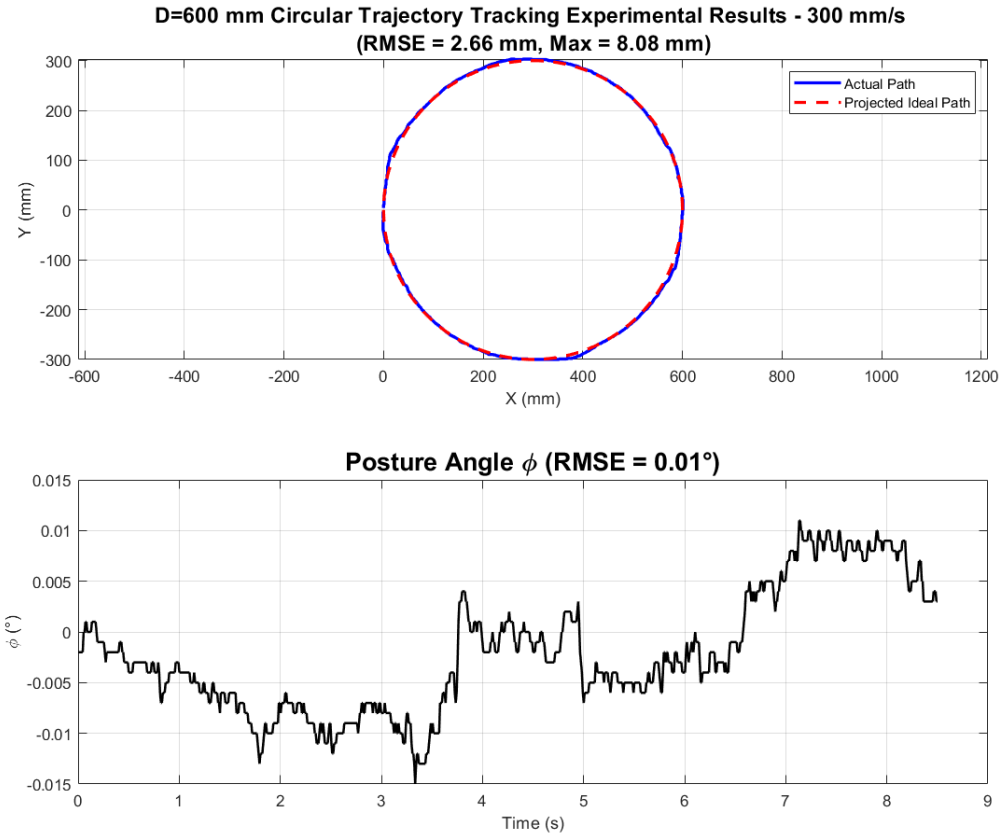

Figure 5.18. Experimental trajectory and posture angle tracking on circular path at 300 mm/s.

Table 5.3. Trajectory tracking errors at 300 mm/s for different path types.

Path Type	**Simulation**			**Experiment**		
	RMSE (mm)	Max Error (mm)	Posture Error	RMSE (mm)	Max Error (mm)	Posture Error
Trapezoid	0.10	0.85	0.00	2.86	13.00	0.01
Square	0.12	0.89	0.00	2.23	9.00	0.01
Circular	0.36	0.59	0.00	2.66	8.08	0.01

5.5. Trajectory Tracking at 400 mm/s

The tracking performance of the system under a reference velocity of 400 mm/s is shown in Figures 5.19, 5.20, 5.21, 5.22, 5.23, and 5.24. As reported in Table 5.4

the best simulation RMSE was obtained on the trapezoidal path with a value of 0.08 mm, while the lowest maximum deviation in simulation occurred on the circular path at 0.62 mm. Experimental performance showed significant degradation at this speed. Although the best RMSE was obtained on the circular path at 4.25 mm, and the smallest maximum deviation also occurred on the same path at 10.23 mm, the overall accuracy and consistency of path tracking were no longer within acceptable thresholds for precise motion tasks. This divergence between simulation and experimental results suggests that at velocities beyond 300 mm/s, the control strategy used in this work—while lightweight and effective at moderate speeds—becomes insufficient to fully compensate for the dynamic effects such as inertia, momentum, and slip. To maintain high-fidelity tracking at such elevated speeds, future implementations may require more advanced controllers that incorporate dynamic modeling of the system, including force and torque compensation, predictive elements, or full-body motion planning frameworks that consider the coupled dynamics of the mobile base and manipulator.

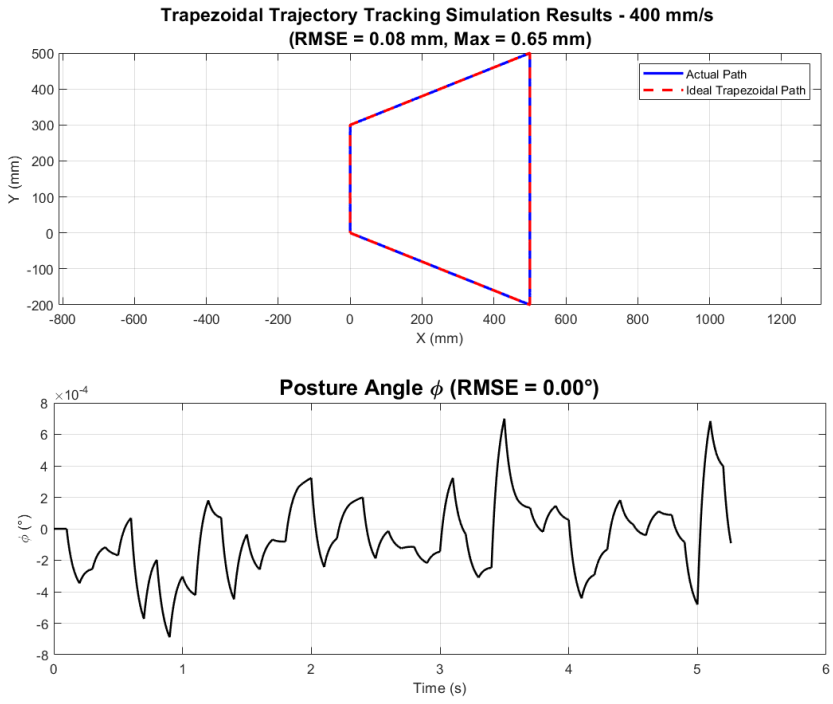

Figure 5.19. Simulated trajectory and posture angle tracking on trapezoidal path at 400 mm/s.

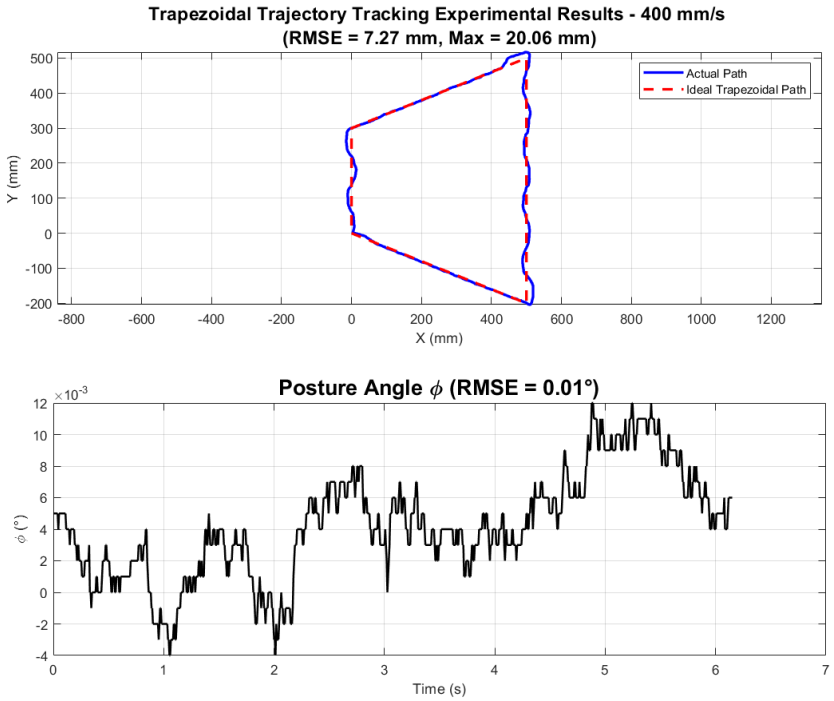

Figure 5.20. Experimental trajectory and posture angle tracking on trapezoidal path at 400 mm/s.

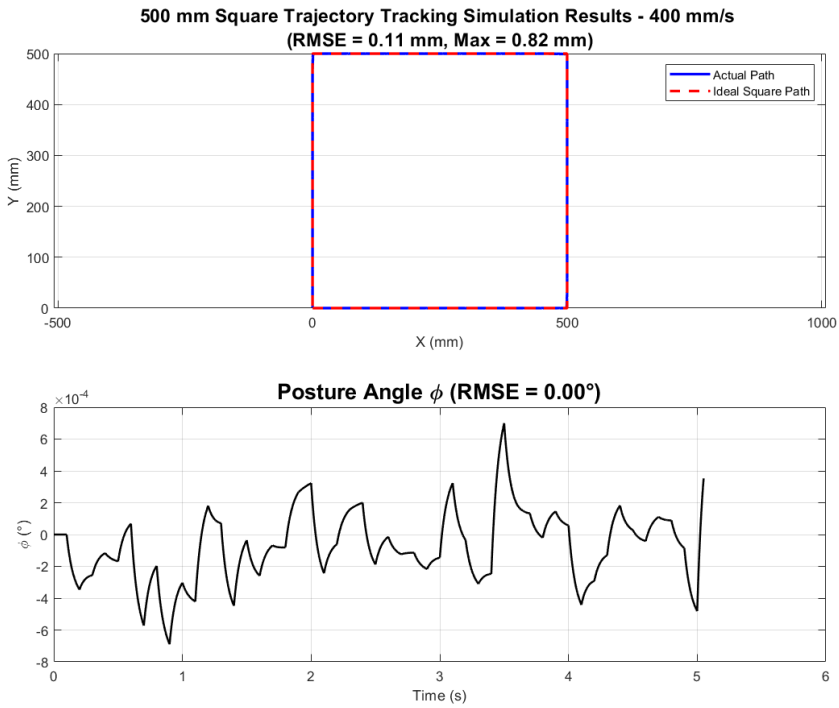

Figure 5.21. Simulated trajectory and posture angle tracking on square path at 400 mm/s.

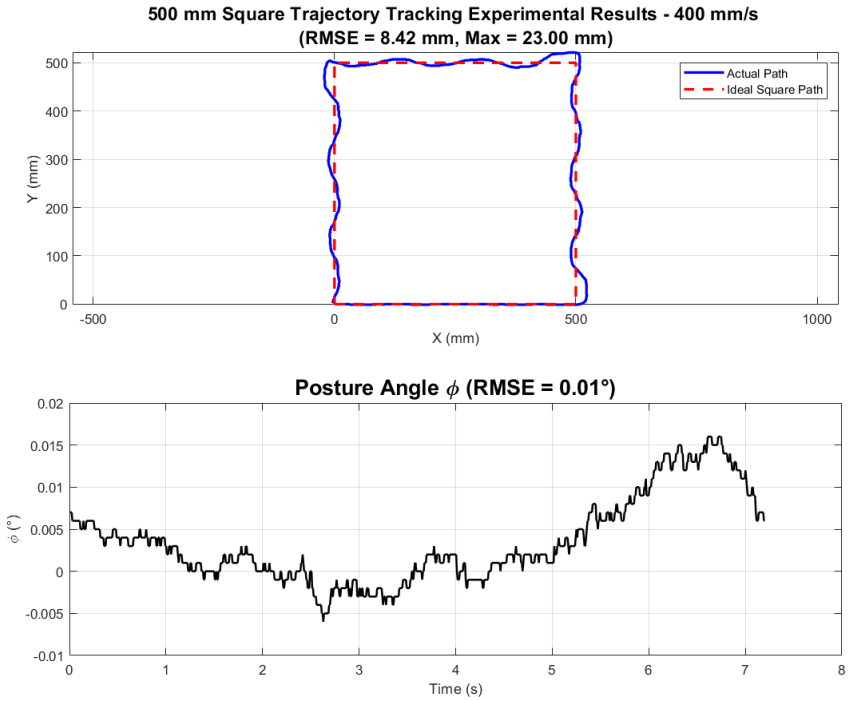

Figure 5.22. Experimental trajectory and posture angle tracking on square path at 400 mm/s.

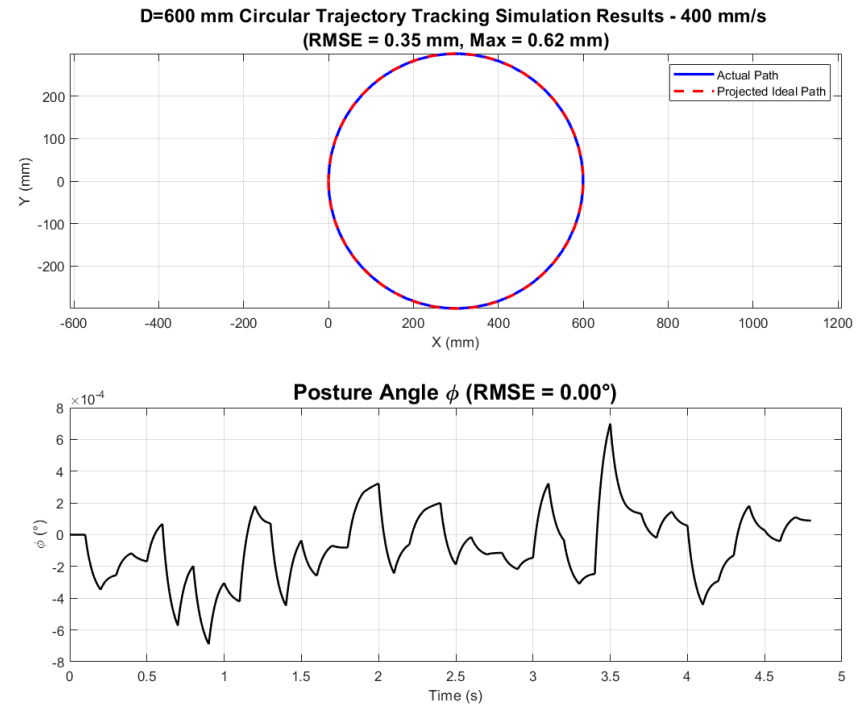

Figure 5.23. Simulated trajectory and posture angle tracking on circular path at 400 mm/s.

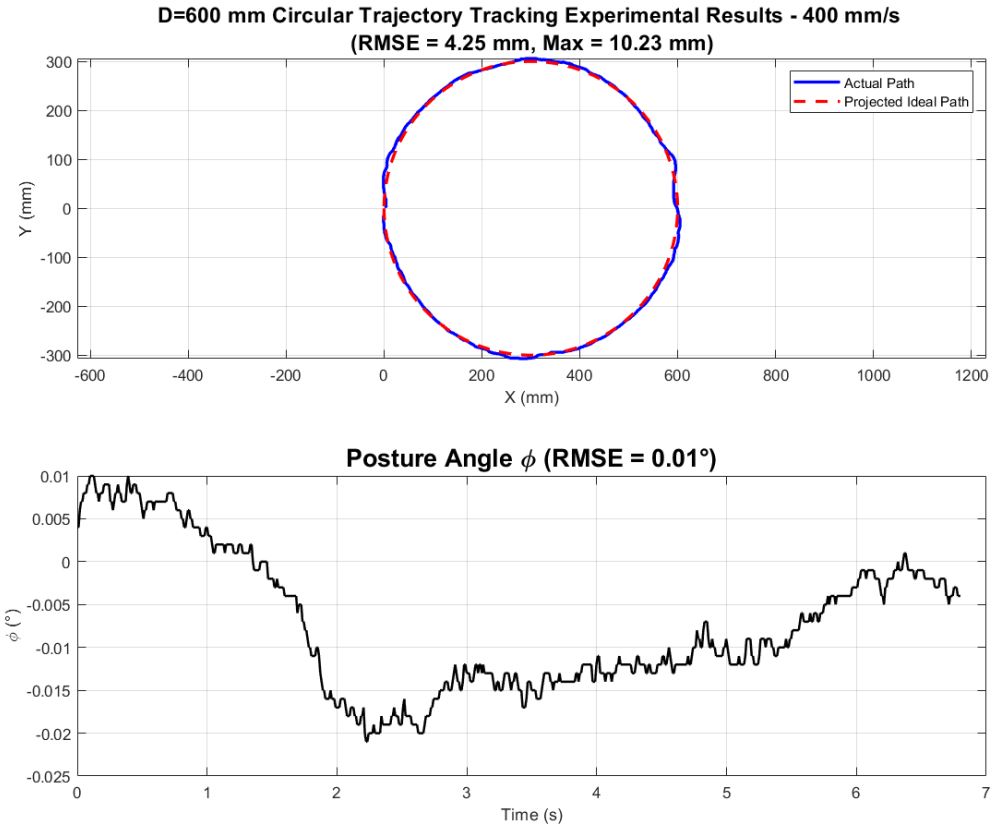

Figure 5.24. Experimental trajectory and posture angle tracking on circular path at 400 mm/s.

Table 5.4. Trajectory tracking errors at 400 mm/s for different path types.

Path Type	**Simulation**			**Experiment**		
	RMSE (mm)	Max Error (mm)	Posture Error	RMSE (mm)	Max Error (mm)	Posture Error
Trapezoid	0.10	0.85	0.00	2.86	13.00	0.01
Square	0.12	0.89	0.00	2.23	9.00	0.01
Circular	0.36	0.59	0.00	2.66	8.08	0.01

5.6. Adaptive Gain Behavior and Local Error Analysis

The adaptive PID controller behavior is studied through this section when tracking circular paths at 100 mm/s. The evaluation of both simulation and experimental

results includes studying the local x-axis error together with the real-time changes in PID gain coefficients.

The experimental results in Figure 5.25 demonstrate that the PID controller adjusts its gain values instantly according to the size of the local error. The integral gain K_i operates with a reverse effect because it quickly reduces its value when the robot moves away from the reference path to force immediate path recovery. The robot's approach to the trajectory causes K_i to increase gradually as the local error becomes smaller.

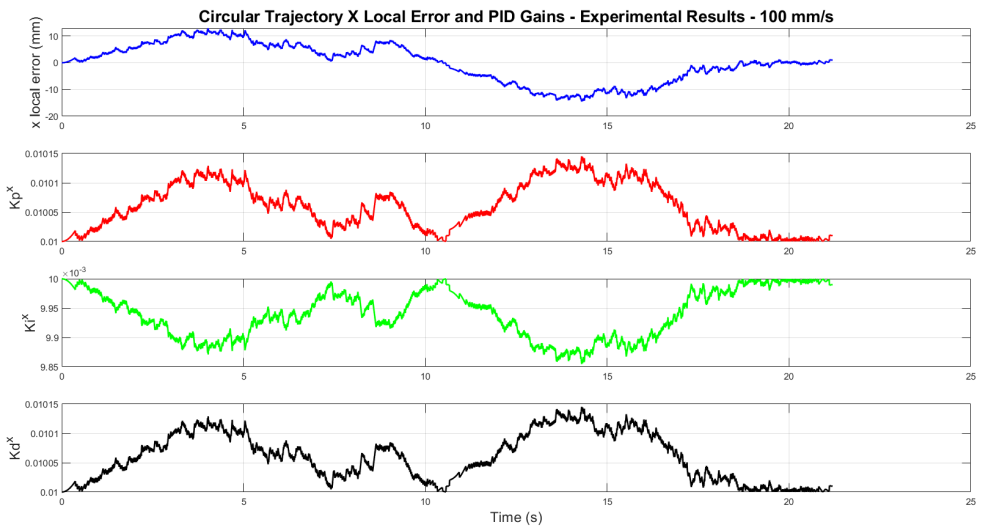

Figure 5.25. Experimental local error in x and adaptive PID gain response for circular path at 100 mm/s.

The simulation results in Figure 5.26 demonstrate this adaptive behavior. The controller successfully maintained a well-bounded local error response despite disturbances in the simulated control signals which represented real-world conditions. The integral term dominates the approach phase by staying lower than K_p and K_d which prevents overcompensation while stabilizing the correction effect.

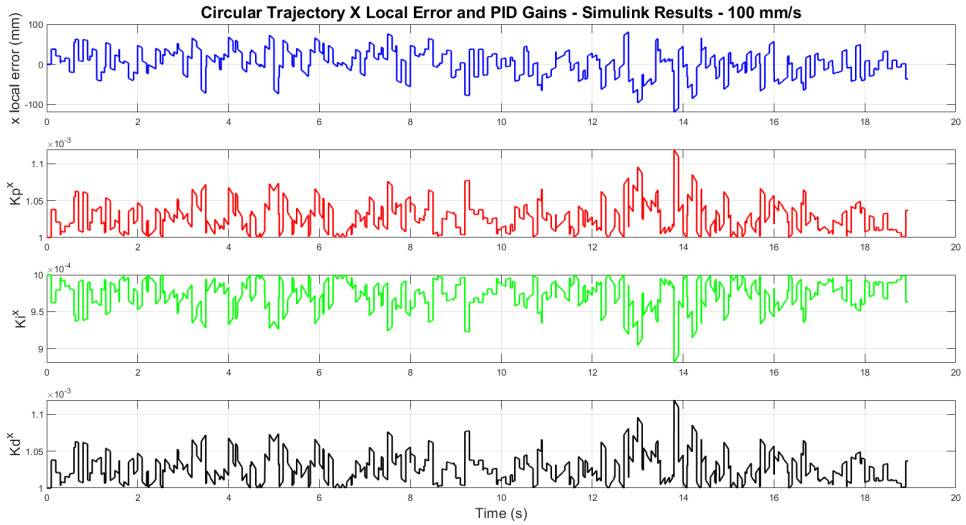

Figure 5.26. Simulated local error in x and adaptive PID gain response for circular path at 100 mm/s.

These results indicate that the integral term contributes effectively to convergence without introducing excessive overshoot. Once the robot closely aligns with the target path, the proportional and derivative terms decrease, ensuring smooth tracking behavior with minimal oscillation.

Additional local error and gain response plots—including x, y, and ϕ components for all path types and both 100 and 200 mm/s speeds—are presented in Appendix A for completeness and extended evaluation.

5.7. Mobile Manipulator Regular Task Simulation

The mobile base executes point-to-point movements in regular task execution but the manipulator operates only when the base stops moving. The manipulator performs two separate intervals of work for object handling during the task execution as shown in Figure 5.27. The tool performs exact movements during these periods while the mobile base stays motionless. The manipulator goes back to its home position after finishing the picking operation while the mobile base starts its movement toward the target

destination. The manipulator performs the placing operation at the target position before returning home while the base travels back to its initial location. The total duration of this configuration amounts to 70.19 seconds.

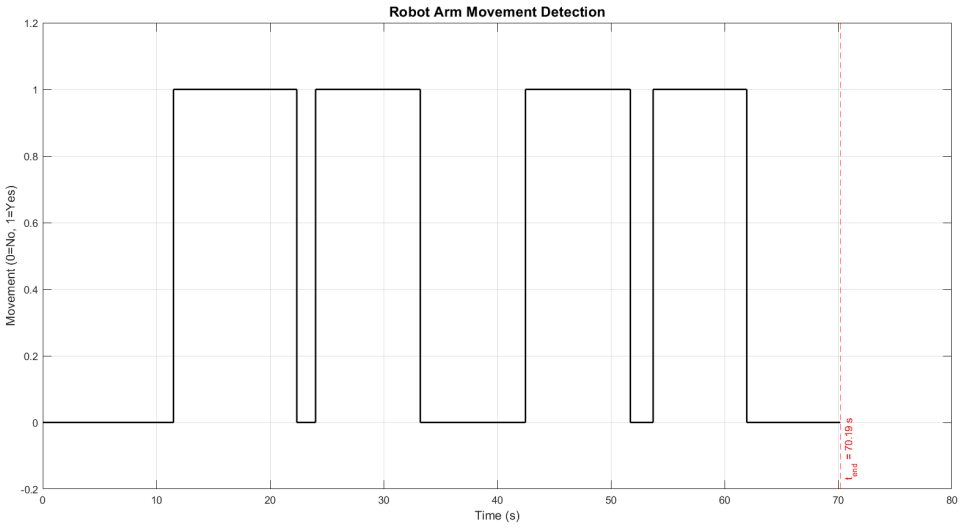

Figure 5.27. Movement detection of the robot arm in the regular task configuration.

The movement of the mobile base is illustrated in Figure 5.28, where the desired and actual trajectories are compared. Additionally, the posture angle tracking of the base is presented to reflect the accuracy of the control algorithm.

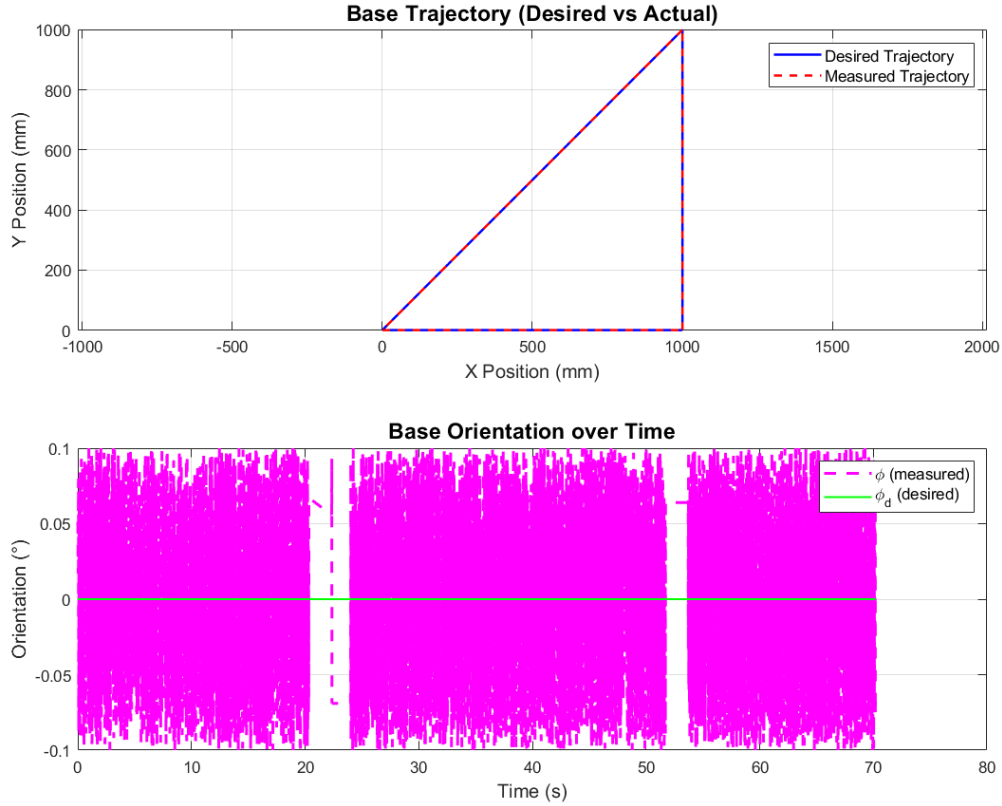

Figure 5.28. Base trajectory tracking and orientation error during regular task operation.

The complete 3D path followed by the tool frame during the regular task is shown in Figure 5.29. It clearly demonstrates that the tool only moves while the base remains stationary. After completing the task, the manipulator returns to its initial position before any base motion is triggered. The separate phases of pick, home return, and place operations are clearly distinguishable.

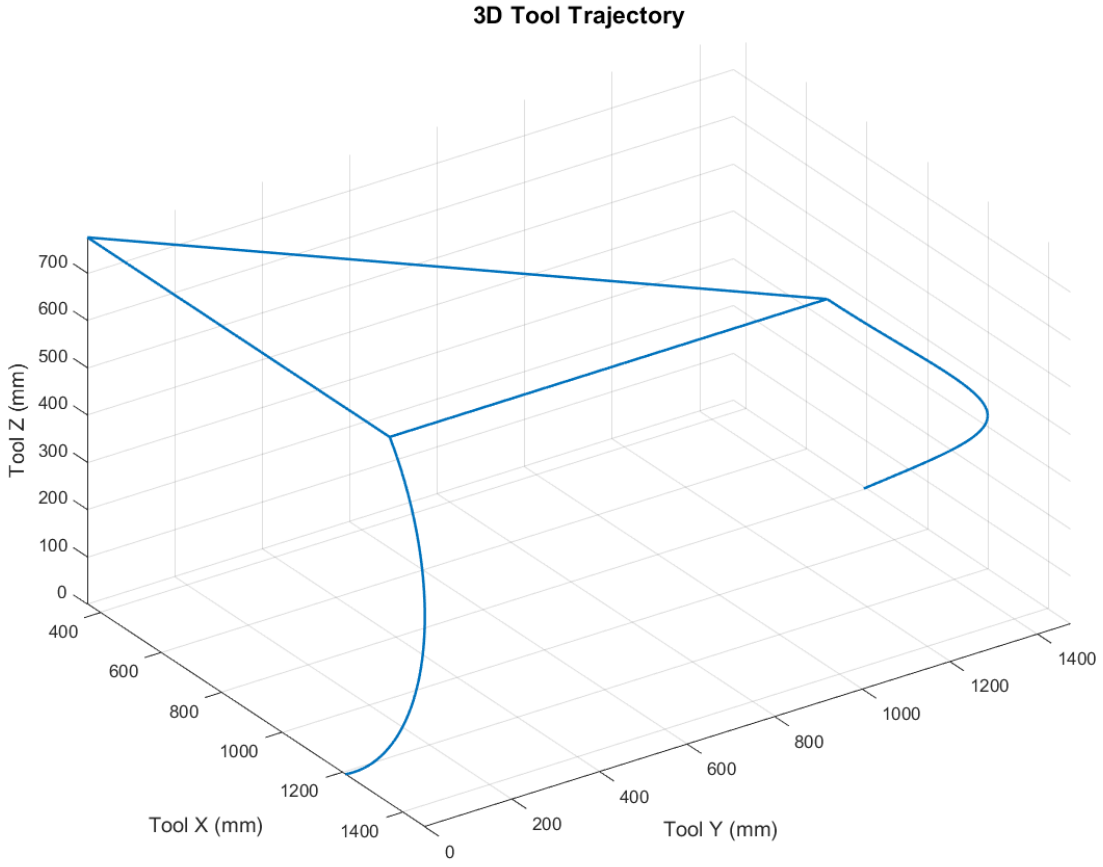

Figure 5.29. 3D trajectory of the tool frame in the regular task operation.

5.8. Mobile Manipulator Synchronized Task Simulation

The synchronized task execution requires the manipulator and mobile base to operate together from start to finish. The robot arm maintains continuous motion in Figure 5.30 as it follows the base to achieve task execution in reduced time. The total execution time for this task reaches 40.38 seconds which represents approximately half of the regular task duration.

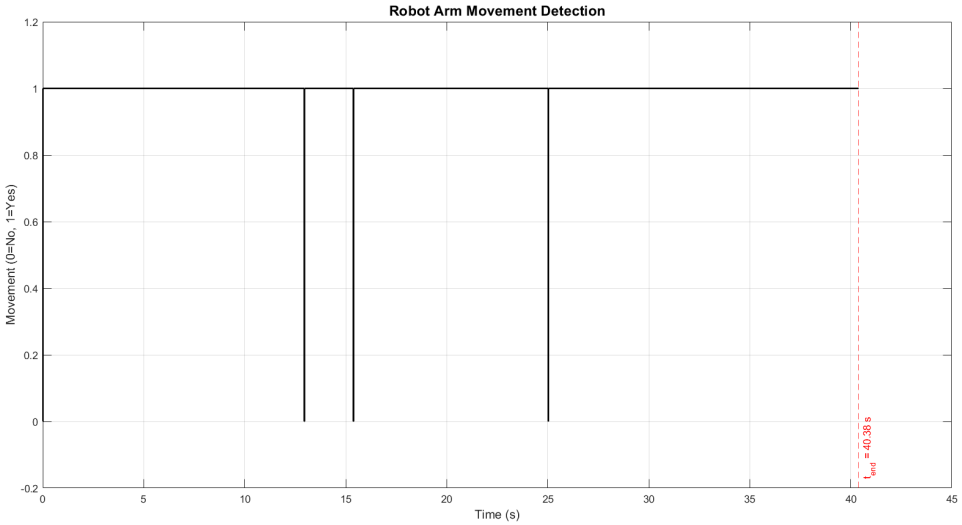

Figure 5.30. Continuous movement detection of the manipulator during synchronized task execution.

The mecanum-wheel-based mobile platform achieves its efficiency through its precise position tracking abilities. The base position feedback in real time enables precise synchronization of manipulator motion which also minimizes tool travel distance. The 3D trajectory of the tool emerges from the coordinated motion as shown in Figure 5.31. The coordination algorithm achieves its purpose through the smooth and efficient motion.

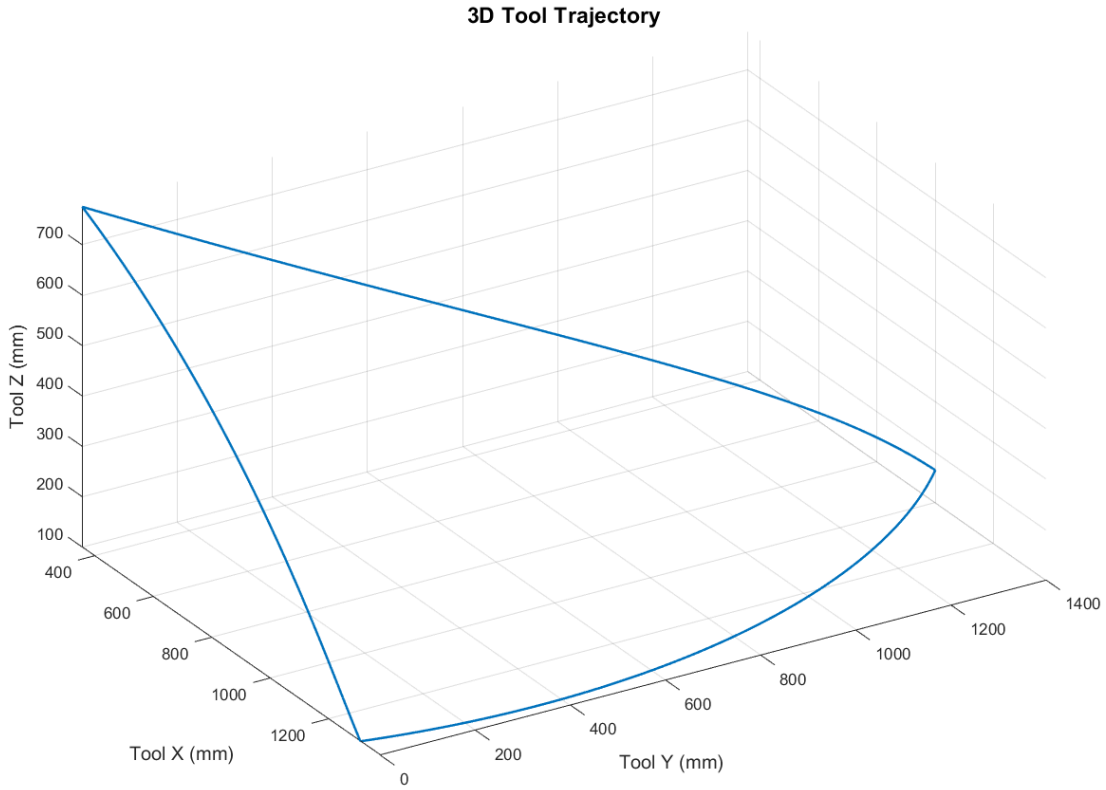

Figure 5.31. 3D trajectory of the tool frame during synchronized task operation.

The base motion tracking accuracy is confirmed through Figure 5.32 which shows the desired and actual base movement together with posture angle monitoring. The synchronized task mode shows base motion without stops which results in continuous posture deviations throughout the graph. The regular task displayed zero-deviation intervals when the base remained stationary.

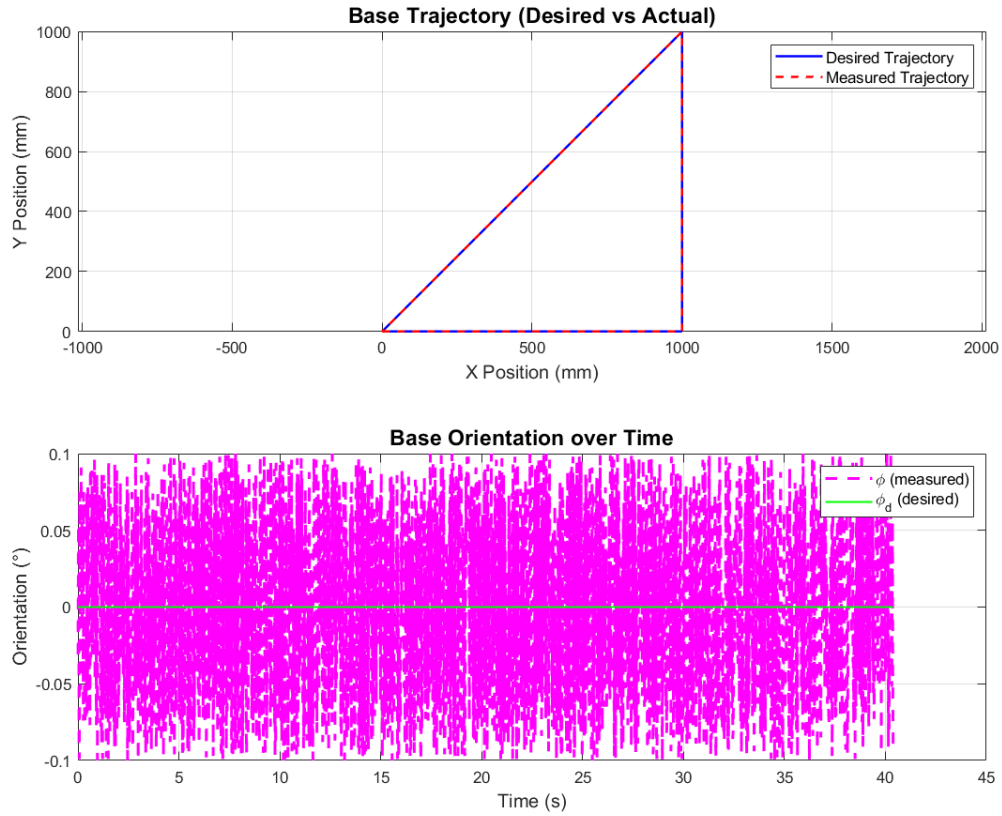

Figure 5.32. Base trajectory tracking and orientation error during synchronized task operation.

5.9. Discussion

The experimental research and simulation data presented in this chapter prove the efficiency and operational boundaries and developmental potential of the proposed adaptive PID control method for real-time trajectory tracking on a mecanum-wheeled mobile manipulator. The system demonstrates consistent position tracking and posture control across different reference paths (trapezoidal, square, circular) and four increasing speed levels (100 mm/s, 200 mm/s, 300 mm/s, 400 mm/s).

The proposed controller reached sub-millimeter RMSE levels in simulation (0.12 mm) and showed strong experimental performance that achieved its best tracking behavior with the square path under real-world disturbances when operating at 100 mm/s.

The controller's precision in trajectory corrections along with minimal posture deviation demonstrates its capability to precisely correct deviations while avoiding oscillatory behavior. The adaptive gain structure especially the dynamic adjustment of integral gain worked as the key element in balancing fast convergence with overshoot suppression as shown by gain evolution plots.

The simulation results maintained excellent tracking accuracy throughout the speed increase to 200 mm/s and 300 mm/s because RMSE values stayed below 0.12 mm and maximum deviations stayed under 1 mm. Experimental trials conducted at these speeds demonstrated reasonable error growth yet maintained acceptable performance for typical motion applications which proved the control strategy's ability to handle moderate dynamic loads. Real-world tracking performance was best on square and circular paths which showed the controller's ability to work with various geometric shapes.

The experimental results at 400 mm/s speed showed significant divergence from simulated data. The simulated tracking accuracy reached its lowest point at 0.08 mm RMSE on the trapezoidal path but experimental results showed substantial deterioration. The RMSE value exceeded 4 mm and maximum errors reached above 10 mm in certain instances especially during circular path execution. The lightweight control architecture provides sufficient performance during moderate speeds but fails to provide the required dynamic compensation needed for handling high-speed inertial lag and slip and unmodeled disturbances. The control law fails to address the coupled dynamics and force interactions as well as time delays stemming from actuation and measurement layers when operating at such velocities.

The task-based simulations provide concrete evidence about how performance of trajectories affects practical operations. During regular task execution the manipulator operated while the base remained stationary for a total duration of 70.19 seconds. The synchronized mode decreased the execution time to 40.38 seconds which demonstrates the benefits of continuous motion. The efficiency benefits from this approach depend

on reliable real-time localization together with quick control responses thus confirming the need for precise feedback systems and fast adaptation mechanisms.

The system's autonomy depended heavily on the self-contained localization module which used optical flow sensing and inertial measurements. The system used global displacement and orientation estimation without external infrastructure to deliver real-time feedback without latency or data loss which was critical for adaptive control. Sensor fusion improvements along with noise filtering enhancements would lead to better tracking performance when the base moves faster or irregularly.

The proposed adaptive PID controller demonstrates a practical and efficient control solution for mobile manipulator operation at speeds ranging from low to moderate. The control approach works well for embedded systems because it combines simple design with efficient computation and local feedback processing. Future research should develop hybrid control methods which combine system dynamics with model-based predictive control and force-aware planning approaches to enhance speed-related and demanding operational capabilities. The implementation of these enhancements would result in robust performance under aggressive motion profiles which would enable the development of more agile and capable mobile manipulation platforms.

6. CONCLUSION AND FUTURE WORK

The chapter summarizes the main findings of the research and examines the overall significance of the mobile manipulator system presented in this paper. Various directions for future development and enhancement are identified based on experimental and simulation-based evaluations.

6.1. Conclusion

This thesis demonstrates the modeling process and control system and simulation of a mecanum-wheeled mobile base and 6-DOF industrial robotic arm mobile manipulator system. The research combined mechanical modeling and trajectory planning and controller design and real-time hardware implementation to achieve precise coordinated motion for mobile manipulation tasks.

The main contribution of this study is the development of a new adaptive PID controller which adjusts its gains based on the real-time local tracking error. Extensive simulation and experimental tests validated this strategy by using trapezoidal, square and circular trajectories with multiple speed levels. The controller provided millimeter-level accuracy while handling different dynamic situations and kept its stability during high speed operations.

The study presented two methods for task execution: regular and synchronized base and manipulator operations. The base movement was only allowed when the arm was not in use in the regular strategy but the synchronized mode allowed both base and manipulator to move simultaneously which reduced the total task time to 40.38 seconds from 70.19 seconds. The synchronized execution success demonstrates how precise base control with localization feedback should be integrated into motion planning systems for better performance. Synchronized control tools move through 3D space which demonstrates that synchronized control provides better efficiency while

minimizing manipulator motion requirements. Enabling energy efficient task execution.

This research proves that the proposed adaptive control approach can successfully be used in mobile manipulation applications. The proposed system which includes real-time localization and error-driven control adaptation and coordinated base-arm motion provides a foundation for future applications in warehouse logistics and autonomous assembly lines and service robotics.

6.2. Contributions

The research introduces multiple original contributions to mobile manipulation studies which concentrate on adaptive control systems and onboard localization methods. The research makes the following scientific contributions to the field:

Lightweight Adaptive PID Framework: A new adaptive PID controller was developed which adjusts its gains through local error magnitude measurements within the robot's coordinate system. The proposed formulation enables proportional controller responsiveness adjustment through a simple method that avoids extensive system identification and heavy computational optimization.

The proposed gain adjustment strategy reduces overshoot by using linear integral term reduction when the robot strays from the path and then restores the integral term when the system reaches the reference point which results in fast convergence without oscillations.

Custom Localization Hardware: A specific onboard localization device received development and implementation. The sensor module combines PMW3360 optical flow camera with MPU-6050 inertial measurement unit to determine real-time planar displacement and orientation without requiring external infrastructure or global positioning systems.

The system provides tracking precision at the micrometer level (2.1 μ m per pixel) and delivers direct pose estimation at the mobile manipulator base center. The system provides continuous posture feedback and accurate trajectory tracking during dynamic conditions.

Experimental Evaluation on Mecanum-Wheeled Platform: The system underwent thorough validation through both simulation tests and real-world experiments that included different path types and speed profiles. The experimental results showed the proposed method maintained accurate tracking at speeds up to 300 mm/s but revealed control limitations at higher speeds which indicate potential directions for future development.

The research advances the field through its combination of robust infrastructure free localization and control capabilities for mobile manipulator platforms using resource-constrained hardware.

6.3. Future Work

Despite the positive results from this thesis there are still multiple opportunities to enhance the work. The adaptive PID controller achieved positive results in structured areas with planned paths but needs evaluation for unstructured and changing environments. Real-time path replanning alongside obstacle avoidance features are necessary to enable this system for advanced real-world applications.

The trajectory profiles applied in this research study made constant velocity assumptions. Future research should explore speed-scaling methods and acceleration-aware planners to minimize overshoot while improving smoothness during transitions particularly when operating at high speeds. Robot motion can become more human-like and efficient through the integration of velocity profiling methods such as S-curve or trapezoidal profiles.

The primary opportunity for advancement exists through the combination of learning-based or optimal control methods. The implementation of Model Predictive Control (MPC) and Reinforcement Learning (RL) enables better long-term decisions and system adaptation to changes through time while optimizing adaptive PID multipliers for energy consumption and task completion time.

The system will gain both robustness and flexibility through the addition of new sensory components. The system will gain better pose estimation accuracy through camera system or UWB-based localization combined with existing odometry and IMU data which enables reliable operation in dynamic and partially observable environments. Such developments would represent the following stage for building autonomous mobile manipulation systems that perform complex tasks in real-world settings.

REFERENCES

1. Röhrig, C., D. Heß and F. Künemund, "Motion Controller Design for a Mecanum Wheeled Mobile Manipulator", *Conference on Control Technology and Applications (CCTA)*, pp. 444–449, Institute of Electrical and Electronics Engineers, Kohala Coast, Hawai, 2017.

2. Sadeghian, H., M. Keshmiri, L. Villani and B. Siciliano, "Priority Oriented Adaptive Control of Kinematically Redundant Manipulators", *2012 Institute of Electrical and Electronics Engineers International Conference on Robotics and Automation*, pp. 293–298, Institute of Electrical and Electronics Engineers, St. Paul, Minnesota, USA, 2012.

3. Burgess-Limerick, B., J. Haviland, C. Lehnert and P. Corke, "Reactive Base Control for On-the-Move Mobile Manipulation in Dynamic Environments", *Institute of Electrical and Electronics Engineers Robotics and Automation Letters*, Vol. 9, No. 3, pp. 2048–2055, 2024.

4. Quan, S. and J. Chen, "Mecanum Wheeled Mobile Platform Odometry Error Analysis and Correction", *Proceedings of IncoME-V & CEPE Net*, pp. 728–736, Springer International Publishing, Zhuhai, China, 2021.

5. Han, K.-L., O.-K. Choi, J. Kim, H. Kim and J. Lee, "Design and Control of Mobile Robot with Mecanum Wheel", *ICCAS-SICE - ICROS-SICE International Joint Conference 2009, Proceedings*, pp. 2932–2937, Institute of Electrical and Electronics Engineers, Fukuoka, Japan, 2009.

6. Nguyen, Q.-N. and Q.-C. Pham, "Planning Optimal Trajectories for Mobile Manipulators under End-effector Trajectory Continuity Constraint", *International Conference on Robotics and Automation (ICRA)*, pp. 14356–14362, Institute of Electrical and Electronics Engineers, Yokohama, Japan, 2024.

7. Burgess-Limerick, B., C. Lehnert, J. Leitner and P. Corke, "An Architecture for Reactive Mobile Manipulation On-The-Move", *Robotics and Automation (ICRA)*, pp. 1623–1629, Institute of Electrical and Electronics Engineers, ExCel London, 2023.

8. Merz, D., M. Robinson and T. Cappellari, "Mobile Manipulator Coordinated Motion for Continuous Processing (IR&D Project 10-R6298)", SwRI Internal R&D Annual Report – Manufacturing & Construction, 2024.

9. Rakita, D., B. Mutlu and M. Gleicher, "Manipulation-on-the-Move Using a Networked Controller Structure", *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4931–4938, Institute of Electrical and Electronics Engineers, Prague, Czech Republic, 2021.

10. Zhang, S., S. Cheng and Z. Jin, "A Control Method of Mobile Manipulator Based on Null-Space Task Planning and Hybrid Control", *Machines*, Vol. 10, No. 12, 2022.

11. Chen, H., X. Zang, Y. Liu, X. Zhang and J. Zhao, "A Hierarchical Motion Planning Method for Mobile Manipulator", *Sensors*, Vol. 23, No. 15, 2023.

12. Sleiman, J.-P., F. Farshidian, M. V. Minniti and M. Hutter, "A Unified MPC Framework for Whole-Body Dynamic Locomotion and Manipulation", *Institute of Electrical and Electronics Engineers Robotics and Automation Letters*, Vol. 6, No. 3, pp. 4688–4695, 2021.

13. Pardi, T., V. Maddali, V. Ortenzi, R. Stolkin and N. Marturi, "Path Planning for Mobile Manipulator Robots Under Non-Holonomic and Task Constraints", *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6749–6756, Institute of Electrical and Electronics Engineers, Las Vegas, NV, USA, 2020.

14. Sayar, E., X. Gao, Y. Hu, G. Chen and A. Knoll, "Toward Coordinated Planning

and Pierarchical Optimization Control for Highly Redundant Mobile Manipulator", *ISA Transactions*, Vol. 146, pp. 16–28, 2024.

15. Yuan, W., Y.-H. Liu, C.-Y. Su and F. Zhao, "Whole-Body Control of an Autonomous Mobile Manipulator Using Model Predictive Control and Adaptive Fuzzy Technique", *Institute of Electrical and Electronics Engineers Transactions on Fuzzy Systems*, Vol. 31, No. 3, pp. 799–809, 2023.

16. Thai, N. H., T. T. K. Ly, N. T. Long and L. Q. Dzung, "Trajectory Tracking Using Linear State Feedback Controller for a Mecanum Wheel Omnidirectional", *Advances in Asian Mechanism and Machine Science*, pp. 411–421, Springer International Publishing, Hanoi, Vietnam, 2022, reproduced with Permission from Springer Nature, License Number: 6076460785751.

17. Wan, A. Y. S., L. Yi, A. A. Hayat, M. C. Gen and M. R. Elara, "Complete Area-Coverage Path Planner for Surface Cleaning in Hospital Settings Using Mobile Dual-Arm Robot and GBNN with Heuristics", *Complex & Intelligent Systems*, Vol. 10, pp. 6767–6785, Springer International Publishing, 2024.

18. Hernández, J. C. O. and D. I. R. Almeida, "Kinematic Control in a Four-Wheeled Mecanum Mobile Robot for Trajectory Tracking", *The Journal of Engineering*, , No. 9, p. e70006, 2024.

19. Chen, Y.-H., "Nonlinear Adaptive Optimal Control Design and Implementation for Trajectory Tracking of Four-Wheeled Mecanum Mobile Robots", *Mathematics*, Vol. 12, No. 24, 2024.

20. Sun, Z., Z. Li, H. Xie, Y. Zheng, J. Zheng and B. Chen, "Precise Trajectory Tracking of Mecanum-Wheeled Omnidirectional Mobile Robots via a Novel Fixed-Time Sliding Mode Control Approach", *Control Theory and Technology*, Vol. 22, pp. 596–611, 2024, reproduced with Permission from Springer Nature, License Number: 6076461300574.

21. Tzafestas, S. G., *Introduction to Mobile Robot Control*, Elsevier, Oxford, 2014.

22. Qian, J., B. Zi, D. Wang, Y. Ma and D. Zhang, "The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System", *Sensors*, Vol. 17, No. 9, 2017.

APPENDIX A: Local Error and Adaptive PID Gain Plots

This appendix provides all local error and corresponding PID gain plots for each path type and control axis (x, y, and ϕ) at 100 mm/s and 200 mm/s. These figures complement the main analysis in Section 4.2 by offering a complete view of the adaptive PİD controller's dynamic behavior.

A.1. Trapezoidal Path

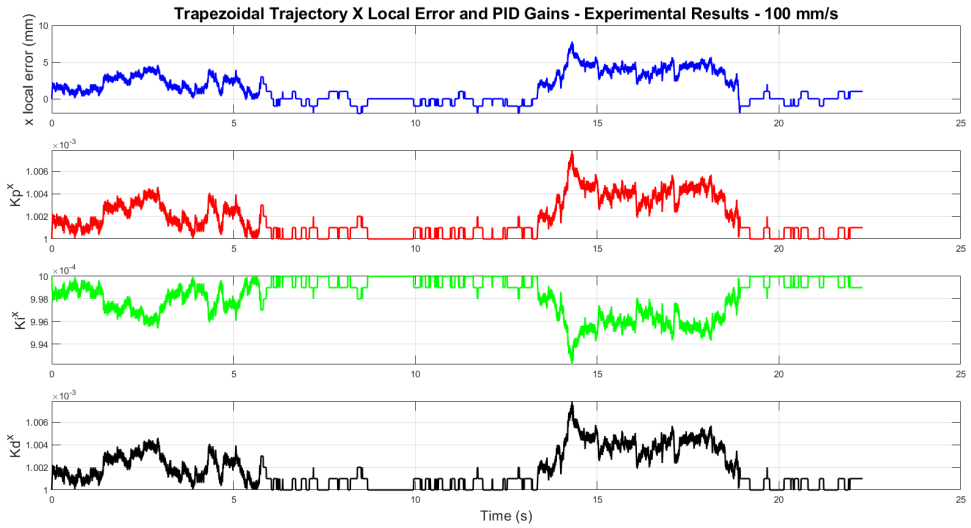

Figure A.1. Experimental local error in x and adaptive PID gain response on trapezoidal path at 100 mm/s.

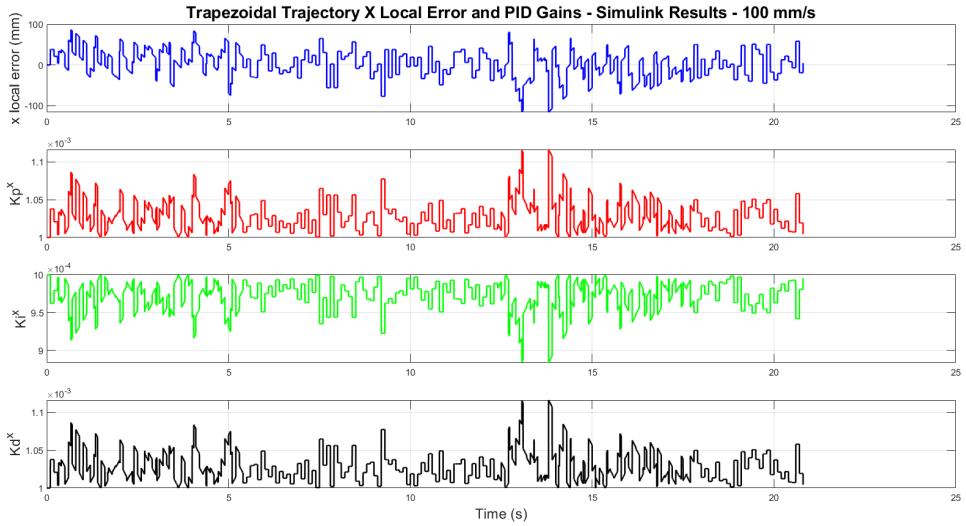

Figure A.2. Simulated local error in x and adaptive PID gain response on trapezoidal path at 100 mm/s.

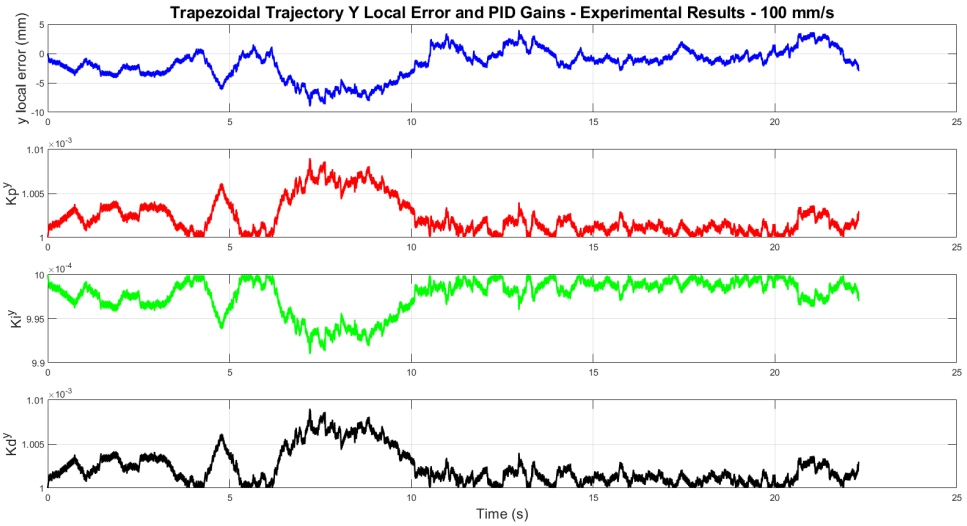

Figure A.3. Experimental local error in y and adaptive PID gain response on trapezoidal path at 100 mm/s.

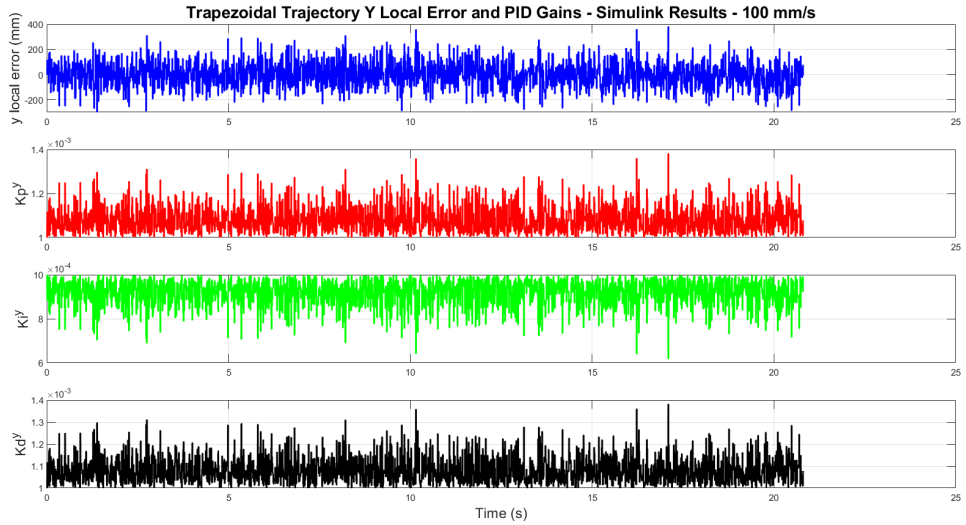

Figure A.4. Simulated local error in y and adaptive PID gain response on trapezoidal path at 100 mm/s.

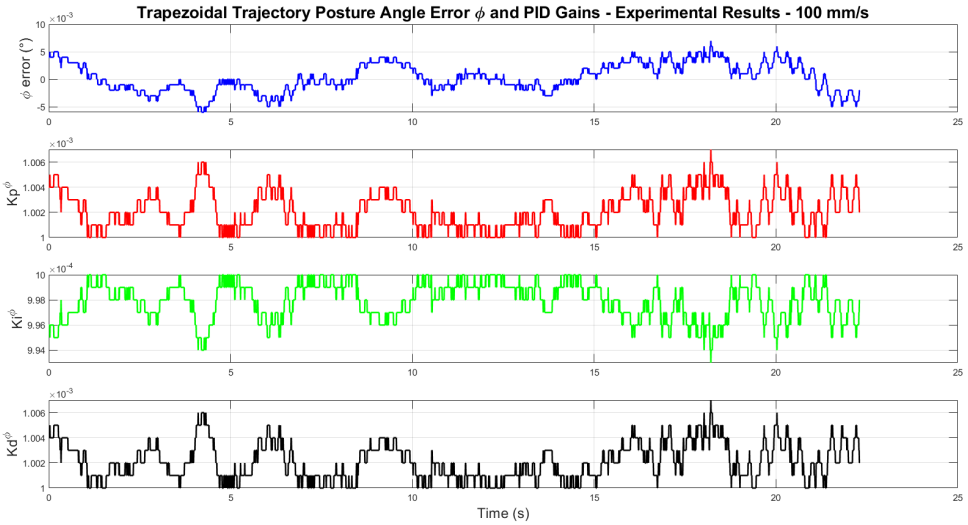

Figure A.5. Experimental local error in ϕ and adaptive PID gain response on trapezoidal path at 100 mm/s.

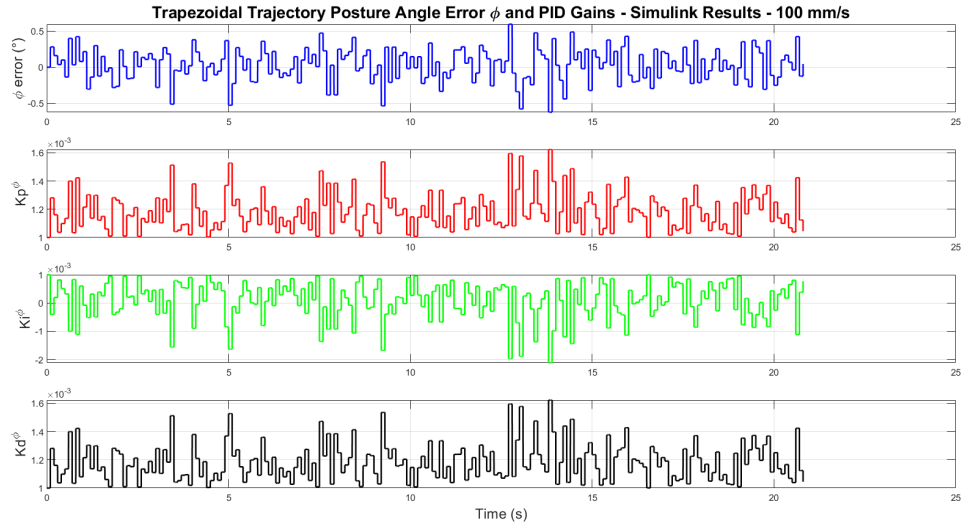

Figure A.6. Simulated local error in ϕ and adaptive PID gain response on trapezoidal path at 100 mm/s.

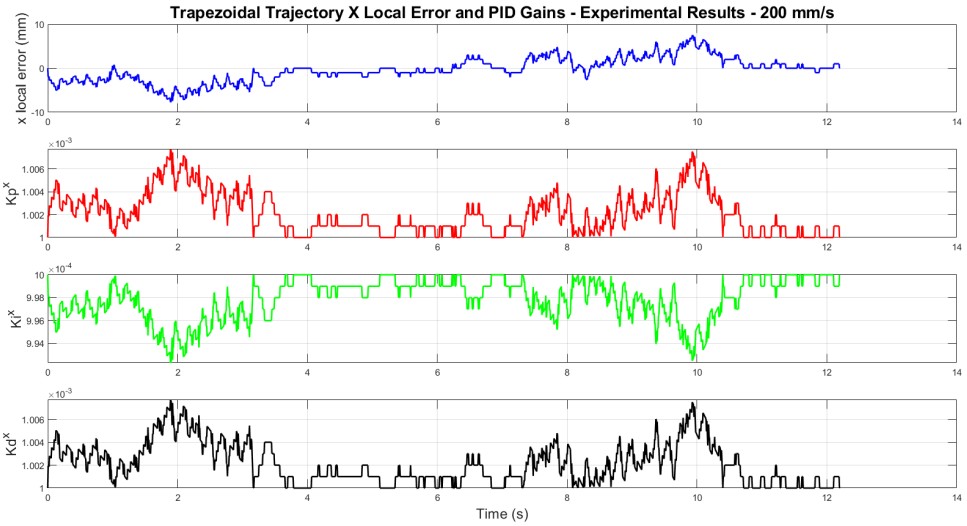

Figure A.7. Experimental local error in x and adaptive PID gain response on trapezoidal path at 200 mm/s.

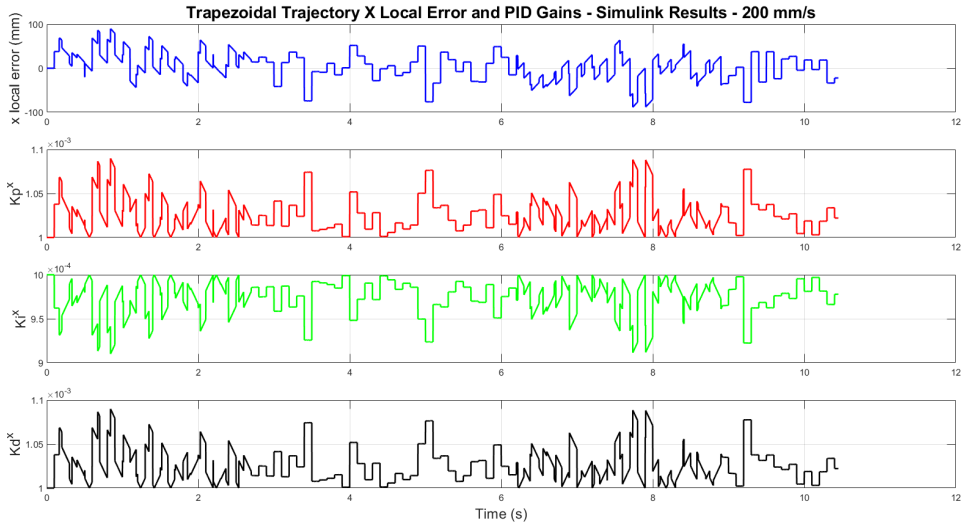

Figure A.8. Simulated local error in x and adaptive PID gain response on trapezoidal path at 200 mm/s.

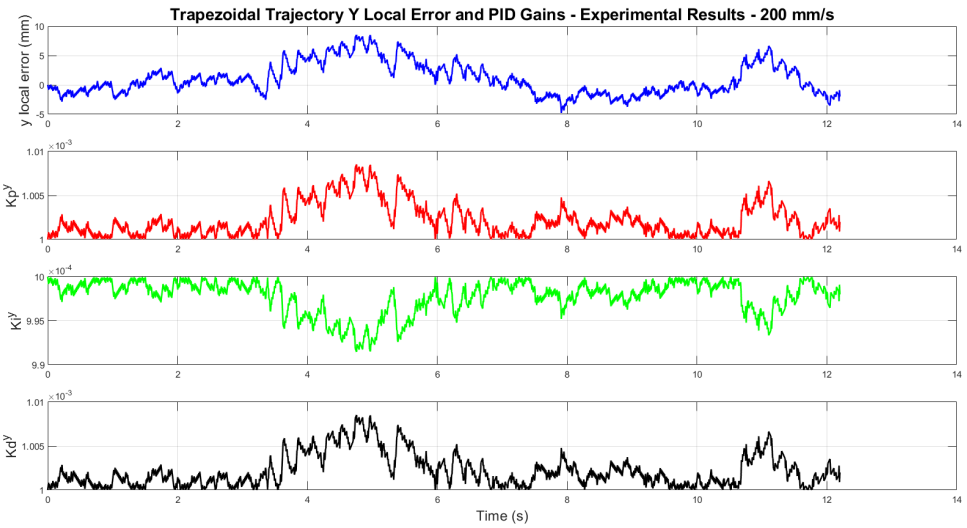

Figure A.9. Experimental local error in y and adaptive PID gain response on trapezoidal path at 200 mm/s.

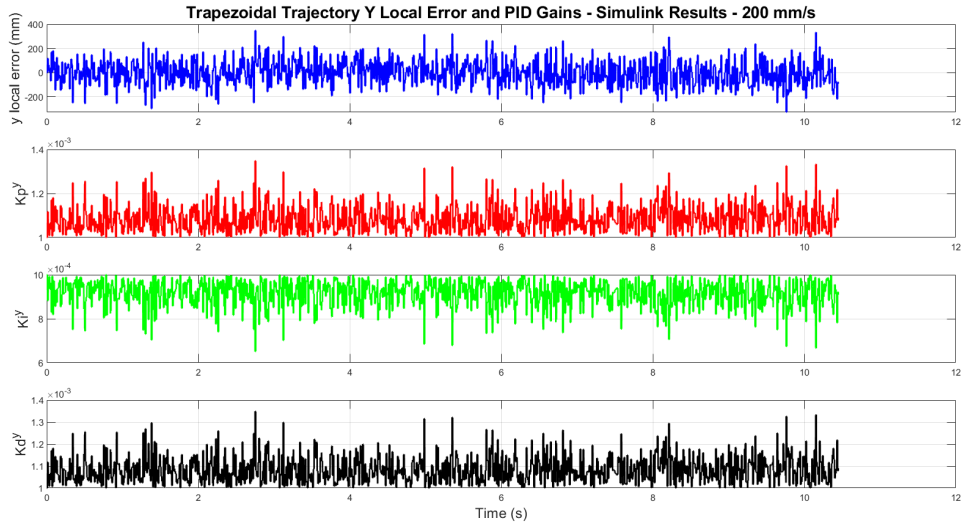

Figure A.10. Simulated local error in y and adaptive PID gain response on trapezoidal path at 200 mm/s.

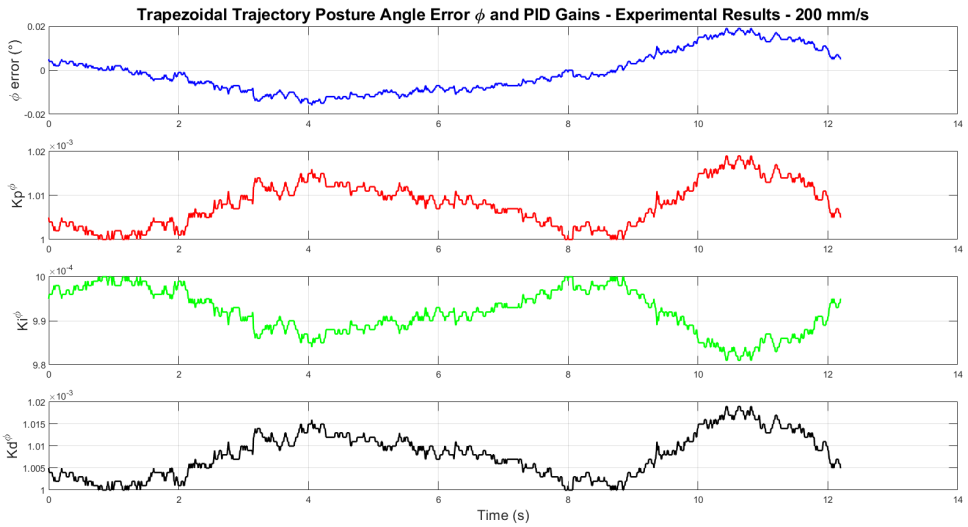

Figure A.11. Experimental local error in ϕ and adaptive PID gain response on trapezoidal path at 200 mm/s.

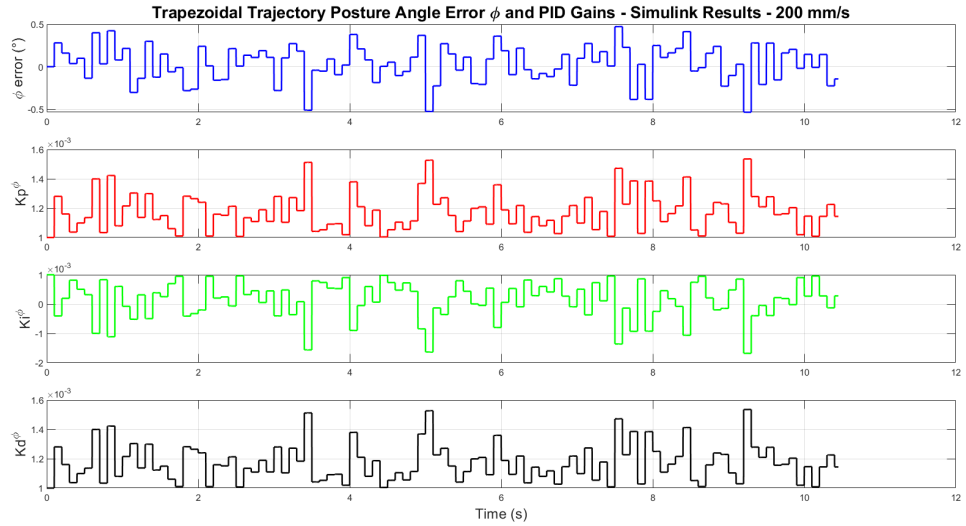

Figure A.12. Simulated local error in ϕ and adaptive PID gain response on trapezoidal path at 200 mm/s.

A.2. Square Path

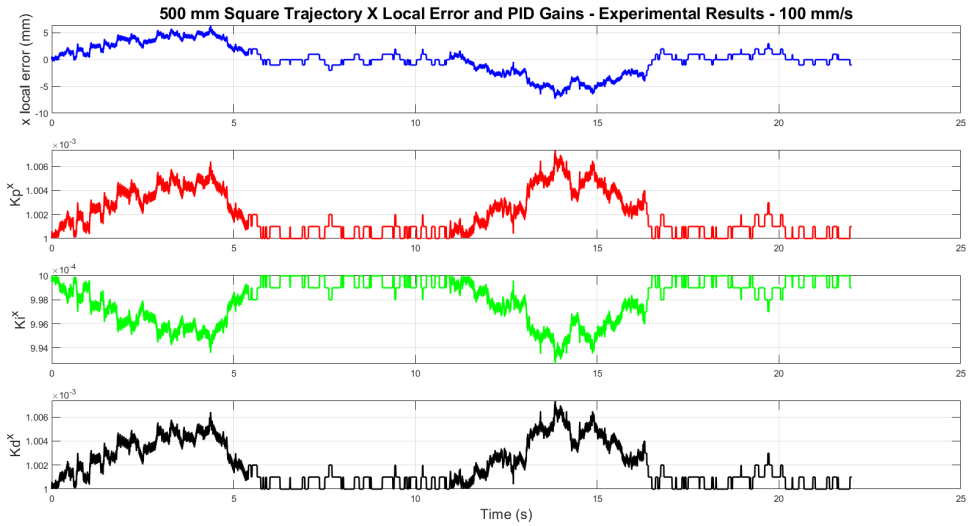

Figure A.13. Experimental local error in x and adaptive PID gain response on square path at 100 mm/s.

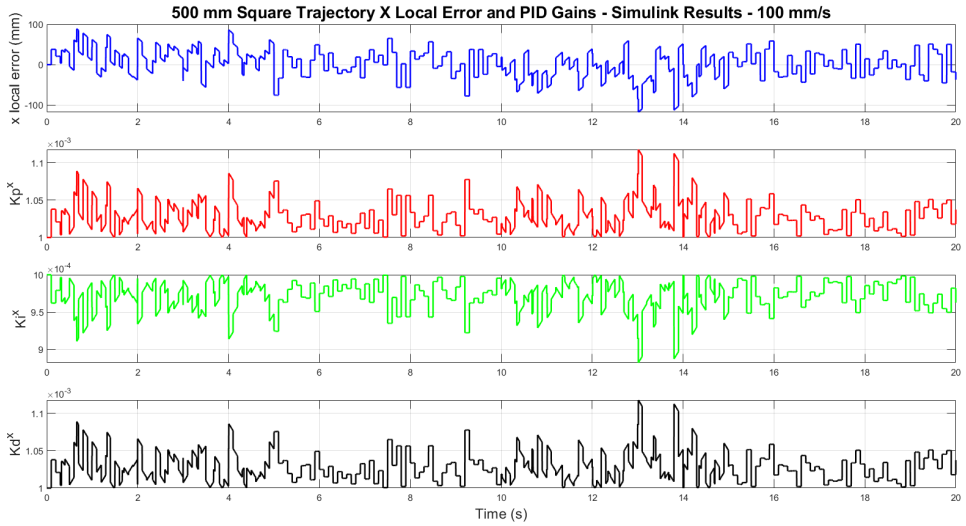

Figure A.14. Simulated local error in x and adaptive PID gain response on square path at 100 mm/s.

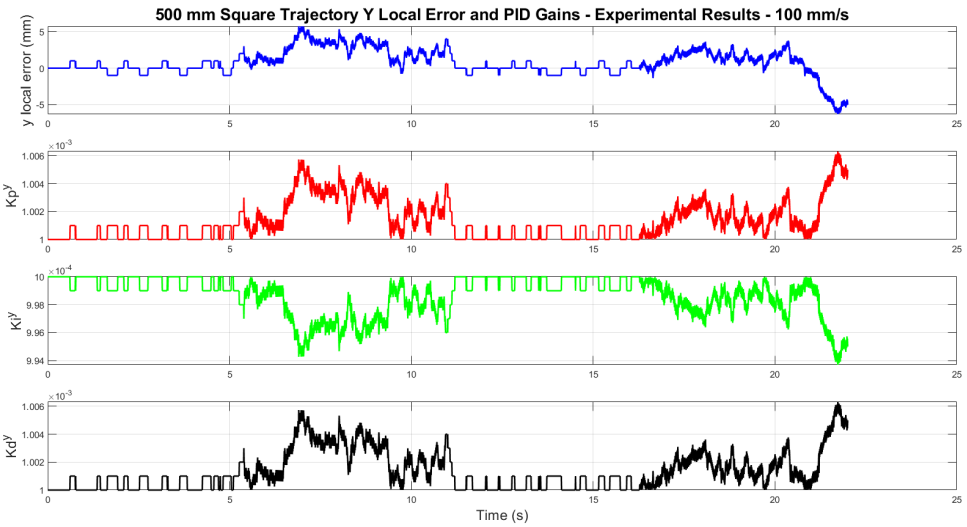

Figure A.15. Experimental local error in y and adaptive PID gain response on square path at 100 mm/s.

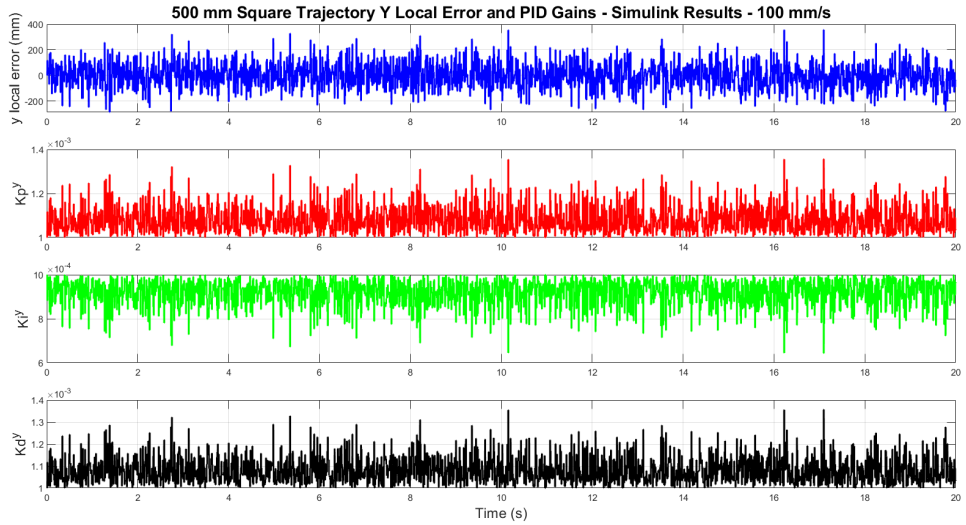

Figure A.16. Simulated local error in y and adaptive PID gain response on square path at 100 mm/s.

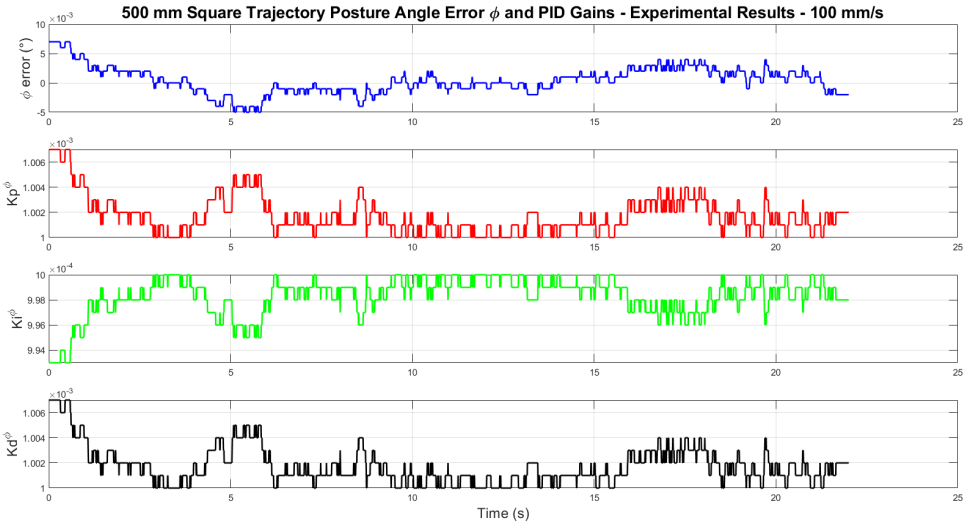

Figure A.17. Experimental local error in ϕ and adaptive PID gain response on square path at 100 mm/s.

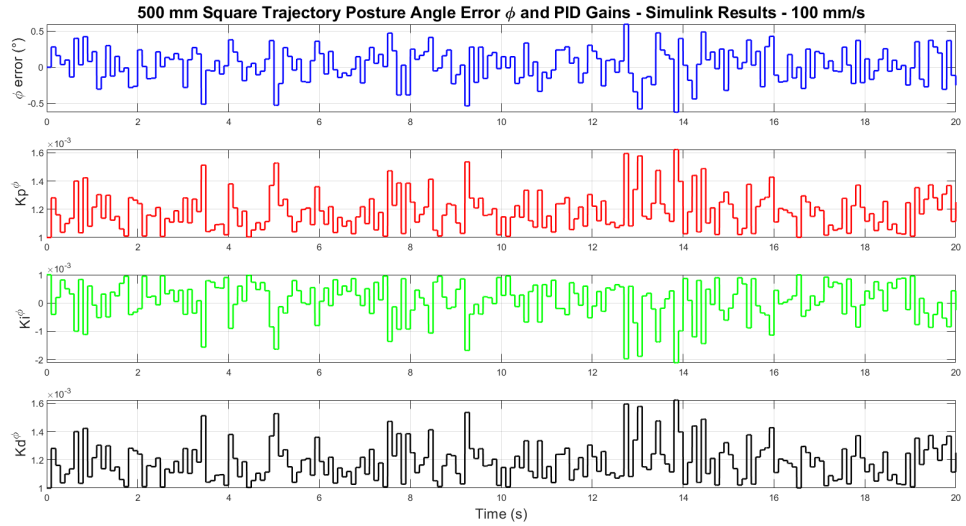

Figure A.18. Simulated local error in ϕ and adaptive PID gain response on square path at 100 mm/s.

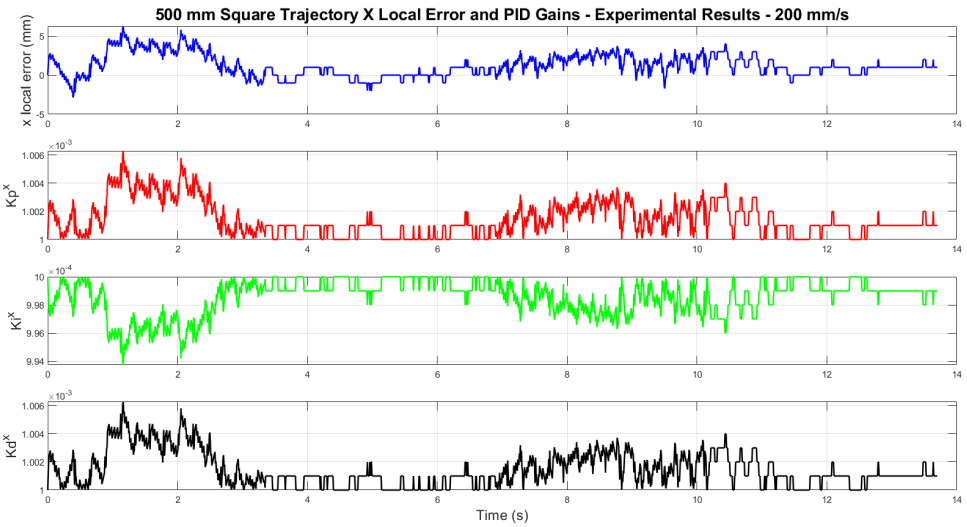

Figure A.19. Experimental local error in x and adaptive PID gain response on square path at 200 mm/s.

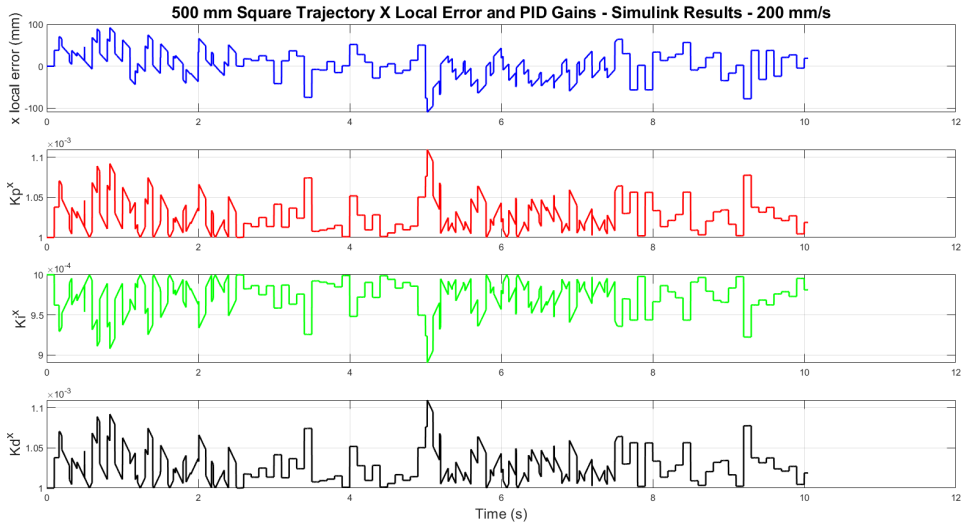

Figure A.20. Simulated local error in x and adaptive PID gain response on square path at 200 mm/s.

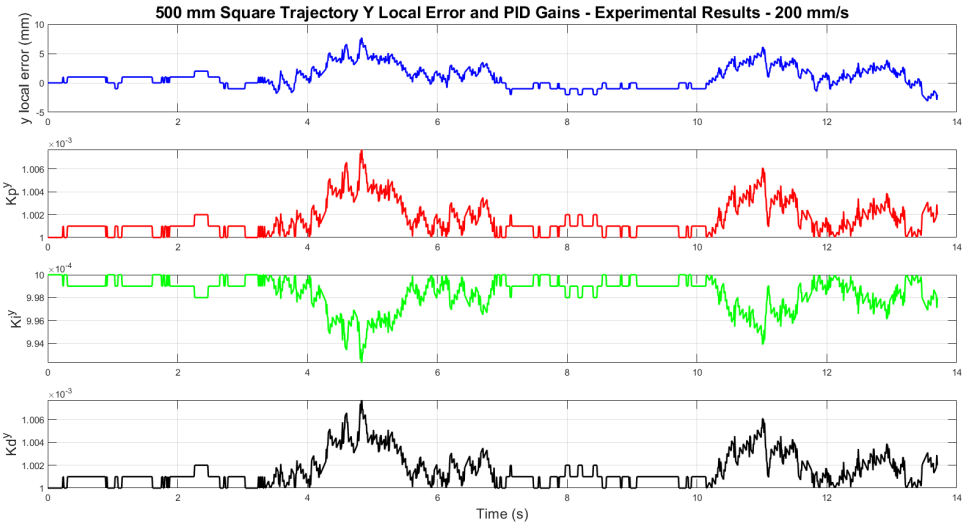

Figure A.21. Experimental local error in y and adaptive PID gain response on square path at 200 mm/s.

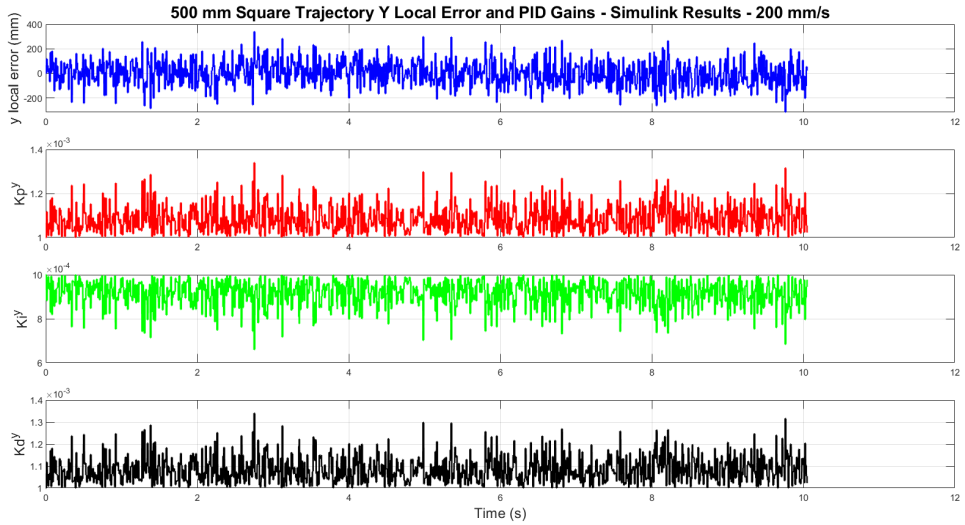

Figure A.22. Simulated local error in y and adaptive PID gain response on square path at 200 mm/s.

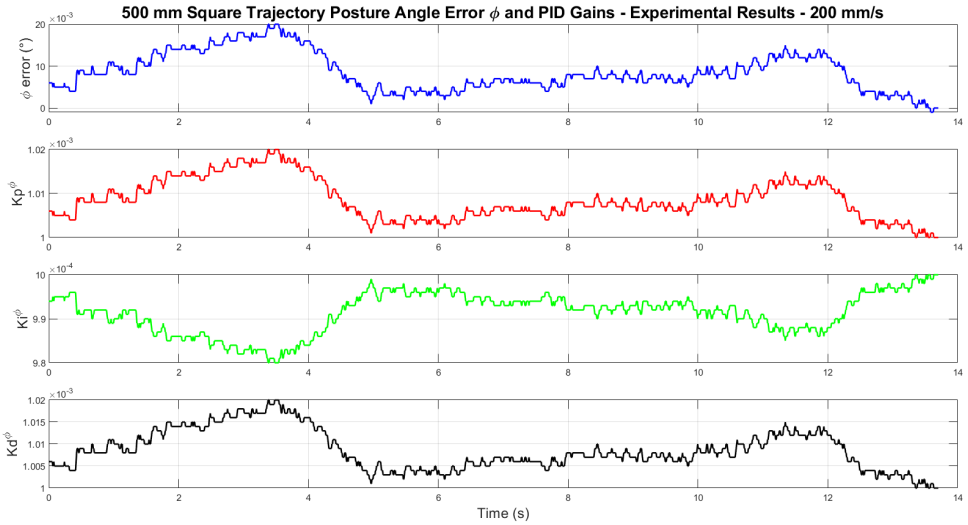

Figure A.23. Experimental local error in ϕ and adaptive PID gain response on square path at 200 mm/s.

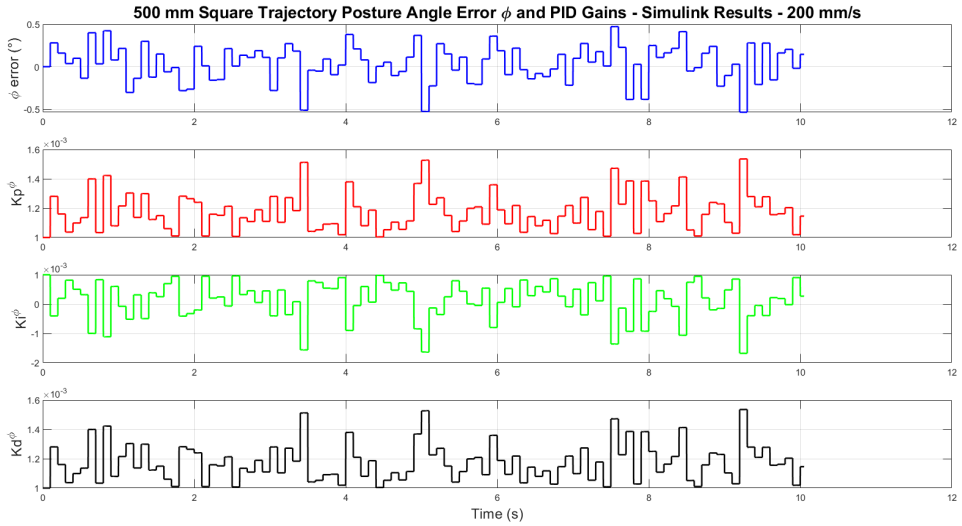

Figure A.24. Simulated local error in ϕ and adaptive PID gain response on square path at 200 mm/s.

A.3. Circular Path

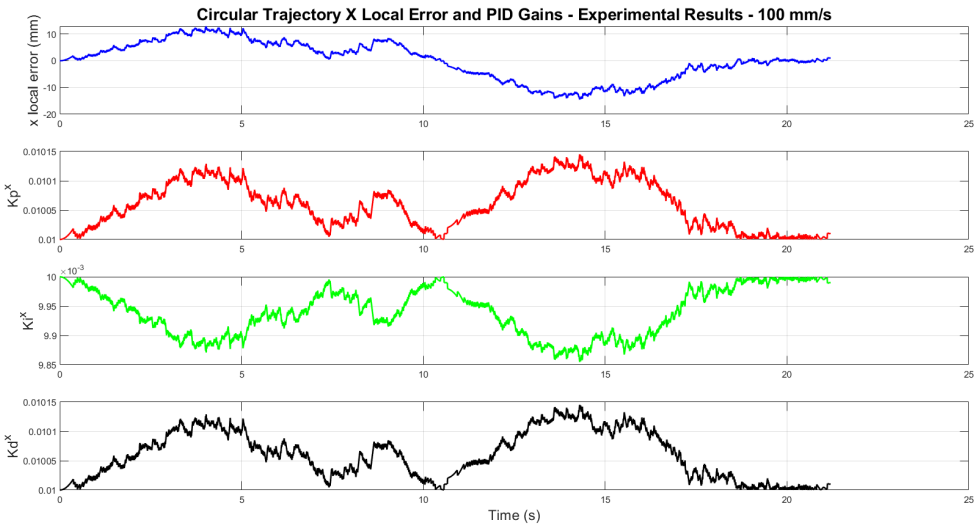

Figure A.25. Experimental local error in x and adaptive PID gain response on circular path at 100 mm/s.

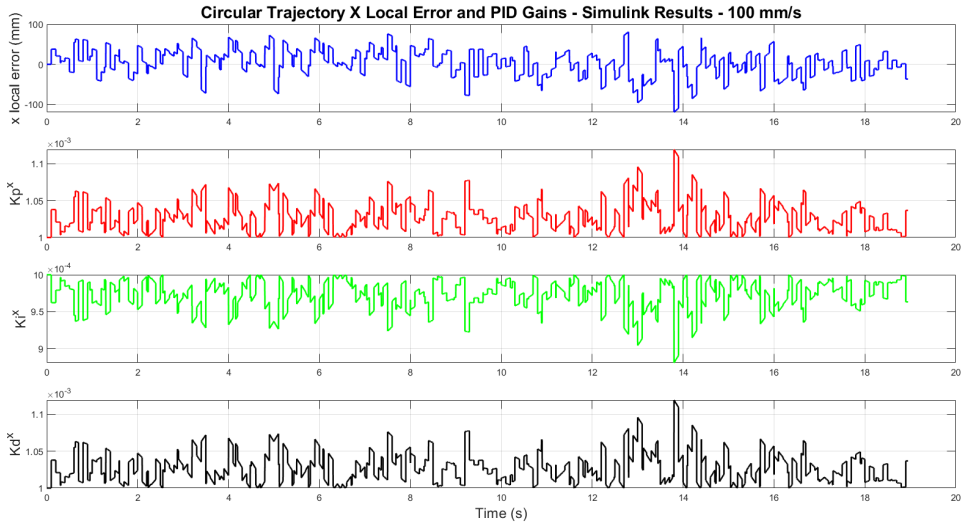

Figure A.26. Simulated local error in x and adaptive PID gain response on circular path at 100 mm/s.

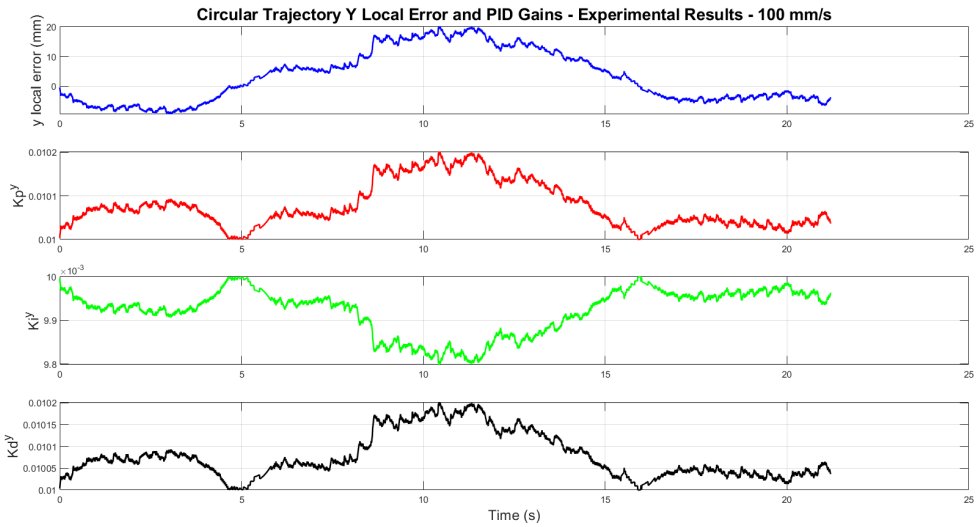

Figure A.27. Experimental local error in y and adaptive PID gain response on circular path at 100 mm/s.

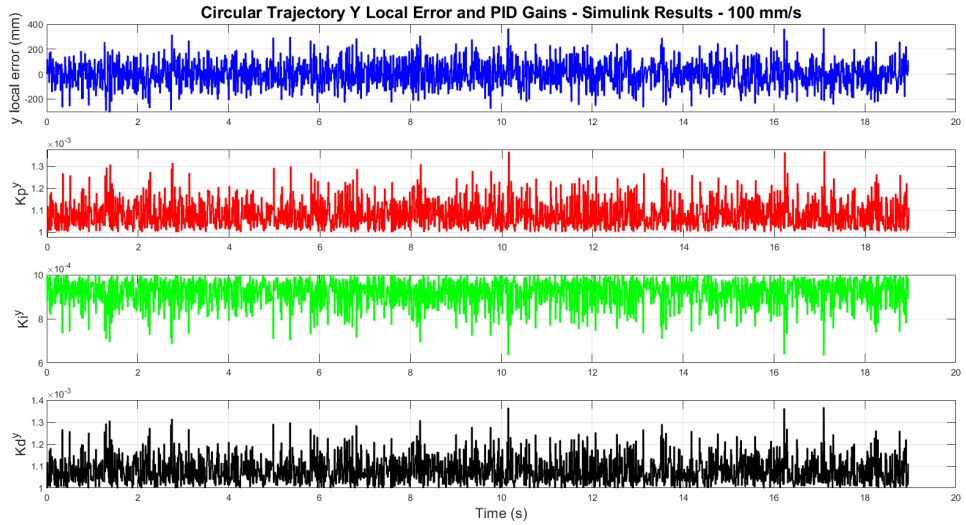

Figure A.28. Simulated local error in y and adaptive PID gain response on circular path at 100 mm/s.

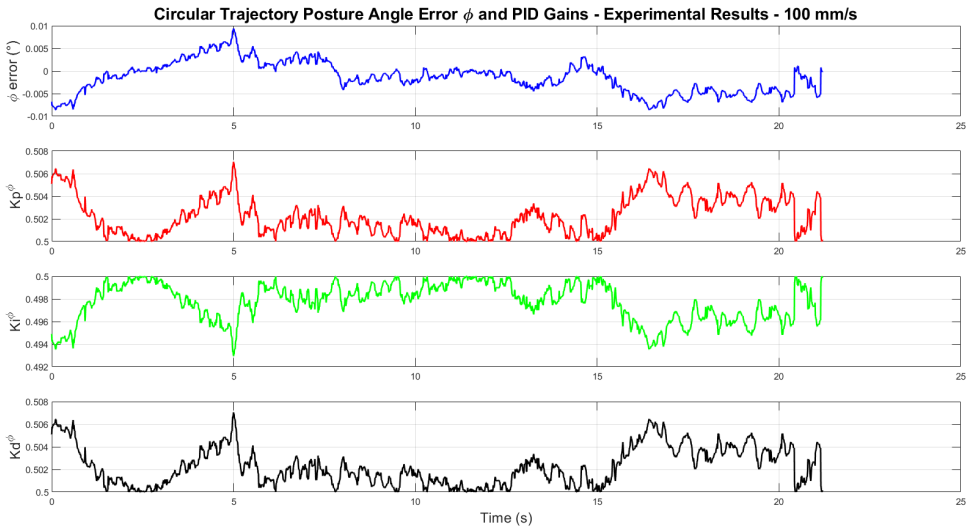

Figure A.29. Experimental local error in ϕ and adaptive PID gain response on circular path at 100 mm/s.

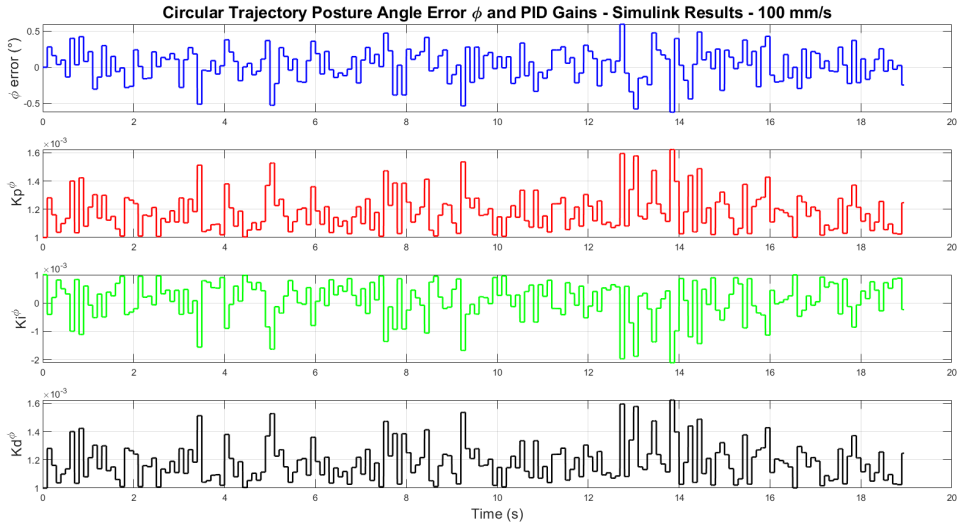

Figure A.30. Simulated local error in ϕ and adaptive PID gain response on circular path at 100 mm/s.

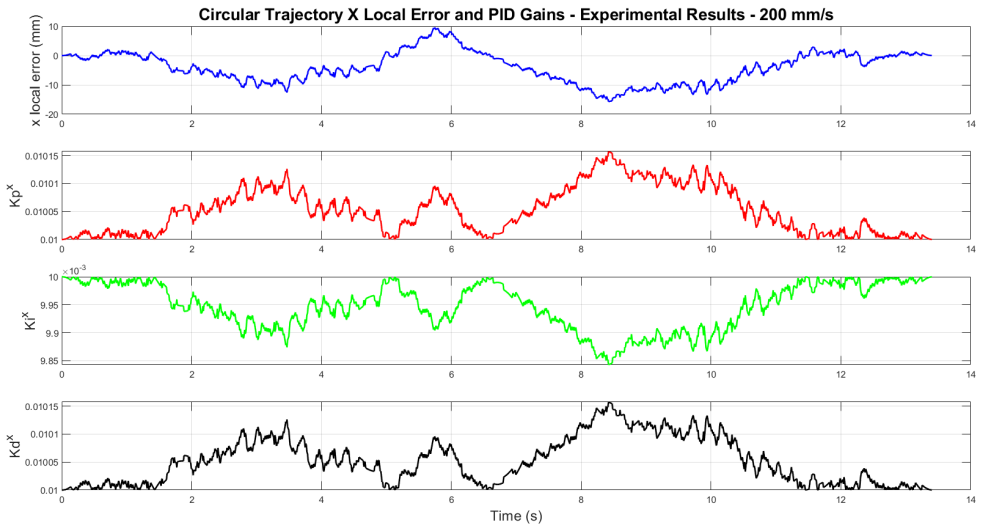

Figure A.31. Experimental local error in x and adaptive PID gain response on circular path at 200 mm/s.

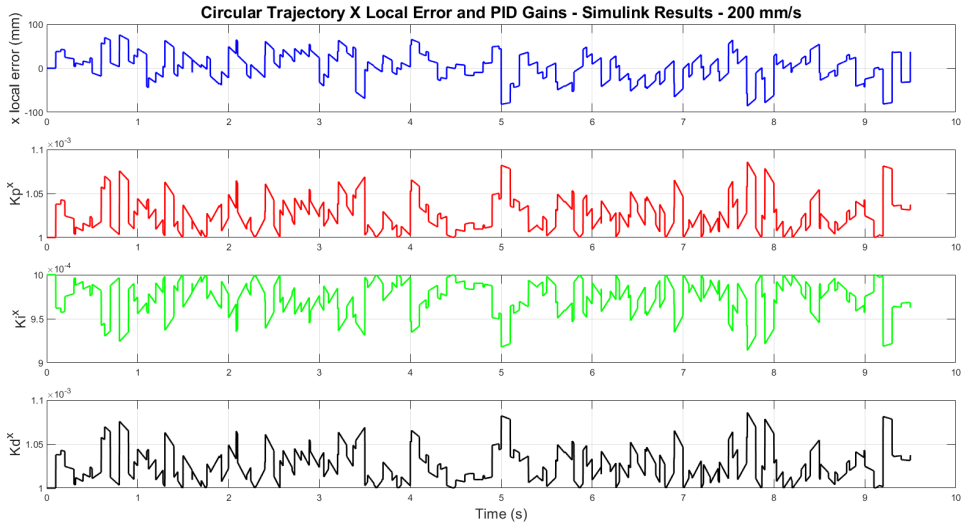

Figure A.32. Simulated local error in x and adaptive PID gain response on circular path at 200 mm/s.

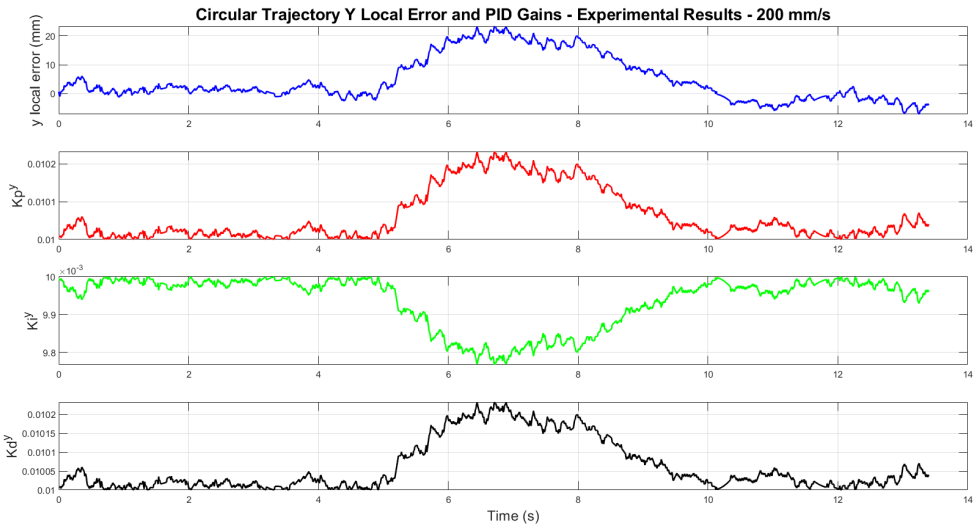

Figure A.33. Experimental local error in y and adaptive PID gain response on circular path at 200 mm/s.

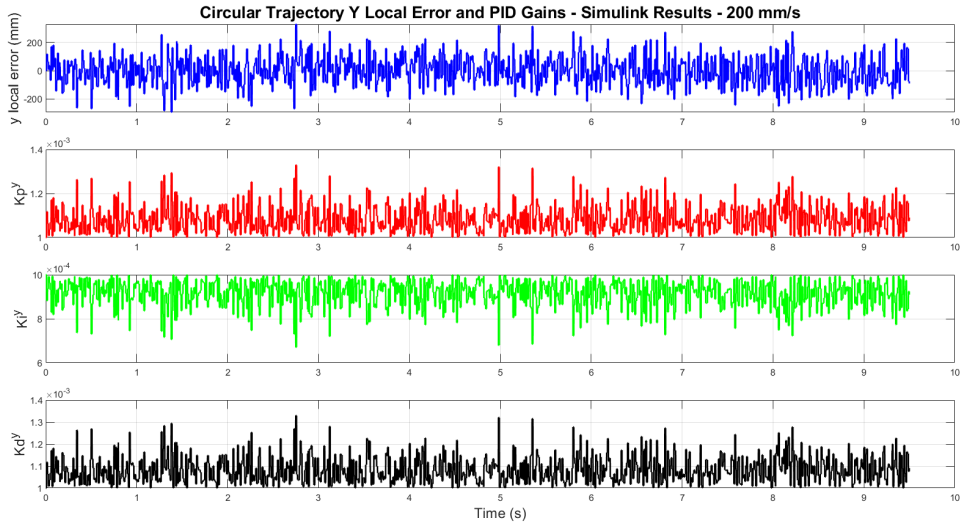

Figure A.34. Simulated local error in y and adaptive PID gain response on circular path at 200 mm/s.

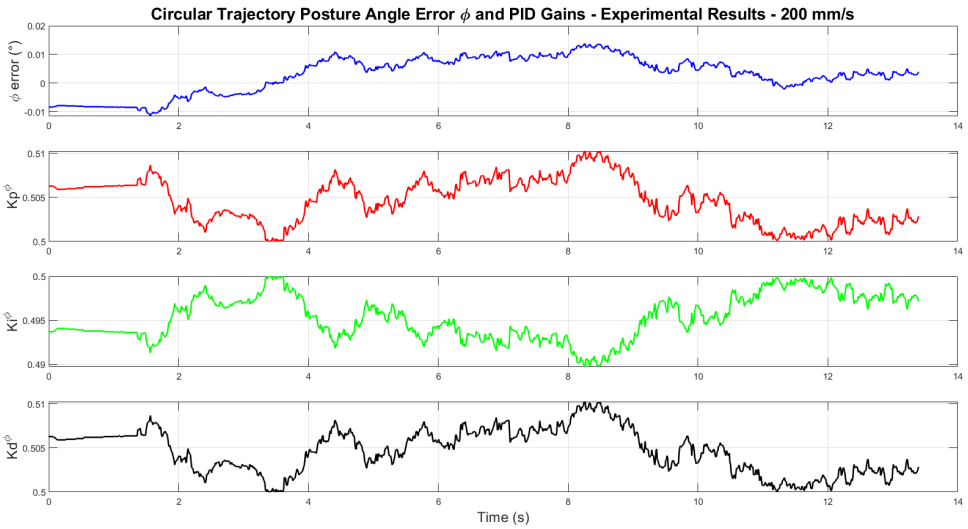

Figure A.35. Experimental local error in ϕ and adaptive PID gain response on circular path at 200 mm/s.

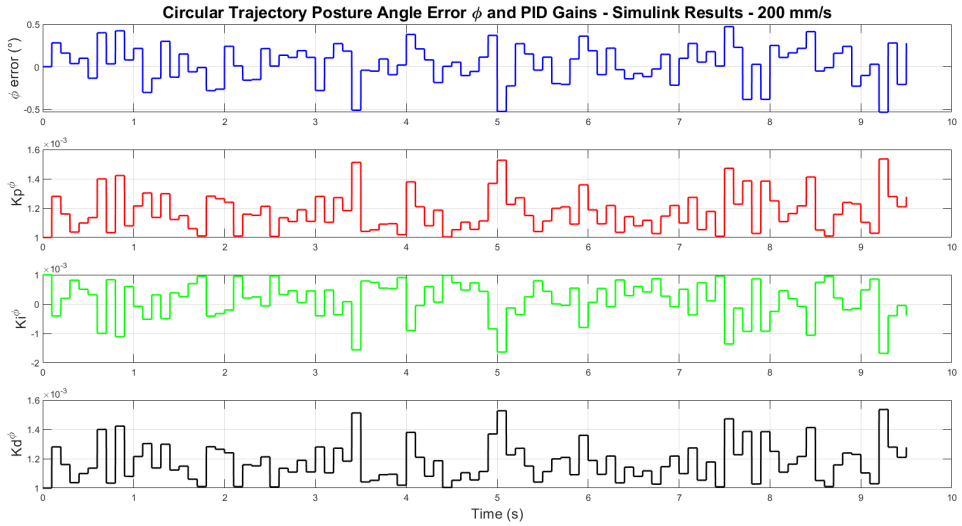

Figure A.36. Simulated local error in ϕ and adaptive PID gain response on circular path at 200 mm/s.