Cooperative Path Planning Implementation for a Robotic Manipulator and Magnetic Levitated Planar Actuator

by

Sebastian Viancha

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2026

© Sebastian Viancha 2026

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Magnetically levitated planar actuators (MLPAs) offer contactless motion, high precision, and kinematic flexibility, making them promising positioning platforms for industrial manufacturing applications such as continuous-seam welding, assembly, and inspection. However, their integration in cooperative tasks with robotic manipulators remains largely unexplored due to inherent controllability complexity and massive workspace. This thesis presents a comprehensive framework to enhance the control, operational range, and cooperative capabilities of a disc-magnet MLPA operating alongside a 4 degrees of freedom (DOF) robotic manipulator. To improve the overall standalone magnetic levitation system, the current implementation's control cycle is enhanced with a discrete Kalman Filter estimation and prediction sequence to provide smooth, optimal state estimation. To close the gap from uncertainties derived from device manufacturing, a mapping interpolation strategy is incorporated into the calibration workflow to increase the certainty of the calculated magnetic fields across the system. To expand the effective operational range, an enhanced magnetic force and torque dataset is generated to train a Dual Neural Network architecture, enabling extended actuation range across the entire levitation workspace without long loading times or high memory usage. The overall improvements and modifications are experimentally tested, obtaining similar results to the current implementation on the positional coordinates, but improved tracking metrics on the disc-magnet MLPA of $\pm$ 50% on all the tilting axes for a tested circular motion.

Building upon these core control enhancements, a cooperative framework is proposed in which the MLPA mover and the robotic arm execute a time-optimized planned path to complete complex trajectories with strict orientation constraints. The path-planning algorithm uses a geometrically modeled arm inverse kinematics and a filtered discrete arm workspace to generate a global search graph, resolving the system redundancies via Jacobian-based state configurations. To validate the flexibility of the proposed cooperative MLPA system, a trajectory-aligned baseline is developed in which the mover remains statically anchored during interaction. Experimental validation on a real MLPA system demonstrated that for standard geometries, the cooperative planner reduced completion time by almost 20% compared to the baseline, achieving positional and angular Root Mean Square (RMS) errors of 0.27 mm and 0.01°, respectively. Furthermore, evaluations on complex continuous-arc trajectories confirmed the critical necessity of cooperative mover–arm operation to synthesize feasible interaction solutions, conclusively demonstrating the potential of MLPA–robotic arm integration to expand the flexibility of kinematically limited industrial manufacturing lines.

Acknowledgements

I am sincerely thankful to my supervisor, Professor Mir Behrad Khamesee, for his continuous guidance, insightful feedback, and invaluable mentorship. His support has been instrumental in shaping this project. I am especially grateful for the opportunity to work with advanced technologies while deepening my understanding of a field that I have grown genuinely passionate about.

I would also like to extend my appreciation to the planar maglev research team and colleagues: Yang (Bob) Wang, Mahdi Naserzad, Yufeng (Hugo) Huang, Ridwan Bari, Andersen Borja, Yanze (Kevin) Li, and Ali Shams. Your mentorship, advice, and friendship have made these past two years both enriching and enjoyable. I am equally grateful to the previous members of the Maglev Lab, particularly Dr. Zhenchuan (Daniel) Xu, Dr. Chanuphon (Cha) Trakarnchaiyo, Dr. Xiaodong Zhang, and Curtis Stewart, whose foundational work made this project possible.

I would also like to acknowledge the support of the RoboHub staff, including Robert Wagner, Professor Brandon DeHart, and Alexander Werner, as well as the RoboHub Director, Professor William Melek, for their contributions to the development and support of the maglev platform (Magfloor). It has been a privilege to witness and contribute to such an impressive engineering system.

My sincere thanks also go to my landlord, Mercedes Sarmiento, whom I now consider family. Her kindness, generosity, and companionship made my time at the university both comfortable and meaningful.

Finally, I gratefully acknowledge the financial support provided through the Buitrago Award, made possible by the generosity of Professional Engineer Jorge Buitrago. Your support has not only enabled this work but has also had a lasting impact on my personal and academic growth. While I may have only been able to contact you through non-physical channels, I want you to know that you will always be in my memory.

Dedication

I would like to express my deepest gratitude to my parents by dedicating this work to them. Although they were not physically present during this journey, their constant communication, unwavering encouragement, and continued prayers sustained me throughout. It is through their enduring support that I was able to carry this project to completion.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

Dedication v

List of Figures x

List of Tables xiii

List of Abbreviations xiv

1 Introduction 1

1.1 Magnetic Levitated Planar Actuators and Manufacturing Industry 1

1.2 Flexible Manufacturing Systems 4

1.2.1 Flexible Manufacturing and Manufacturing Transportation Systems 4

1.2.2 Cooperative Robotic Systems in Manufacturing 7

1.3 Objectives and Contributions 9

1.3.1 Main Objectives 10

1.3.2 Contributions 11

1.4 Thesis Outline 12

2 Literature review **13**

2.1 Magnetic Levitated Planar Actuators 13

2.1.1 Developed MLPA systems review 13

2.1.2 MLPA modeling methodologies 16

2.2 Robotic manipulators 21

2.2.1 Common robotic manipulators in the industry 22

2.2.2 Robotic manipulators modeling strategies 24

3 System Setup **30**

3.1 Overall Testbench Setup 30

3.2 Hardware Subsystems 32

3.2.1 Mover System 32

3.2.2 Manipulator model and parameters 34

3.3 Control and Communication Architecture 37

4 Implementation and Methodology Improvements **40**

4.1 System Enhancements and Calibration 40

4.1.1 Coil Position Mapping and Calibration 40

4.1.2 Kalman Filter State Estimation 45

4.2 Wrench implementation methodology 49

4.2.1 Limitations of the Rigid Array Lookup Table 49

4.2.2 Data Generation via Sampling 50

4.2.3 Dual Neural Network Architecture and Training 52

4.2.4 Controller Integration and Validation 53

4.3 Interaction Path Generation 55

4.3.1 Mesh-Based Trajectory Extraction 55

4.4 Cooperative Path Planning and Workspace Filtering 58

4.4.1 Definition of the Interaction Process 58

4.4.2 Workspace Filtering and Testbench Constraints 60
4.4.3 Anchor Generation and Selection 60
4.4.4 Optimization and Trajectory Generation 62
4.4.5 Arm-only Baseline Configuration Search 65

5 Experimental Results and Discussion **66**
5.1 Mover Model Calibration and State Estimation 66
5.1.1 Path Tracking Performance 67
5.1.2 Tracking Error Metrics 71
5.1.3 Power Consumption Analysis 73
5.2 Cooperative Path Planning Evaluation 74
5.2.1 Interaction Cases and Experimental Execution 74
5.2.2 Path 1: Circle Trajectory 74
5.2.3 Path 2: Line Trajectory 77
5.2.4 Path 3: Arc Trajectory 79
5.2.5 Individual Tracking Performance 81

6 Conclusions **82**
6.1 Future Works 83

References **85**

APPENDICES **92**

A Dual Neural Network Structure and Wrench Modeling **93**
A.1 Dataset Generation 93
A.2 Wrench Quarter Mapping Logic 95
A.3 Neural Network Structure and Topology Selection 98
A.4 Extrapolation Capabilities and LUT Comparison 99

B Robot Manipulator Vicon Calibration **101**

B.1 Robot Arm Base Vicon Object 101

B.2 End-Effector Forward Kinematics Calibration 103

List of Figures

1.1 Testbench overview with the four disc-magnet mover and the Vicon tracking system. . . . 3
1.2 Transportation systems in manufacturing 5
1.3 Commercially available small factor MLPA systems. . . . 6
1.4 Maglev/MLPA application for direct energy deposition operations. . . . 7

2.1 1-D Halbach arrays for mover design 14
2.2 Single DMM during levitation 16
2.3 Analytical calculation of the magnetic field 17
2.4 Magnodes method on a cuboidal magnet 19
2.5 Magnodes method on a disc magnet 19
2.6 Robotic manipulator PUMA 560 22
2.7 Photo of an Epson SCARA T6. . . . 23
2.8 3-link planar manipulator scheme 24
2.9 Discretized reachable workspace of a 3-link manipulator 29

3.1 Testbench stator arrangement. The coils and amplifiers are magnified for clarity. . . . 31
3.2 Single coil geometry and profile. . . . 31
3.3 Overall communication Testbench setup. . . . 32
3.4 Mover schematic. . . . 33
3.5 Simulated static power consumption. . . . 33

3.6 Igus 4-DOF manipulator. 34
3.7 Overall communication flow for manipulator-Testbench cooperation 37
3.8 Master-slave controller diagram 38

4.1 Interface for stator coil coordinates calibration. 42
4.2 Bilinear coil mapping 44
4.3 Thin-plate surface for coil mapping 45
4.4 Original model velocity state estimation result 46
4.5 Enhanced model velocity state prediction with Kalman Filter 49
4.6 Original model LUT range boundaries 50
4.7 Comparison between LUT and dual-NN wrench modelling 51
4.8 Change of Z force acting on the mover with respect to air gap. 52
4.9 RMSE comparison across the entire workspace range 55
4.10 Definition of the waypoint coordinates with respect to the target frame. . . 56
4.11 Definition of the orientation trajectory 57
4.12 Assembly of mover, target and trajectory path representation. 57
4.13 Proposed cooperative interaction sequence 59
4.14 Levitation area arm workspace filtering 61
4.15 DMM and a target object with an interaction circular path 62

5.1 Tracking results for a circle path levitation 68
5.2 Experimental results for a rectangle path levitation 68
5.3 Experimental results for an in-place rotation with high tilting 69
5.4 Experimental results for a circle path trajectory with high air gap 70
5.5 Experimental results for a rectangle path trajectory with high air gap . . . 70
5.6 Multiple recorded circular and rectangular paths during high air gap motion. 72
5.7 Power consumption during motion trajectories. 73
5.8 Experimental tracking results for the circle interaction trajectory for both models 76

5.9 Experimental tracking results for a line path interaction trajectory for both models 78
5.10 Proposed curve path for cooperative interaction. 79
5.11 Experimental tracking results for the arc path interaction trajectory 80

A.1 Generated dataset distribution of dual-NN setup 94
A.2 Visual representation of the collision filter for dataset generation 95
A.3 Mover quadrant wrapping during levitation 96
A.4 Training and validation loss values for the dual-NN setup 99

B.1 Setup for manipulator base markers recognition 102

List of Tables

2.1 Denavit-Hartenberg D-H parameters for the 3R planar manipulator. 25

3.1 DMM parameters 33
3.2 PID Parameters 33
3.3 D-H Parameters for the 4-DOF Manipulator 34

4.1 MLPA Kalman Filter Covariances 48
4.2 Total 16-Coil Error Comparison Relative to Analytical Baseline 54

5.1 Parameters of the Circular Long Stroke Path 67
5.2 Parameters of the Rectangular Long Stroke Path 67
5.3 Tracking Error Metrics Comparison 71
5.4 Cooperative Execution Tracking Accuracy (Aggregated Runs Dataset) . . . 81

A.1 Comparison of Evaluated Neural Network Topologies (Single Coil vs Analytical) 98
A.2 Extrapolation Performance Comparison Outside LUT Range ($Z \in [30, 60]$ mm, Tilts $\pm 25^\circ$) 100

List of Abbreviations

AGV automated guided vehicle 4, 5

AMR autonomous mobile robot 4, 5

DED direct energy deposition 8

DMM disc magnet mover 2, 3, 9, 33, 46, 82

DNN deep neural network 49, 53

DOF degree of freedom 2, 3, 7, 9, 46

FK forward kinematics 60

FMS flexible manufacturing system 4, 7

KF Kalman filter 45, 46

LUT lookup table 49, 53

MLPA magnetically levitated planar actuator 2–5, 7, 9, 40, 49, 73, 83

MLPM magnetic levitated planar motor 6

Chapter 1

Introduction

1.1 Magnetic Levitated Planar Actuators and Manufacturing Industry

The growing demand for precision, adaptability, and sterile operations in modern industrial environments has spurred research into contactless actuation technologies. Among these, magnetic levitation (Maglev) has been extensively studied in recent decades. The primary objective of a Maglev system is to use a magnetic field to continuously control and suspend objects within a defined volume, thereby eliminating mechanical connections between moving parts. In general, there are two ways to achieve "levitation": 1) using attractive magnetic configurations [43] or 2) using repulsive forces [28]. The choice between them depends on application-specific constraints, including clearance, maintenance, and safety requirements for the levitated object. Common applications of Maglev technology include long-range transportation; magnetically levitated trains achieve frictionless movement at remarkable speeds, with the Shanghai Maglev currently holding the commercial speed record at 460 km/h [68]. In aerospace, magnetically levitated bearings have replaced conventional rolling bearings in turbines and satellite attitude control wheels, eliminating the need for lubrication and enabling active control of bearing position [51]. In medicine, actuated forces from a distance have been explored for wireless actuation of surgical devices and drug containers inside the human body [30]. The versatility of Maglev across such a range of domains underscores its role not only as a transport technology but also as a general mechanism for contactless interaction.

Within the broader field of Maglev technology, a specialized branch focuses on planar

actuation, known as magnetically levitated planar actuators (MLPAs). Following Earnshaw's theorem, which states the impossibility of achieving a static stable equilibrium using only fixed magnetostatic fields, researchers often introduce diamagnetic materials such as superconductors [16] or paramagnetic objects [34] to obtain passive levitation. However, a common solution for MLPAs is dynamic stabilization via a feedback control law. This approach allows the use of magnetostatic elements such as permanent magnets and DC-driven coils, which are preferred due to their ease of modeling, maintenance, and implementation. Defining the levitated object and the static platform as the **mover** and **stator**, respectively, these systems are typically found in three main configurations: a permanent magnet array mover with a coil stator [57], a coil array mover with a magnet array stator [42], or a magnet array for both [73].

Due to their contactless operation, MLPAs are particularly compelling for industrial applications sensitive to mechanical wear or requiring cleanroom standards, such as semiconductor wafer transport [32] and high-precision photolithography [29]. The elimination of friction between surfaces results in more energy-efficient operation, reduced noise and vibration, and flexible omnidirectional product placement. Recently, with the increasing complexity of assembly automation tasks, MLPA technology has gained increased commercial recognition through significant industrial investments in platforms such as XPlanar [3] and Planar Motor [47]. Despite these commercial successes, current industrial setups commonly restrict the full potential of MLPAs when integrated with external robotic systems, even though the platform is inherently capable of six-degree of freedom (DOF) manipulation [70]. For example, in automated electronics assembly lines, MLPA platens are typically deployed as high-speed 2D conveyor shuttles to transport printed circuit boards (PCBs) or component fixtures between workstations, while overhead SCARA or 6-axis robotic arms execute all component insertion, micro-alignment, and orientation changes [48]. This operational paradigm treats the MLPA strictly as a flat, translational workpiece carrier rather than utilizing its full rotational and tilting capacities.

This reduced potential is a product of an engineering decision based on the mover design. In the literature, movers are predominantly categorized as either Halbach array [67] or disc magnet mover (DMM) actuators [59]. Halbach-array movers are widely implemented in commercial manufacturing because their magnetic flux is concentrated on the side facing the stator. However, they are constrained by small air gaps and severely restricted tilting angles, typically limited to $\pm 2°$. In contrast, DMMs achieve significantly larger air gaps and wider tilting ranges (up to $\pm 30°$) due to their lighter weight, making them better suited for applications requiring higher levels of orientation.

Because Halbach arrays have remained the industrial standard, the benefits of incorporating DMMs into the broader automation chain remain largely unexplored. While

lower-DOF manipulators like SCARA arms offer high operational speeds and excellent repeatability, they are inherently restricted to planar operations [22]. To overcome this, industry often resorts to integrating higher-DOF robotic arms; however, this approach is not only expensive but also increases operational error due to the added mass, joints, and compliance of the system. A clear research gap exists in using the out-of-plane tilting capability of a DMM to complement lower-DOF manipulators.

To bridge this gap, the primary objective of this research is to design, implement, and validate a path planner that couples a 4-DOF manipulator and a DMM within a unified optimization framework. In this architecture, the DMM exploits its wide tilting range to provide the missing out-of-plane motion required by the manipulator. To support this expanded operational volume, the system's modeling and control cycle is enhanced by integrating pre-levitation stator calibration and a Kalman filter. Additionally, a dual neural network structure is implemented to accurately estimate the magnetic forces acting on the mover across larger tilting angles, enabling complex manipulation tasks that are infeasible for the robotic arm alone. The experimental validation of this work is conducted on the MLPA Testbench located in the Maglev Microrobotics Laboratory at the University of Waterloo [65], as shown in Figure 1.1 and discussed in the system description in Section 3.1.

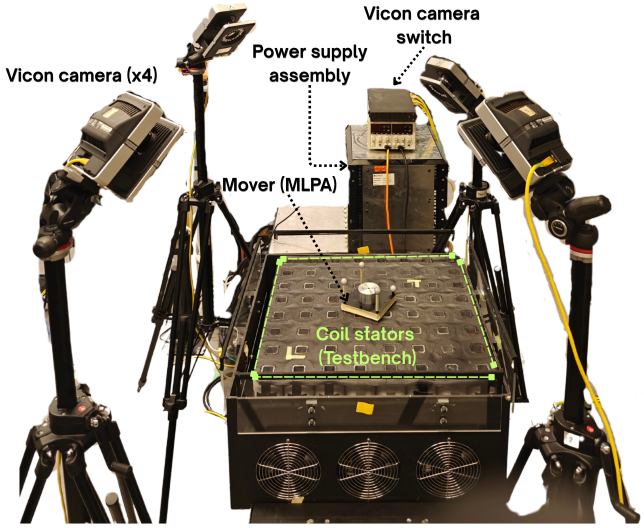


Figure 1.1: Testbench overview with the four disc-magnet mover and the Vicon tracking system.

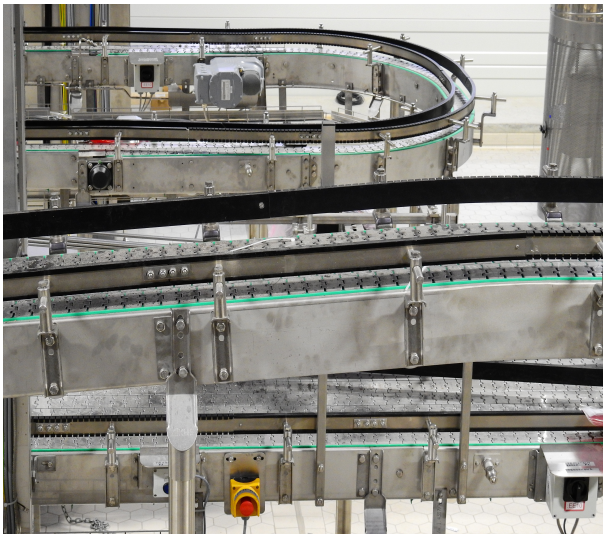
1.2 Flexible Manufacturing Systems

1.2.1 Flexible Manufacturing and Manufacturing Transportation Systems

Understanding the potential of implementing an MLPA in the automation industry requires explaining current production lines and the concept of flexible manufacturing. Flexible manufacturing systems (FMSs) represent an optimization approach designed to rapidly adapt production environments to operational changes, commercial and manufacturing metrics, and shifting work capacities [40]. To achieve this, a successful FMS typically addresses three main objectives. The first is routing flexibility, which allows products to be quickly rerouted through different or additional paths in the production line. Next is expansion flexibility, which dictates the system's modularity for expansion. Finally, material handling flexibility relates directly to the efficiency of the manipulator or the operator in performing the assigned task. To achieve optimality within an FMS, engineers analyze physical equipment configurations and perform iterative simulations and layout optimizations to obtain an improved manufacturing path [39]. However, the extent to which these optimized manufacturing paths can be realized in practice depends largely on the physical transportation system connecting the workstations.

As the most common conventional transport solution, mechanical conveyor belts introduce structural limitations that constrain the practical output of FMS optimization. Limited to linear motion, conveyor devices are mostly used with external devices to increase overall flexibility, such as pneumatic pushers, sweepers, roller actuators, or specialized branching, merging, and accumulation zones. Figure 1.2a shows a typical linear conveyor system. While these enhancements widen the range of possible manufacturing paths, they also increase the required operational footprint, which could have been used to install new workstations. Omnidirectional platforms attempt to resolve these single-axis constraints by enabling translational and rotational positioning of objects within a plane without requiring the aforementioned fixed pathway junctions. Nonetheless, both systems suffer from the required friction forces for operation, inducing excessive vibrations, high-frequency noises, and constant maintenance costs due to wear, in addition to being complex systems to disassemble and rearrange.

As a modern approach to the flexibility challenge, an attractive new transport technology has been studied for FMS integration. With increased research on unmanned vehicles, automated guided vehicles (AGVs) and autonomous mobile robots (AMRs) have added flexibility to conventional conveyor systems. By following a strict, predetermined route



(a) Conventional conveyor system [25]



(b) An AMR with a manipulator [46]

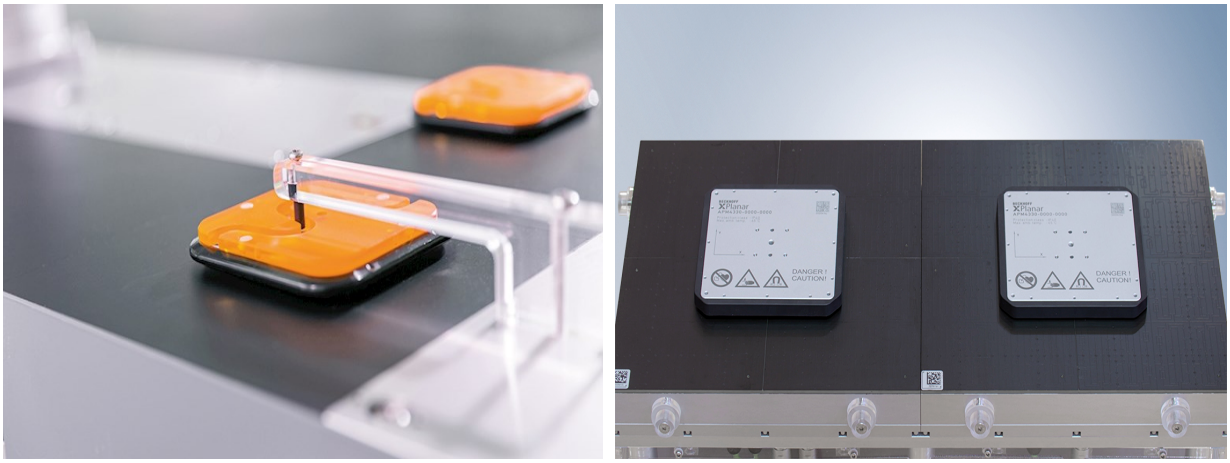
Figure 1.2: Different transportation solutions for flexible manufacturing systems.

through critical points of the production line (AGVs) or by using on-board sensors and a real-time planner (AMRs), both robotic systems provide new degrees of freedom for process optimization. While they have been found to be a valid solution in certain scenarios, the added freedom comes with the known technical difficulties that robotic applications introduce, such as real-time path planning for obstacle avoidance with simultaneous localization and mapping, docking stations for recharging and task allocation, and object placement and manipulation. All of them are equally critical, since the robotic system will have to traverse human-populated zones. Even then, another factor that prevents the general use of AMRs in FMS applications is that they remain contact-based transport solutions and are therefore inadequate for friction-sensitive or cleanroom processes.

MLPAs overcome both of these limitations: they provide omnidirectional, unconstrained motion within the workspace while operating entirely without mechanical contact. Moreover, the natural positioning flexibility of an MLPA facilitates integration with already installed manipulator systems, such as robotic arms with special end effectors, by enabling tilting during the manipulation task. This added operability is a crucial requirement for welding and painting procedures. Moreover, MLPAs eliminate the need for constant maintenance and environmental cleaning required to maintain cleanroom operations in pharmaceutical settings, expanding the range of tasks achievable within a single workcell. Recognizing the potential in MLPAs, highly regarded engineering brands have

developed their own magnetic levitated planar motors (MLPMs) systems, including highly capable commercial products like the ACOPOS 6D from B&R and the XPlanar by Beckhoff Automation Ltd. Figure 1.3 visualizes both small factor planar motors and their ability to achieve a theoretically infinite range of planar motion thanks to their modular setup.

Although levitation technology provides competitive advantages over traditional transportation methods, magnetic levitation remains relatively unknown in manufacturing due to high initial investment costs, a required steep learning curve to understand the inner working principles, and typically lower payload capacities per mover (under 5 kg). Additionally, current commercial Halbach movers have clear limitations, such as not being able to support large air gaps (typically restricted from 2 to 5 mm), forcing the system to only levitate with a maximum tilt angle of $\pm$ 2° on the mover roll and pitch axes, which limits the potential for cooperation between the magnetic device and an external manipulator for tasks where simultaneous motion could improve the performance, such as painting, visual inspection, welding and assembly. Given these motion restrictions, commercial small-factor movers are commonly used for simple top-side operations, making them more of a positioning-table device than a fully 6 DOF robotic agent.



(a) ACOPLOS 6D planar actuator from ABB-B&R. Image taken from the B&R website [7].

(b) XPlanar planar actuator from Beckhoff Automation. Image taken from Beckhoff website [3]

Figure 1.3: Commercially available small factor MLPA systems.

1.2.2 Cooperative Robotic Systems in Manufacturing

In precision manufacturing applications such as welding, surface treatment, and additive manufacturing, it is often necessary to design and implement distinct positioning devices to meet part tolerances and the hardware capabilities of external robotic actuators, such as robotic manipulators. As such, complex and repetitive operations often use tailored conventional conveyor lines designed solely to transport a product family or type, making it harder to implement FMS procedures and limiting optimization when the positional tool is treated as an isolated system. For example, Gueta et al. [24] demonstrated that, by combining a 6 DOF robot arm with a single-axis rotating table for multi-goal point-to-point tasks, it is possible to obtain faster completion times by enabling collaborative work between the positional device and the manipulator, showing that coordinating both systems reduces task completion time by nearly 18% compared to a single arm. Another often considered way to optimize the manufacturing procedure is to add extra manipulators and exploit the added DOFs. Zeng et al. [69] studied the kinematic capabilities of distributing a bilateral laser welding task across three coordinated manipulators on a shared translational truss, enabling simultaneous execution of symmetric weld seams on aerospace panels. While a valid way to optimize the process, it inherently adds the complexity and cost of multiple robotic arms sharing the same workspace.

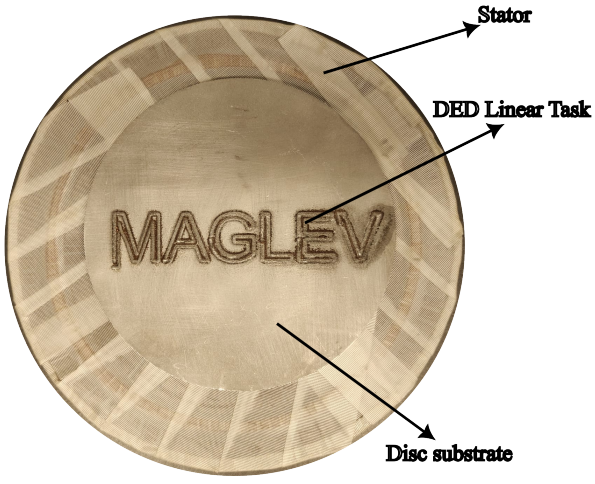


Figure 1.4: Maglev/MLPA application for direct energy deposition operations. Same device as the one used in [33]. The disc can only be levitated in the Z-axis.

As expected, MLPAs ability for natural 6-DOF control gives them an extra advan-

tage over conventional systems by enabling the mover to not only be an omnidirectional transporting device but also an integrated robotic actor during the entire manufacturing process, allowing the possibility for cooperation with external manipulators. This interaction between Maglev devices and robotic arms has already been explored, with a particular interest among researchers in medical contexts. Sun Tonxing [54] implemented a unified control framework for the interaction between a magnetically levitated particle and an electromagnet end effector, achieving real-time optimal positioning for force generation through coordinated arm joint trajectory and coil current actuation. Similarly, A. Khabbazian et al. [30] implemented a robot arm-magnetic levitation interaction for drug delivery through enclosed environments, with potential applications in non-intrusive and kidney disease treatments. In contrast to medical advancements in robotic cooperation, industrial manufacturing implementations of maglev devices have largely been explored as replacements to meet cleanroom requirements or for heavily constrained cooperation interactions that use a single mover axis. Kumar et al. [33] demonstrated the benefits of magnetically suspending a disc-shaped substrate during a direct energy deposition (DED) powder-feeding process, freeing the system from mechanical clamps, reducing completion time, and eliminating the surface marks that require post-processing treatment.

1.3 Objectives and Contributions

The literature reviewed in this chapter shows that MLPAs have strong potential as enabling technologies for flexible manufacturing because they combine contactless transport, high-precision positioning, and natural 6-DOF motion within a single platform. At the same time, although cooperative robotic systems have been widely studied using rotating tables, translational stages, and multiple manipulators, the integration of MLPAs with external robotic manipulators remains largely underexplored in industrial manufacturing environments. Existing maglev-manipulator cooperation studies are predominantly found in medical applications or in highly constrained industrial demonstrations where the levitated device is used in a limited manner. Consequently, a gap remains between the positioning and flexibility capabilities offered by MLPAs and their practical deployment as cooperative robotic actors in manufacturing tasks.

Therefore, this thesis addresses that gap by investigating the cooperative integration of the University of Waterloo Testbench MLPA system with an external 4-DOF robotic arm. The research is centered on a four-magnet DMM, selected for its suitability for tasks that require larger air gaps and tilting capability during manipulation. By improving the modeled range of actuation and controllability of the mover and by studying its interaction with a robotic manipulator, this work aims to advance the use of MLPAs from stand-alone positioning devices toward collaborative manufacturing platforms.

The MLPA environment studied in this thesis was developed and built as prior work by Dr. Xiaodong Zhang and Dr. Chanuphon Trakarnchaiyo; this includes the stator module and structure, power devices, and amplifier selection for actuation. The original design, assembly, and controller of the used DMM in this thesis were developed by Dr. Chanuphon Trakarnchaiyo. Both devices are detailed in Section 3.2.

The subsequent sections, Sections 1.3.1 and 1.3.2, list the defined objectives to achieve the proposed advancement towards MLPA and robotic manipulator cooperation for the flexible manufacturing industry, respectively.

1.3.1 Main Objectives

To achieve the proposed cooperative integration between the MLPA and robotic arm, the following objectives were selected:

1. Improve levitation on the four-disc magnet mover at the Testbench by implementing robustness in the modeling prediction and increasing the actuation range.
   (a) Implementation of a proper prediction-estimation state for the PID controller (Kalman Filter).
   (b) Decoupling the usage of a fixed-size LUT for a single model of independent coil activation by the development of a deep neural network strategy.
   (c) Experimentation and data collection for comparison between the original and improved methodology.
2. Integration of the MLPA platform with a 4-DOF robotic manipulator and evaluation of their coordinated operation in a manufacturing-oriented setting.
   (a) Integration of a discrete workspace filtering for possible mover-arm interaction states.
   (b) Analytical modeling of the forward and inverse kinematics of the manipulator for computational efficiency and repeatability regarding numerical methodologies.
   (c) Development of the communication and sequence pipeline for robot manipulator - MLPA continuous contact.
   (d) Development of the path planning algorithm for cooperative interaction under discretized continuous path tasks.

1.3.2 Contributions

In direct fulfillment of the stated objectives, this thesis presents the following major contributions to the fields of magnetic levitation and cooperative robotics:

- **Enhanced MLPA Control and Actuation Modeling:** Developed a robust state estimation and force generation framework for the DMM. This includes the successful implementation of a discrete Kalman Filter to eliminate noise-induced jitter associated with raw state estimation, and the deployment of a Dual Neural Network architecture to calculate forces for accurate coil current control. This NN-based approach decoupled the system from the restrictive, localized boundaries of a fixed-size look-up table, effectively expanding the reliable actuation range across the entire workspace without failing through the Curse of Dimensionality of the previous rigid structure.

- **System Integration and Modeling for Mover-Arm Cooperation:** Formulated a complete interaction architecture bridging the MLPA testbench with a 4-DOF robotic manipulator. This encompasses the development of closed-form forward and inverse kinematic models for rapid computational solving, the generation of a discrete workspace filter to map geometrically viable interaction states, and the establishment of a real-time master-slave communication pipeline to synchronize physical execution. Additionally, this includes the overall programming and development of the entire cooperative workflow, including the interaction objects' parameters and virtualization.

- **Kinematic-Aware Cooperative Path Planning Algorithm:** Designed and experimentally validated a novel, time-optimized cooperative path planner capable of producing feasible discrete chained configurations for both the MLPA and manipulator systems to accomplish discretized trajectories. By enabling the MLPA mover to dynamically reorient the workpiece in tandem with the robotic arm, this algorithm successfully facilitates complex manufacturing operations—such as continuous dynamic arcs—that are geometrically unfeasible for a statically anchored manipulator with insufficient required DOFs.

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2: Literature Review presents a comprehensive examination of the current state of magnetic levitation technologies in industrial settings. It surveys existing MLPA control and modeling methodologies and reviews prior research regarding cooperative robotic manipulation in manufacturing environments.

Chapter 3: System Setup details the hardware architecture and system specifications. It describes the physical components of the MLPA testbench, the selected four-disc magnet mover (DMM), the Vicon motion tracking system, and the kinematics and reference frames of the 4-DOF robotic manipulator used for the cooperative interaction.

Chapter 4: Implementation and Methodology Improvements outlines the theoretical modeling and practical implementation of the proposed systems. The chapter is divided into two main sections: first, the development of the enhanced MLPA control framework (encompassing the Kalman Filter, active coil mapping interpolation, and the Dual Neural Network for the extended workspace), and second, the detailed design of the cooperative path planning algorithm (including inverse kinematics, workspace filtering, and the time-optimized graph search mechanism).

Chapter 5: Experimental Results and Validation presents the empirical results of the implemented strategies. It first evaluates the standalone path tracking performance of the upgraded MLPA model against the original baseline. Subsequently, it details the execution of the cooperative path planner on three distinct manufacturing trajectories (a circle, a straight line, and a complex arc), comparing the time and accuracy metrics of the cooperative mover-arm interaction against a statically anchored arm-only baseline.

Chapter 6: Discussion and Conclusions concludes the thesis. It summarizes the core contributions and the logical sequence of development, discusses how the cooperative framework expands the flexibility of kinematically limited manufacturing lines, and proposes future research directions such as energy-aware planning and force-sensing integration.

Chapter 2

Literature review

2.1 Magnetic Levitated Planar Actuators

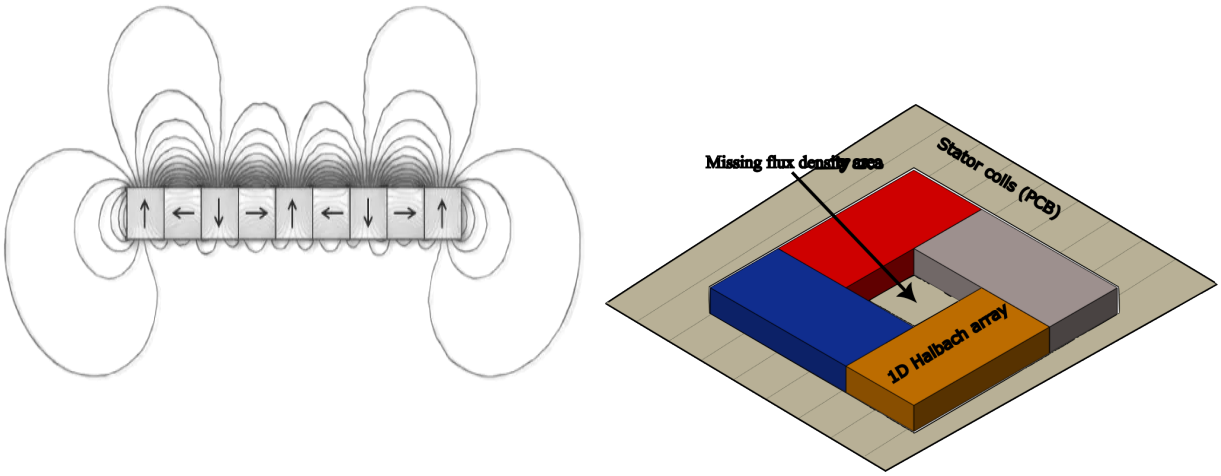
2.1.1 Developed MLPA systems review

As the core element shared by all dynamically stable MLPA systems, the stator is the element related to the controllable electromagnetic field generator. Following the references provided in the introduction and the known relationship between electric and magnetic fields, stators on an MLPA device can be stationary DC or AC-driven coils, moving permanent magnets, superconductors, or a combination of the mentioned setups. While having its own drawbacks, fixed coil stators are preferred for simplicity and clear element division. Having an arrangement of electromagnets designed by modules enables the flexibility for future expansion, thanks to two clear advantages: fast access to wiring extension and standalone cooling systems. In addition, having the mover as a purely passive carrier simplifies the mover design and required considerations for multi-mover setups. It is for these reasons that the moving-magnet, stationary-coil topology has become dominant in large-stroke MLPA research, and all systems reviewed in this section adopt this configuration.

Among stationary-coil stators, the geometry of the individual coil is one of the primary design variables. One of the first designs introduced was in [31]. In this setup, the stator is composed of four horizontal wound coils, where each one is designed with a specific winding direction to decouple the magnets of the platen (mover). While simplifying the Maglev implementation, the setup fixed the planar actuator to the stator module, limiting the stroke movement to the size of the single stator. In an improvement over the previous project, the design in [36] addresses this by replacing dedicated coil stators with multiple

layers of planar PCB coils, with each layer designed to control a dedicated mover axis. With the motivation of having multiple movers sharing the same stator space, [72] changed the single line of PCB coils for standard rounded square ones and applied a commutation law to resolve the cross-coupling of the different magnet arrangements on the mover.

All three stator systems reviewed above share a common mover topology: four one-dimensional (1-D) Halbach permanent magnet arrays arranged with different alignments. A 1-D Halbach array is a linear sequence of permanent magnets whose magnetization direction rotates by 90° between adjacent elements, producing a field that is concentrated on one side of the array while largely canceling on the other (Figure 2.1, a).



(a) 1-D Halbach array. Adapted from [63]

(b) Commonly found four 1-D Halbach array configuration. Recreated from [36]

Figure 2.1: Arrangement of 1-D Halbach arrays for mover design.

While praised for its straightforward modeling and design, 1-D Halbach array movers result in a loss of power efficiency given a non-usable space between the arrangements. An alternative that solves the optimization issue is the 2-D Halbach array, where the magnetization pattern repeats spatially in both in-plane directions. The 2-D arrangement concentrates flux beneath the entire mover footprint rather than in four separate strips, yielding a higher force density for the same magnet volume and enabling a single contiguous magnet array to interact with the full coil field below it [28]. As found in [65], the 2-D Halbach array is capable of carrying a higher payload, with an increase in modeling

complexity, as the force and torque contributions from the stator coils must now account for a two-dimensional field distribution, including edge effects along both in-plane axes. The modular maglev system presented in [67] adopts a 2-D Halbach mover above a square coil stator and resolves the wrench calculation problem through a data-driven method, where a Force Decay Ratio and Effective Torque Arm smoothly divide the far and near field actuation zones over the mover, reducing the required parameters' table from over 964,000 data points per levitation height to only 791 and achieving a computation time of 2.1 μ s per active coil. With the wrench calculation solution, the researchers in [66] improved the dynamical response of the system by combining a Kalman filter with a frequency filter for delay-free state prediction. The proposed predictor-based MPC reduced the ramp tracking RMSE by 87% compared to a PID controller and by 69% compared to standard MPC, while increasing the maximum stable mover speed from 300 mm/s under PID to 800 mm/s.

While the 2-D Halbach arrangement offers clear force density and yaw advantages over the four-array topology, its reliance on robust framing to contain the strong internal forces between adjacent magnet elements naturally limits the achievable air gap and mover tilt operating ranges. An alternative mover topology that directly addresses these structural constraints is the usage of a disc magnet mover (DMM), with its basic form composed of a single cylindrical permanent magnet that replaces the entire array. Without the strong internal attraction forces characteristic of Halbach arrangements, the DMM frame can be made significantly lighter. However, the axially symmetric geometry of a disc magnet eliminates the spatially periodic magnetic field that underlies harmonic analytical wrench models of Halbach movers, making real-time force and torque computation considerably more challenging.

To overcome the heavy computational requirement of modeling the DMM wrench, researchers have proposed both data-driven and analytical alternatives. For instance, a residual-based deep neural network (DNN) wrench model trained on datasets of 8 million randomly sampled DMM poses achieved a per-coil computation time of 4.1 µs on a programmable logic controller [59]. This performance is comparable to analytical solutions on Halbach systems, yielding translation and rotation control resolutions of 10 µm and 10 mdeg, respectively. Alternatively, an analytical magnet-coil role-switching model mathematically recasts the square coils as stacked permanent magnets and the disc magnet as a thin-walled solenoid. By leveraging this model and the lighter DMM frame, levitation was achieved at impressive air gaps up to 65 mm and roll/pitch rotations of up to 45°. Furthermore, this role-switching approach computes the full wrench matrix between a single disc magnet and nine coils in 96.5 µs using parallel computing, demonstrating experimental translation and rotation resolutions of 10 µm and 0.02° [61].

Given the advantages of extended air gap and tilting range, in this thesis a DMM

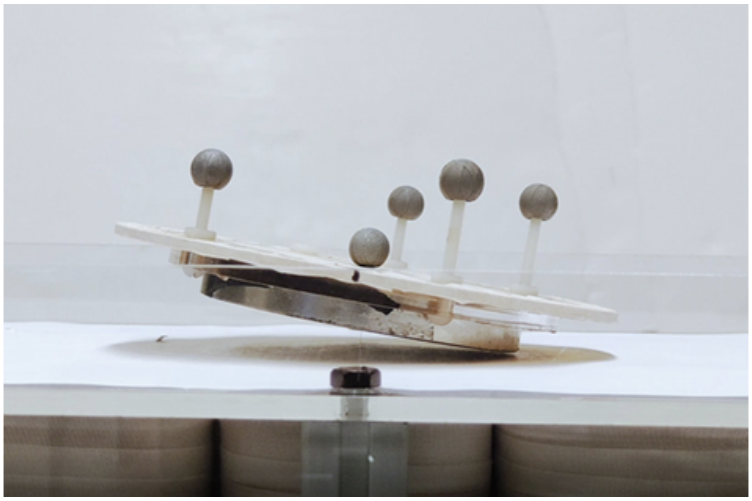


Figure 2.2: Photo of a single DMM. Note that given the disc geometry, the mover does not have yaw rotation control. Image adapted from [38].

actuator is studied. The DMM details are provided by Dr. Chanuphon in [55], designed to be an arrangement of four disc permanent magnets. As a contribution to the overall system, and to enable high-precision wrench calculation on the new range of levitation required for mover-robot arm interaction, a DNN approach is embraced for the system levitation. The specifics of the implementation are studied in the modeling Section 4.2.

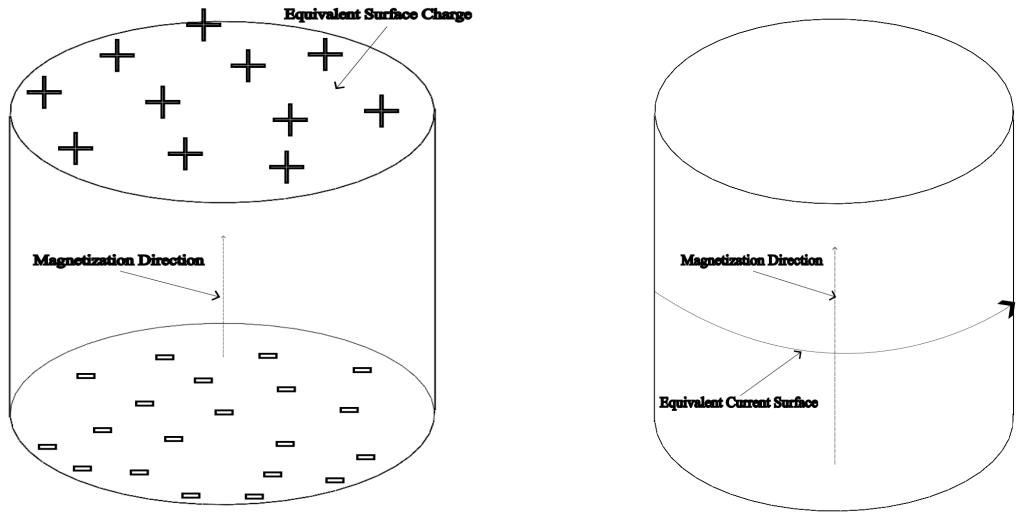
2.1.2 MLPA modeling methodologies

Controlling any MLPA requires the modeling and calculation of the interaction force applied on the mover arrangement. This wrench is obtained through a chained process with two steps: first, computing the magnetic field and magnetic flux $\mathbf{B}$ produced by one of the elements (mover or stator) at any point in the working volume. Second, mapping the magnetic field to the force generated, which depends on the application and elements used. For the implemented maglev system, the mover is the magnetic field generator and, by the application of a current-driven device (the stators), the wrench is calculated by the Lorentz force law.

For the magnetic flux density computation, there are two well-known analytical approaches in the literature for permanent magnets with uniform magnetization. The first is the **current model**, in which the magnetization $\mathbf{M}$ of the magnet is replaced by an equivalent magnetic volume current density, obtained by applying the curl operator $(\nabla\times)$

as $\mathbf{J}_m = \nabla \times \mathbf{M}$. This equivalent current distribution is then used as a source in the electromagnetic general equations, from which the magnetic flux density $\mathbf{B}$ is computed [19].

The second approach is the **charge model**, which instead represents the magnet as two parallel surfaces carrying fictitious magnetic charge densities $\sigma_m = \pm B_r$ on the top and bottom faces [49]. This is formally analogous to the surface charge distribution in electrostatics and leads to a field expression of the same integral form as the current model. While providing a physically intuitive framework and analytical equations for permanent magnets, the charge and current models share one critical limitation in their practical applications: both methods require the solution of elliptic integrals, which can be computationally intensive when it is required to calculate the magnetic field for a large number of points in the workspace.



(a) The surface charge method. (b) The equivalent current model.

Figure 2.3: Analytical approaches for magnetic field calculation of an electromagnet.

As an alternative to analytically calculating the magnetic field on cuboidal magnets, Bancel in 1999 developed the magnetic node method [1]. Building on the charge model formulation, Bancel demonstrated that the exact magnetic field of any uniformly rectangular permanent magnet can be decomposed into contributions from eight nodes *nodes* placed at the corners of the magnet, each carrying a charge $\varepsilon_k = \pm B_r$ with alternating signs as illustrated in Figure 2.4. The magnetic flux density contributed by node k at position

(x_k, y_k, z_k) to an observation point $P(x, y, z)$ is then found as:

$$B_{xk} = \frac{\varepsilon_k}{4\pi} \ln\left(-y_r + \sqrt{x_r^2 + y_r^2 + z_r^2}\right), \tag{2.1}$$

$$B_{yk} = \frac{\varepsilon_k}{4\pi} \ln\left(-x_r + \sqrt{x_r^2 + y_r^2 + z_r^2}\right), \tag{2.2}$$

$$B_{zk} = \frac{\varepsilon_k}{4\pi} \arctan\left(\frac{x_r\, y_r}{z_r \sqrt{x_r^2 + y_r^2 + z_r^2}}\right), \tag{2.3}$$

where $x_r = x - x_k$, $y_r = y - y_k$, and $z_r = z - z_k$. The total flux density at P from all N nodes in the system is the direct superposition:

$$\mathbf{B}(P) = \sum_{k=1}^{N} \begin{pmatrix} B_{xk} \\ B_{yk} \\ B_{zk} \end{pmatrix}. \tag{2.4}$$

Equations (2.1)–(2.3) involve only logarithms and arctangents, making them significantly cheaper to evaluate than the elliptic integrals required by the current and charge models. Furthermore, since each magnet adds exactly eight nodes, when the faces of the magnet are shared, the assigned nodes cancel each other, effectively reducing the linear growth of required calculations for cuboidal arrangements, such as Halbach movers. In the case of disc magnets, a discretization is required to apply the previous magnetic nodes equations. By discretizing the top and bottom faces of the magnet with a staircase pattern that follows the circular geometry, the same superposition principle holds. While now an approximation, the magnetic nodes method for disc magnets still provides a significant computational advantage with a minimal loss of accuracy when applied with a sufficiently fine number of nodes [55]. Figure 2.5 illustrates the staircase discretization of a disc magnet following the magnetic node model.

With the magnetic field established, the forces and torques on the mover are computed using the Lorentz force law. In this MLPA device, the disc magnet mover is suspended above an array of electromagnetic coils driven by DC current, where the current flowing in each coil is immersed in the field $\mathbf{B}$ generated by the permanent magnets, and the resulting Lorentz force on the coil, calculated by applying the cross-product operation $\times$ is used to find the reaction wrench on the mover following Newton's third law. For a three-dimensional system with current density $\mathbf{J}$ distributed over the coil volume V_{coil}, the

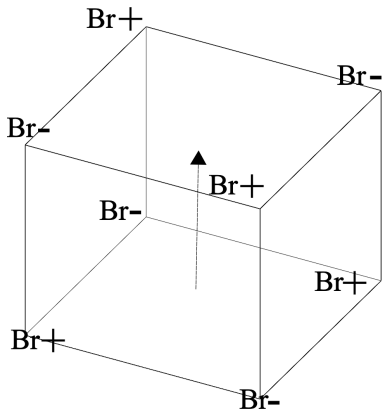
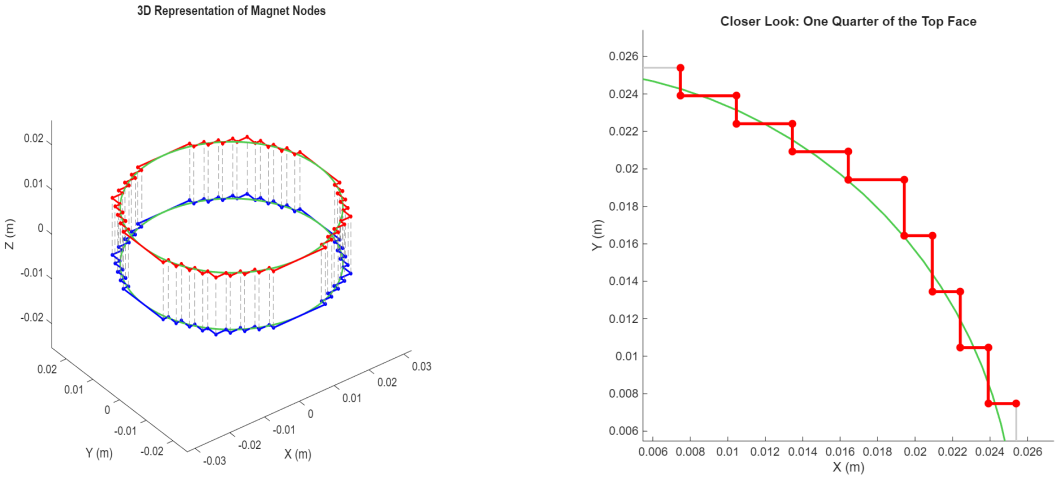


Figure 2.4: Magnetic node representation of a cuboidal permanent magnet, with one node at each corner carrying charge $\varepsilon_k = \pm B_r$.



(a) magnodes methodology for a disc magnet.

(b) Zoom in view on the discretization of the disc with magnodes.

Figure 2.5: Application of the magnodes method on a disc magnet. Note that, since the circle is being discretized, the resulting magnetic field is an approximation, in contrast to a cube magnet.

force $\mathbf{F}$ and torque $\mathbf{T}$ on the mover are given by:

$$\mathbf{F} = -\int_{V_{\text{coil}}} \mathbf{J} \times \mathbf{B}\, \mathrm{d}V, \tag{2.5}$$

$$\mathbf{T} = -\int_{V_{\text{coil}}} \mathbf{r} \times (\mathbf{J} \times \mathbf{B})\, \mathrm{d}V, \tag{2.6}$$

where $\mathbf{r}$ is the moment arm from the mover's center of mass to each integration element, and $\mathbf{B}$ is evaluated using Equation (2.4). Since the coil cross-section can be approximated as uniform, the current density magnitude $|\mathbf{J}| = I/A_{\perp}$ is constant throughout the conductor, simplifying the integration process by using the cross-section area $A_{\perp}$ instead of a turn-by-turn integration, also enabling the implementation of Gaussian quadrature [71] for faster calculations.

For a multiple magnet mover, the resultant arrangement force ($\mathbf{F}_{\mathbf{mover}}$) is the linear superposition of the individual magnetic forces. To calculate the dynamic torque generated by the lever arms of the magnets, the cross product $\mathbf{r}_i \times \mathbf{F}_i$ is mapped into a linear matrix multiplication using the skew-symmetric matrix representation $[\mathbf{r}_i]_{\times}$ of the position vector. The overall resultant torque on the mover is derived by summing the local magnetic torques with the dynamic lever-arm torques as

$$\mathbf{T}_{\text{mover}} = \sum_{i=1}^{N} \left(\mathbf{T}_i + [\mathbf{r}_i]_{\times} \mathbf{F}_i\right), \tag{2.7}$$

where $\mathbf{T}_i$ and $\mathbf{F}_i$ are the torques and forces of the i-th magnet, respectively.

The complete system state for a multi-magnet arrangement and a single coil can thus be compactly described by the resultant 6-DOF wrench vector $\mathbf{W}_{\text{mover}} \in \mathbb{R}^6$

$$\mathbf{W}_{\text{mover}} = \begin{bmatrix} \mathbf{F}_{\text{mover}} \\ \mathbf{T}_{\text{mover}} \end{bmatrix}. \tag{2.8}$$

The combined wrench from all active coils is assembled into a **wrench matrix** $\mathbf{K}$, which encodes the current-to-wrench relationship from the full system in the form of $\mathbf{K} = [\mathbf{W}_{\text{mover},1} ... \mathbf{W}_{\text{mover},N}]$. For a single mover magnet, the relationship is:

$$\mathbf{w} = \mathbf{K}\, \mathbf{i}, \tag{2.9}$$

where $\mathbf{w} = [F_x, F_y, F_z, T_x, T_y, T_z]^{\top}$ is the control effort required to obtain a desired

motion on the mover and $\mathbf{i} \in \mathbb{R}^N$ is the vector of currents for the N active coils. To independently control the 6 degrees of freedom (DOF) of the mover, the stator requires a minimum of 6 active coils. In practice, to distribute the load and ensure stable levitation, more than 6 coils are activated ($N > 6$), rendering the system underdetermined and making $\mathbf{K}$ a non-square matrix. To resolve this actuator redundancy and decouple the coil currents while minimizing the overall power consumption (by minimizing the L_2-norm of the current vector), the Moore-Penrose pseudoinverse $\mathbf{K}^+$ is commonly employed. Following [45]:

$$\mathbf{K}^+ = \mathbf{K}^\top (\mathbf{K}\mathbf{K}^\top)^{-1}, \tag{2.10}$$

which yields the control current vector via $\mathbf{i} = \mathbf{K}^+\mathbf{w}$. Since the pseudoinverse is unable to account for the physical limitation of the actuators, the wrench null space projection [17] is used as a refinement step to distribute the load of oversaturated coils. The orthogonal projection of the null space ($\mathbb{N}$) of $\mathbf{K}$ is given by:

$$\mathbb{N} = I - \mathbf{K}^\dagger \mathbf{K}. \tag{2.11}$$

Denoting the current of saturated coils to be distributed as δ_{sat} and the isolated projection of saturated coils $\mathbb{M} = S\mathbb{N}S + I - S$, where S is a diagonal matrix with ones for a saturated coil and zero for a non-saturated coil, the required combination in the null space α_{null} is found using

$$\alpha_{null} = \mathbb{M}^{-1}\delta_{sat}. \tag{2.12}$$

The final current vector is given as $\mathbf{i_{new}} = \mathbf{i} + \mathbb{N}\alpha_{null}$. The process is iteratively done for each new saturated coil that could arise, generally being limited to a maximum number of iterations for computational time and system feasibility, where it is often set to end once there are enough saturated coils to surpass the system power limitations.

2.2 Robotic manipulators

Serial robotic manipulators are open kinematic chains formed by a sequence of rigid links connected by actuated joints, most commonly revolute, that together deliver a controllable pose to an end effector attached at the last link of the chain [13]. Their widespread adoption in industrial automation stems from a combination of large reachable workspaces, mechanical simplicity, flexible applications by changing the end effector, and a mature body of kinematic theory that enables accurate and fast modeling and planning solutions. The following subsections review the principal arm architectures found on factory floors, the kinematic modeling frameworks used to describe them, and the cooperative strategies

through which they are paired with additional manipulators and positional devices to extend their capability.

2.2.1 Common robotic manipulators in the industry

Industrial serial manipulators are classified primarily by the number of degrees of freedom they provide and by the geometric arrangement of their joints, both of which determine the set of tasks they can execute without supplementary equipment. The minimum number of DOFs required to position an end effector at an arbitrary point with any required orientation in three-dimensional space is six [37]. Accordingly, the **6-DOF articulated arm** is the main driver for manufacturing processes requiring welding, painting, quality control, palletizing, and assembly on conventional lines. Figure 2.6 shows one of the first 6-DOF manipulators, developed by the Unimation company with their PUMA line and composed of six revolute joints, with a 3-DOF wrist whose three intersecting joint axes form a kinematically spherical wrist for the last link.



Figure 2.6: Photo of a PUMA 560. Image adapted from [50].

When the task does not require all 6-DOF, lower-DOF arms are preferred for their reduced cost, lighter structure, simpler control, and specialized design. The **SCARA** (Selective Compliance Assembly Robot Arm) is one of the most relevant 4-DOF industrial arms. It is composed of three in-plane revolute joints and a prismatic joint that provides

vertical stroke movement, as seen in Figure 2.7, making it a viable arm for planar operations such as pick-and-place, PCB assembly, and CNC machine tending. A common guideline in manipulator design is to select the minimum number of degrees of freedom required by the task, since each additional joint typically increases hardware cost, weight, and control complexity without necessarily providing proportional benefits. As a consequence, many installed industrial arms have four or five DOF, since they were installed to optimize a specific task, heavily penalizing a production line's flexibility for new tasks.



Figure 2.7: Photo of an Epson SCARA T6.

In contrast to manipulators that have only the minimum degrees of freedom required by a task, engineers and researchers often benefit from additional DOFs by exploiting redundancy for task optimization. Instead of a single feasible solution, a redundant manipulator admits an infinite set of joint configurations for the same end-effector pose, which can be used to study secondary objectives such as reducing task operation time, minimizing energy consumption, limiting unnecessary joint motion, or lowering joint loads. In Chen et al. [11], a 7-DOF anthropomorphic arm is commanded to follow linear, circular, and figure-eight trajectories. Using a particle swarm optimization method to optimize the extra DOF, the overall motion time was reduced by between 16% and 30% compared to that obtained with no redundancy optimization.

2.2.2 Robotic manipulators modeling strategies

Forward Kinematics and the Denavit-Hartenberg Convention

Forward kinematics (**FK**) addresses a fundamental concept in robotics: determining the exact position and orientation of a robot's tip (end-effector) in 3D space, relative to the robot arm base, given the current angle or displacement of every joint in its arm. Mathematically, this involves mapping a set of joint variables $\mathbf{q}$ to the final position and orientation of the end-effector, often expressed as a transformation matrix [37].

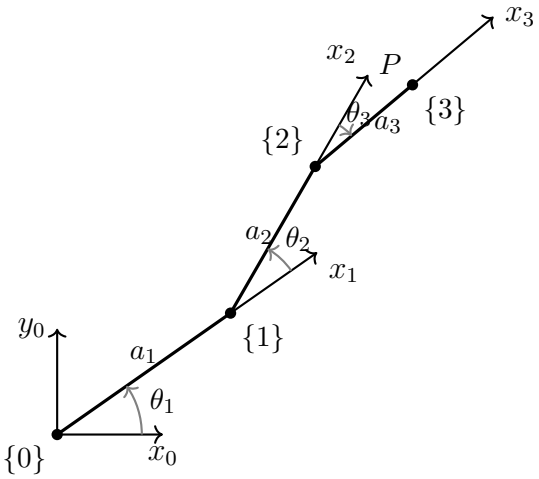


Figure 2.8: A 3-Link planar manipulator, composed of three revolute joints (3R). The end-effector pose (**P**) is represented by a homogeneous transformation matrix.

To have a general definition for any n-DOF manipulator, researchers commonly use the **Denavit-Hartenberg (D-H) convention**, first introduced in 1955 [14]. Instead of calculating complex 3D geometry all at once, the D-H convention breaks the robot down step-by-step. It assigns a coordinate frame to each link and describes how to move from one joint to the next using exactly four geometric parameters: the joint angle θ_i, the link offset d_i, the link length a_i, and the link twist α_i. Moving from one joint frame to the consecutive one is simply a product of four basic rotations and translations:

$$^{i-1}T_i = \mathrm{Rot}_{z,\theta_i}\,\mathrm{Trans}_{z,d_i}\,\mathrm{Trans}_{x,a_i}\,\mathrm{Rot}_{x,\alpha_i}, \tag{2.13}$$

By multiplying these step-by-step transformations together from the base of the robot to its tip, the complete FK is obtained: $T(\mathbf{q}) = {}^0T_1 \cdot {}^1T_2 \cdots {}^{n-1}T_n$. Figure 2.8 shows the

diagram of a three-link planar robot arm, where the corresponding D-H parameter table is given by:

Table 2.1: Denavit-Hartenberg D-H parameters for the 3R planar manipulator.

Link i	θ_i	d_i	a_i	α_i
1	θ_1	0	a_1	0
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0

Using the D-H table and substituting its parameters into Equation 2.13, we can calculate the individual transformation matrix for each link. Because the arm operates in a 2D plane, there is no twist ($\alpha_i = 0$) and no offset ($d_i = 0$). Thus, the product of the four operations simplifies to $\text{Rot}_{z,\theta_i}\,\text{Trans}_{x,a_i}$, yielding a general transformation for link i:

$$
{}^{i-1}T_i = \begin{bmatrix} c_i & -s_i & 0 & 0 \\ s_i & c_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

where c_i and s_i are shorthand notations for $\cos(\theta_i)$ and $\sin(\theta_i)$, respectively. Applying this to each row of Table 2.1 gives the three individual transformations:

$$
{}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

Finally, by sequentially chaining these matrices, the complete end-effector transformation relative to the base frame is found:

$$
{}^0T_3 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ s_{123} & c_{123} & 0 & a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.14}
$$

where $c_{i\ldots j}$ and $s_{i\ldots j}$ denote $\cos(\theta_i + \cdots + \theta_j)$ and $\sin(\theta_i + \cdots + \theta_j)$.

The enduring popularity of the D-H convention lies in this simplicity. It elegantly condenses the complex structure of any arbitrary robotic arm into a compact table, where each row translates directly into a standard 4×4 matrix. Given its extended use in robotic research, it is used as the basis for the FK modeling of the manipulator studied in this thesis, with the D-H table shown in Table 3.3.

Inverse Kinematics: Analytical and Numerical Solutions

While Forward Kinematics asks where the robot's tip is based on its joint angles, the **Inverse Kinematics** (IK) problem asks the exact opposite: knowing a specific position and orientation where we want the end-effector to be, what is the joint configuration $\mathbf{q}$ required to obtain the requested pose? This is inherently more complex than the FK problem, because a single target pose might be reachable through multiple different arm configurations, or it might be completely unfeasible. A common approach to find the IK mapping is by finding these angles numerically, with classical iterative approaches such as the Newton-Raphson method [9] appreciated for their ease and flexibility of implementation, with high levels of accuracy and speed compared to other approaches such as neural network-based approaches. Nonetheless, Jacobian-based solvers suffer from local minima (as a product of being gradient based) and also tend to be computationally heavy as the DOFs of the manipulator increase, given the increased burden of finding the inverse of the Jacobian, making them better suited for environments with low DOFs [35].

Following the 3-link planar arm introduced in Figure 2.8. The end-effector pose can be described by its position (x, y) and planar orientation $\phi = \theta_1 + \theta_2 + \theta_3$. From the derived FK transformation matrix 2.14, the position and orientation are given by:

$$\mathbf{x}(\mathbf{q}) = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix}.$$

The analytical Jacobian matrix $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{3\times3}$, which maps the change of the end-effector state vector with the joint configuration, is found by computing the partial derivatives of $\mathbf{x}(\mathbf{q})$ with respect to each joint angle θ_i:

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \\ \frac{\partial \phi}{\partial \theta_1} & \frac{\partial \phi}{\partial \theta_2} & \frac{\partial \phi}{\partial \theta_3} \end{bmatrix} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix}.$$

The iterative solver begins with an initial guess for the joint configuration, $\mathbf{q}_0$. At each step k, the Cartesian error between the target pose $\mathbf{x}_{\text{target}}$ and the current pose $\mathbf{x}(\mathbf{q}_k)$ is computed as $\Delta\mathbf{x} = \mathbf{x}_{\text{target}} - \mathbf{x}(\mathbf{q}_k)$. The necessary change in joint angles $\Delta\mathbf{q}$ is estimated using the first order step $\Delta\mathbf{x} \approx \mathbf{J}(\mathbf{q}_k)\Delta\mathbf{q}$.

Because the 3-link arm has exactly 3 degrees of freedom to control a 3-dimensional task space, its Jacobian is a square matrix. In cases where extra DOFs are available to complete a task, pseudoinverse methods such as the Moore-Penrose are used, often involving damping terms to circumvent singularities [64]. As long as the arm is not in a singular configuration (where the determinant of the Jacobian is zero and the matrix loses its full rank), the necessary joint updates can be solved using the standard matrix inverse $\mathbf{J}^{-1}$:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}^{-1}(\mathbf{q}_k)\Delta\mathbf{x},$$

until the error magnitude $\|\Delta\mathbf{x}\|$ falls below a predefined tolerance threshold.

While the iterative Jacobian method converges to a single configuration heavily dictated by the initial guess $\mathbf{q}_0$, it does not inherently reveal multiple valid poses (such as elbow-up or elbow-down). In contrast, if the robotic arm is designed with specific geometric features, such as having parallel or intersecting joints, we can use **closed-form geometric solutions**. This method uses direct trigonometric formulas to calculate the exact joint angles instantly, bypassing the need for numerical iterations. Due to their direct mathematical formulation, analytical solutions often produce almost zero computational cost in comparison to numerical IK methods. Geometrical solutions are highly preferred when a system needs to quickly evaluate thousands of potential movements during path planning, but they are heavily dependent on the specific manipulator configuration. In the same family as geometrical solutions, algebraic solvers that reformat the FK transformation matrix using trigonometric identities, algebraic eliminations, and mathematical substitutions to obtain closed-form equations are also extremely efficient given their polynomial form [15]. However, algebraic solvers are less common in practice for redundant manipulators (over 6 DOFs), since they typically require locking one of the extra DOFs to obtain the IK equations or for manipulators where the dependent joint configurations are too complex to produce relationships between them.

Consider the previous 3-link planar arm targeting a desired pose (x, y, ϕ). Since the end-effector orientation ϕ and the length of the final link a_3 are known, the position of the second joint (often referred to as the wrist center, x_w, y_w) can be decoupled from the orientation:

$$x_w = x - a_3 \cos(\phi), \quad y_w = y - a_3 \sin(\phi). \tag{2.15}$$

This effectively reduces the system to a 2-link inverse kinematics problem. The first two links (a_1 and a_2) and the distance from the base to the wrist center ($r = \sqrt{x_w^2 + y_w^2}$) form a triangle. Applying the Law of Cosines to this triangle yields the second joint angle θ_2:

$$\cos(\theta_2) = \frac{x_w^2 + y_w^2 - a_1^2 - a_2^2}{2a_1 a_2}. \tag{2.16}$$

The nature of the cosine function dictates that there are generally two valid solutions for θ_2 (one positive and one negative), corresponding respectively to the **elbow-down** and **elbow-up** configurations. Once θ_2 is chosen, the base angle θ_1 is found by subtracting the inner triangle angle from the global angle of the wrist:

$$\theta_1 = \text{atan2}(y_w, x_w) - \text{atan2}(a_2 \sin(\theta_2), a_1 + a_2 \cos(\theta_2)). \tag{2.17}$$

Finally, since the total orientation is the sum of all joint angles ($\phi = \theta_1 + \theta_2 + \theta_3$), the final joint angle is simply calculated as:

$$\theta_3 = \phi - \theta_1 - \theta_2. \tag{2.18}$$

This method provides exact, instantaneous solutions while naturally exposing multiple valid manipulator configurations, completely avoiding the singularity-induced mathematical failures inherent to numerical solvers. Given the advantages of using analytical solutions and the geometrical configuration of the studied 4-DOF arm, in this thesis a geometrical IK is modeled with a procedure very similar to the one provided in this section, since the 4-DOF arm has three revolute joints in the same parallel plane. For the combined system of the DMM and the arm, a Jacobian-based method with a damping term is applied for the redundancy local solver (see the modeling Section 4.4.4).

Workspace Representation via Randomized Sampling

The **reachable workspace** of a manipulator is the set of all end-effector positions attainable by at least one joint configuration satisfying the joint limits [37]. While the reachable

workspace is generally related to only the possible coordinates of the end-effector without regard to the orientation, manufacturing tasks that require strict angle control must also consider the feasible end-effector alignments. Analytically characterizing this set is tractable only for very simple geometries, such as 3R planar arms. For general serial chains, it is common to find numerical approaches using a developed FK solution to obtain a discrete representation of the workspace. One of the most common approaches is the usage of random sampling techniques, with implementations such as Monte-Carlo being typically applied. In this method, the joint configuration $\mathbf{q}$ is drawn from a uniform distribution (which is often restricted to the known joint limits), the FK is then evaluated for the resulting configuration, and the end-effector pose is recorded [23]. Repeating this for a large sample N_{test} yields a dense point cloud approximating the reachable workspace without any assumption about the manipulator geometry. The drawback of workspace discretization is that, due to the use of a uniform distribution, the resulting poses will tend to accumulate in a zone of the sampled range. Even then, postprocessing filters can be used to ensure a minimal threshold of samples on determined configuration space widths. Figure 2.9 represents the workspace of a planar 3R manipulator.

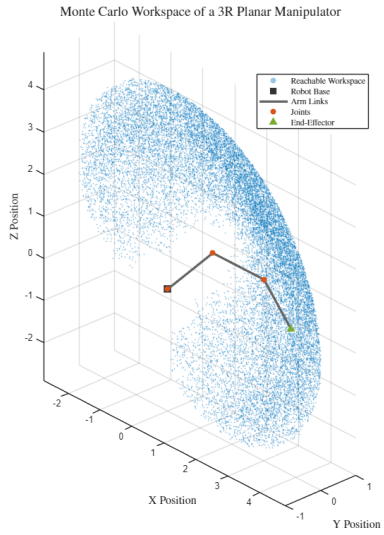


Figure 2.9: Reachable workspace (positions) of the 3R planar arm. Apart from the configuration of the joints, the workspace is defined by the arm's link length and joint limits.

Chapter 3

System Setup

This chapter details the physical and communication setup of the robotic system used for the implementation. It provides an overview of the Testbench, describes the key hardware components including the mover and the robot arm, and explains the control architecture connecting these systems.

3.1 Overall Testbench Setup

The Testbench consists of a single module with 64 DC-driven coils arranged in an 8×8 grid ($0.61 \times 0.61\,\mathrm{m}^2$ total area with a coil pitch $\Delta_{\mathrm{coils}} = 0.0762\,\mathrm{m}$), as shown in Fig. 3.1. Each integrated coreless coil features a rounded square profile with an inner side of 32 mm (4 mm inner radius) and an outer side of 75 mm (25.5 mm outer radius), as illustrated in Fig. 3.2. With a coil height of 82 mm, the windings consist of 61 layers of 16 turns of AWG 16 copper wire, totaling 976 turns.

The Testbench is energized through two power supplies connected to a relay panel that distributes energy to 32 amplifiers. Each amplifier individually manages two stator coils. The stator coils found in the Testbench were designed for a total absolute current of 9 A, as reported in [55]. To prevent multiple coils from saturating simultaneously during levitation and overloading the power distribution system, Curtis [53] developed a current recalculation algorithm to employ the available capacity of non-saturated coils for levitation, effectively distributing the excess current from coils exceeding 9 A, enhancing the robustness of the system in the form of better stability during complex motions.

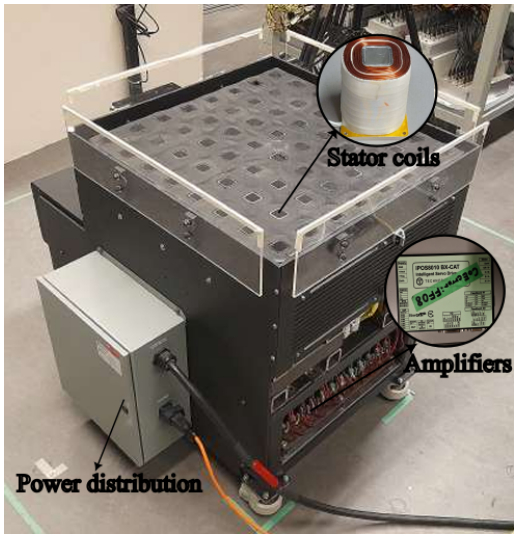


Figure 3.1: Testbench stator arrangement. The coils and amplifiers are magnified for clarity.

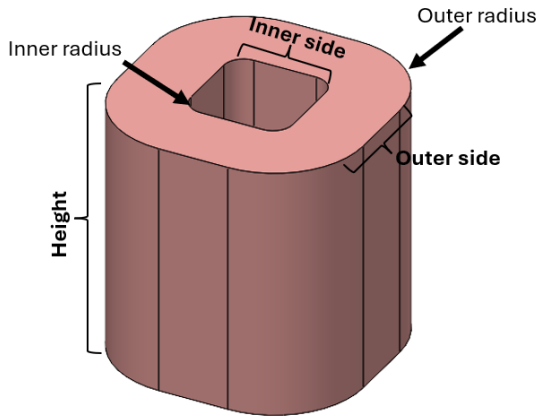


Figure 3.2: Single coil geometry and profile.

An industrial Beckhoff PLC (model C6670-0010) serves as the main controller for the coil current commands. In each cycle, which is configured to be computed by an isolated core every 1000 μs, the requested current for each individual coil is transmitted to the coil amplifiers via the EtherCAT protocol. Once the coils are actuated, a feedback system comprising four Vicon Vantage V5 cameras captures the motion of the levitated mover using the markers attached to its frame. The cameras send the collected data of each individual marker to a dedicated Vicon computer, which processes all cameras' information and computes the mover's pose with an accuracy reported in micrometers (μm) [58]. This pose is then transmitted as a UDP packet over an Ethernet connection to the PLC to be consumed every cycle. Figure 3.3 illustrates the flowchart of the Testbench system.

An external PC, referred to as the engineering PC, is utilized for development and serves as the user interface to the PLC. On the engineering PC, the developed wrench models, current calculation algorithms, controllers, and any additional features are tested and compiled for the PLC using TwinCAT software. An additional module provides a translation and compilation layer from Simulink to C++ code, which is subsequently loaded into the PLC as a TcCOM object.

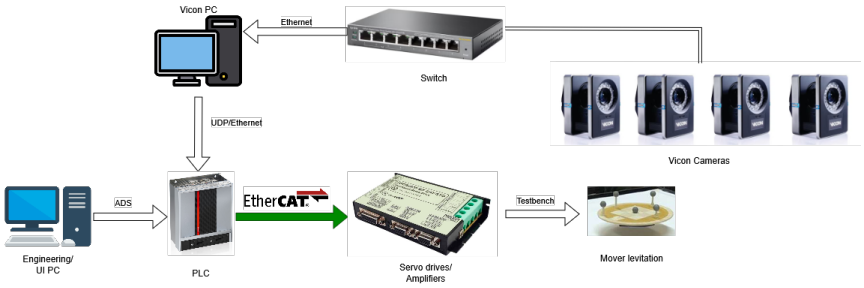


Figure 3.3: Overall communication Testbench setup.

3.2 Hardware Subsystems

The independent levitation capabilities of the Testbench allow for the use of various types of movers. This section reports the parameters of the specific levitated actuator studied in this work. Furthermore, by establishing a communication channel between the PLC and an external stream, the internal PLC variable handling system, known as ADS, can be exposed to additional devices. By exploiting real-time data regarding the mover's status, external hardware that relies on this information can be integrated into the system. Accordingly, this section also introduces the parameters of the robotic manipulator.

3.2.1 Mover System

Developed by Dr. Chanuphon Trakarnchaiyo and presented in [56], the four-disc mover is a compact planar actuator consisting of four N52 grade NdFeB permanent magnets. These magnets are arranged in a square configuration with an optimized central distance of 40 mm. The mover was originally designed to improve on the single-disc magnet mover, as the four-disc configuration enables yaw motion control. Although the techniques discussed in Section 2.1.2 are applicable for wrench modeling, a fixed-size lookup table (LUT) was implemented for the mover due to its simplicity and accuracy. During levitation, the control system activates a 4×4 array of coils centered around the mover's global X-Y position. Table 3.1 presents the geometrical parameters used for the construction of the mover, with a schematic diagram for reference.

Given the mover's geometry, mass, coil resistance, and coil activation methodology during levitation, the expected static power consumption can be approximated through simulations. Figure 3.5 showcases the Testbench power consumption with the mover at an air gap of 2 mm. The mover is simulated across the active 4×4 coil array without any

tilt or yaw angle. Under the simulated parameters, the mover requires a maximum power consumption of 160 W to levitate its own weight. The selection of the specific air gap and angle parameters for the mover simulation follows previously reported configurations from available commercial devices [3, 47].

Table 3.1: DMM parameters

Remanence (B_r)	1.48 T
Magnet Diameter ($\frac{d}{2}$)	50.8 mm
Magnet Thickness	12.7 mm
Mover Side (L_{dmm})	120 mm
Mover Thickness (H_{dmm})	19.8 mm
Magnets distance from center (R)	40 mm
Magnets arrangement angle (θ)	90°
Mover Mass	990 g
Mover Tilt Inertia [I_{xx}, I_{yy}]	0.0012 kg·m^2
Mover Yaw Inertia I_{zz}	0.0024 kg·m^2

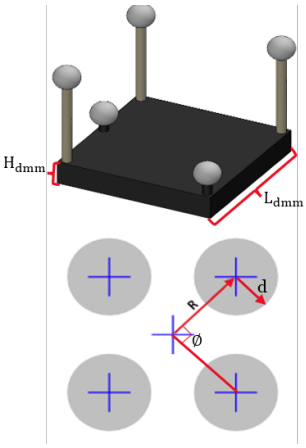


Figure 3.4: Mover schematic.

Following the PID controller embedded on the Testbench [53], the four DMM have been tuned with the parameters shown in Table 3.2. These parameters have been tested during levitation with speed and acceleration limits of 1000 mm/s and 2500 mm/s^2, respectively.

Table 3.2: PID Parameters

Axis	K_p	K_i	K_d
X	1200	6000	65
Y	1200	6000	65
Z	1200	6000	70
Roll (ϕ)	1.5	7	0.06
Pitch (θ)	1.5	7	0.06
Yaw (ψ)	2.0	8	0.07

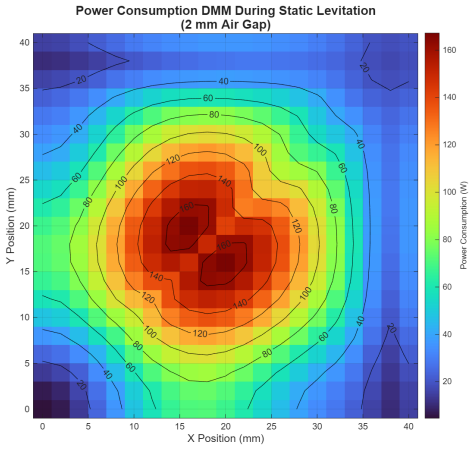


Figure 3.5: Simulated static power consumption.

3.2.2 Manipulator model and parameters

The robotic manipulator utilized in this study is a 4-DOF serial robotic arm with four revolute joints from the Igus Robolink series (RL-D-RBT-3322-BC) [26], as illustrated in Figure 3.6. Joint 1 provides rotation about a vertical base axis, while joints 2, 3, and 4 are mutually parallel and operate continuously within a vertical plane. Together, joints 2–4 form a planar link chain that controls the Cartesian position and the pitch angle of the end-effector. The end-effector itself is custom-made for operation and testing purposes, featuring attached markers for calibration during setup. The kinematic and physical parameters of the manipulator are defined in conventional Denavit-Hartenberg (D-H) form and are summarized in Table 3.3, where θ_i, d_i, L_i and α_i represent the joint angle variable, the link offset, the link length and the angle offset, respectively. The last column summarizes each joint kinematic limit, which prevents self-collision and exceeding the joint maximum angles. Due to the repeatability of the arm's motors, calculated to be $\pm 0.5^\circ$ by the manufacturer, the end effector repeatability at the end of the link is stated to be approximately ± 1 mm.

Table 3.3: D-H Parameters for the 4-DOF Manipulator

i	θ_i	d_i **(m)**	L_i **(m)**	α_i **(rad)**	**Limits (deg)**
1	θ_1	0.2175	0	$\pi/2$	$(-140, 140)$
2	θ_2	0	0.270	0	$(-45.0, 90.0)$
3	θ_3	0	0.240	0	$(-85.0, 55.0)$
4	θ_4	0	0.250	0	$(-30.0, 120.0)$



Figure 3.6: Igus 4-DOF manipulator.

As discussed in Section 2.2.2, the FK model of the robotic arm describes the pose of the end-effector with respect to the local arm base frame. By applying the elementary D-H transformations defined in Equation (2.13) and combining them sequentially, the complete analytical forward kinematics matrix ${}^{b}T_{a}$ is obtained:

$$ {}^{b}T_{a} = \begin{bmatrix} c_1c_{234} & -c_1s_{234} & s_1 & c_1(L_2c_2 + L_3c_{23} + L_4c_{234}) \\ s_1c_{234} & -s_1s_{234} & -c_1 & s_1(L_2c_2 + L_3c_{23} + L_4c_{234}) \\ s_{234} & c_{234} & 0 & d_1 + L_2s_2 + L_3s_{23} + L_4s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1} $$

where $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$, $c_{ijk} = \cos(\theta_i + \theta_j + \theta_k)$, and $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k)$. The fourth column of ${}^{b}T_{a}$ yields the Cartesian coordinates (x, y, z) of the end-effector tip, and the upper-left 3×3 submatrix defines its orientation relative to the base frame. This result is similar to Equation 2.14, with the addition of an extra joint to position the end-effector in the global X-Y plane.

To solve the IK problem, an analytical geometric approach is employed, similar to the 3-link planar manipulator kinematics and geometric inverse kinematics reviewed in Section 2.2.2 (specifically referencing Equations (2.15)-(2.18)) [13]. Given a target position (x, y, z) and pitch orientation ϕ, the joint variables $\mathbf{q} = [\theta_1, \theta_2, \theta_3, \theta_4]^\top$ are solved sequentially.

First, the base rotation angle θ_1 aligns the arm's vertical plane of interaction with the target coordinates:

$$ \theta_1 = \text{atan2}(y, x). \tag{3.2} $$

The pitch angle ϕ is obtained by extracting the approach vector $\vec{a}$ from ${}^{b}T_{a}$. This vector represents the local Z-axis of the end-effector relative to the world frame and is defined as $\vec{a} = [r_{13}, r_{23}, r_{33}]^\top$, where r_{ij} represents the element in the i-th row and j-th column of the transformation matrix. To find the equivalent ϕ, $\vec{a}$ must be projected onto the 2D vertical plane. The vertical component of the 2D approach vector maps directly to the arm frame's Z-axis. The radial component, a_r, is determined by projecting the X and Y components of $\vec{a}$ along the radial axis using the known base angle θ_1, such that $a_r = r_{13}\cos(\theta_1) + r_{23}\sin(\theta_1)$. Finally, the 2D planar pitch angle ϕ is calculated as:

$$ \phi = \text{atan2}(r_{33}, a_r). \tag{3.3} $$

Next, the coordinates of the wrist center (r_w, z_w) in the vertical plane are computed:

$$ r_w = \sqrt{x^2 + y^2} - L_4\cos(\phi), \tag{3.4} $$

$$ z_w = (z - d_1) - L_4\sin(\phi), \tag{3.5} $$

and the squared distance between the first revolute joint axis and the wrist center is computed as $D^2 = r_w^2 + z_w^2$.

The triangle formed by links L_2, L_3, and distance D is then solved using the law of cosines to find the inner joint angle θ_{in}:

$$\cos(\theta_{in}) = \frac{D^2 - L_2^2 - L_3^2}{2L_2L_3}, \tag{3.6}$$

$$\sin(\theta_{in}) = \pm\sqrt{1 - \cos^2(\theta_{in})}. \tag{3.7}$$

To satisfy safety constraints and clear the levitation air gap above the MLPA Testbench, an elbow-up configuration is enforced by selecting the positive root for $\sin(\theta_{in})$. The third joint angle is then computed, offset by $\pi/2$ to account for the physical folded state of the zero-angle configuration:

$$\theta_3 = \text{atan2}(\sin(\theta_{in}), \cos(\theta_{in})) - \frac{\pi}{2}. \tag{3.8}$$

Using the geometric relations within the triangle, the second joint angle is determined:

$$\theta_2 = \alpha + \beta, \tag{3.9}$$

where $\alpha = \text{atan2}(z_w, r_w)$ and $\beta = \text{atan2}(L_3 \sin(\theta_{in}), L_2 + L_3 \cos(\theta_{in}))$.

Finally, the fourth joint angle θ_4 is retrieved by subtracting the cumulative angles of the preceding links from the target pitch angle ϕ:

$$\theta_4 = \phi - (\theta_2 + \theta_3). \tag{3.10}$$

The robot arm servomotors are actuated via a CAN connection between an external PC and the stepper and digital I/O modules. Since the manipulator lacks an embedded controller, the proprietary iRC software from Igus (version 14) [27] is used to provide the communication layer. However, because this application only offers basic command tools and cannot incorporate the mover's current position, the software is configured to run as a server using the Commonplace Robotics Interface (CRI) [12]. This setup enables external client applications to visualize and modify the manipulator's parameters.

3.3 Control and Communication Architecture

Interaction between the robotic manipulator and the maglev device requires robust communication between the two systems. The robotic arm must know the mover's pose, both initially and continuously during operation, to determine its subsequent motions. Simultaneously, the mover must receive coordinated motion commands. This bidirectional communication is essential not only for basic collision avoidance but also for any type of path planning optimization.

As illustrated in the diagram in Figure 3.7, the mover's pose and parameters can be read by an external computer via an Ethernet connection using the ADS protocol from the maglev PLC. This external computer, designated as the planner PC, is responsible for synchronizing the actions between the robot arm and the MLPA device.

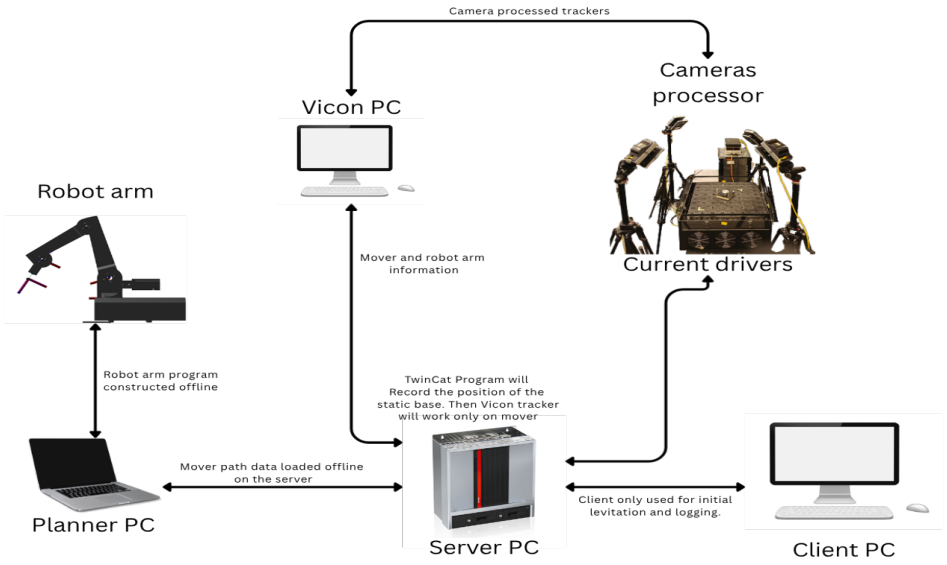


Figure 3.7: Diagram illustrating the connection and communication flow between the Testbench system and the robot arm control system.

While the maglev device operates at a 1 kHz cycle, the Igus manipulator can only be sampled and commanded at a much lower frequency. Consequently, the MATLAB-based client is designed to run at 50 Hz to comply with the requirements of the CRI connection [12]. In order to address this frequency mismatch and ensure smooth operation, a master-slave controller correlates the commands sent to the mover and the arm.

The interaction framework begins by initializing the software representations of the physical elements within the planner PC. Instances for the Testbench, the maglev mover, the robotic arm, and the target object are spawned in the simulation environment. The target object loads the predefined trajectory path, and its geometry is assigned to the mover, accounting for the physical offset required for transport. To guarantee an accurate shared reference frame, the base transformation of the robotic arm is calibrated relative to the global workspace using the real-time Vicon camera feedback. Once this initialization is complete and the collision-free waypoints for both systems are computed, the hardware execution phase begins.

With the initialization of the master-slave controller, the robotic arm acts as the master, dictating the pace of the movement, while the maglev mover operates as the slave. For each designated waypoint in the trajectory, the arm is commanded to move toward its target joint configuration. Because the arm operates at a slower and variable actuation speed compared to the high-frequency maglev device, sending simultaneous blind commands to both systems would lead to collisions or tracking errors. Instead, as the arm physically actuates, the planner PC samples its current joint positions at the 50 Hz client frequency. By comparing the arm's current state to its starting and target joint angles, a normalized progress variable (ranging from 0 to 1) is computed based on the joint that has to travel the maximum distance.

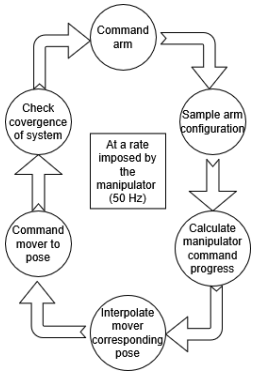


Figure 3.8: Diagram of the master-slave controller for motion commands between the systems.

This real-time progress metric is then utilized to perform a linear interpolation of the mover's trajectory, updating both its spatial translation and Euler angle orientation between the current and target Cartesian poses. The newly interpolated pose is immediately sent to the maglev PLC as a position command. This coupling ensures that the mover's

position is strictly bound to the arm's actual physical progress, seamlessly absorbing any mechanical delays or speed discrepancies from the robotic manipulator. The controller continuously verifies the position of both elements and only advances to the next waypoint when both the arm and the mover have converged to their respective targets within a strict error threshold, guaranteeing synchronized and safe operation throughout the entire interaction path. Figure 3.8 illustrates the controller cycle for any path loaded during interaction. The step-by-step logic of the master-slave execution, including the initialization, synchronized loop, and safe termination procedure, is summarized in Algorithm 1.

Algorithm 1 Master-Slave Controller Execution Logic

Require: Planned arm joint waypoints $\mathbf{Q}_{arm}$, Planned mover pose waypoints $\mathbf{P}_{mover}$

Command Mover to anchor pose $\mathbf{P}_{mover}[1]$ and wait for stabilization
Command Arm to initial configuration $\mathbf{Q}_{arm}[1]$ and wait for stabilization
for $i = 2$ **to** N **do**
    $q_{target} \leftarrow \mathbf{Q}_{arm}[i]$
    $p_{target} \leftarrow \mathbf{P}_{mover}[i]$
    $q_{start} \leftarrow$ Read current Arm configuration
    $p_{start} \leftarrow$ Read current Mover pose
    $\Delta q_{max} \leftarrow \max(|q_{target} - q_{start}|)$
    Command Arm to move to q_{target}
    while Arm and Mover have not reached target error thresholds **do**
        $q_{curr} \leftarrow$ Read current Arm configuration
        progress $\leftarrow \min\left(1.0, \frac{\max(|q_{curr} - q_{start}|)}{\Delta q_{max}}\right)$
        $p_{cmd} \leftarrow p_{start} + \text{progress} \times (p_{target} - p_{start})$
        Command Mover to p_{cmd}
    end while
    Command Mover to exact p_{target} to correct minor lag
end for
Execute sequential return-to-home safety protocol

Chapter 4

Implementation and Methodology Improvements

This chapter details the methodological improvements and the novel integration algorithms developed for the maglev Testbench and the robotic manipulator. Building upon the foundational models and hardware described in the previous chapters, the focus here shifts to the software, modeling, and planning architecture that enables precise, synchronized interaction between the two systems. Specifically, this chapter covers the enhancements made to the state estimation and physical modeling of the maglev system, followed by the robust path planning algorithm designed for cooperative execution.

4.1 System Enhancements and Calibration

As mentioned in Section 2.1, dynamically stable maglev devices require a robust controller to handle the changing fields during levitation in real time. As such, improvements on the control cycle of the system increase the performance and precision of the levitation drastically, making it as important as the fidelity of the wrench modeling scheme. In this section, the contributions to the controller loop are discussed.

4.1.1 Coil Position Mapping and Calibration

Wrench modeling for MLPAs depends heavily on the exact relative position of the mover magnets to the active stator coils, as formulated in the Lorentz force equations (2.5)

and (2.6). While analytical wrench models are derived under an idealized stator grid, with the coils centered at the plane $Z = 0$, physical assembly tolerances, structural deflections, and modifications to the electromagnet modules introduce misalignments. Positioning errors of the stator coils relative to the mover's frame can significantly distort the estimated force and torque matrices, leading to control inaccuracies or levitation instability.

To address these geometric discrepancies, a global calibration and mapping module was implemented. Unlike real-time adaptive strategies that continuously adjust parameters under load, this methodology performs a one-time global calibration of the Testbench stator using the readings from the same Vicon system as the previous levitation. Taking advantage of the geometric design of the coils, a fixture with a Vicon marker is momentarily fitted into the electromagnet's center. The readings from these markers are then implemented to warp the nominal coil grid to match the true physical layout. If no calibration is triggered, the system defaults to the ideal nominal coordinates. Since the positional data from the selected markers is used as the control points for the interpolation setup, it is necessary to reduce the induced tracking noise from the Vicon readings. Multiple recordings of the placed marker are collected for mean and variance calculation, where a standard deviation filter ($\pm 3\sigma$) is applied to refine the control point accuracy. The methodology not only improves the overall coil map interpolation output for the proposed methods, but also reduces the effect of overfitting on the final interpolated coil map.

Nominal Stator Grid Generation

The nominal grid serves as the coordinate seed for both the calibration routines and the initial uncalibrated state. The stator module consists of an 8×8 array of coreless coils with a nominal pitch $P = 76.2\,\text{mm}$. The nominal coordinates $(x_{\text{nom},k}, y_{\text{nom},k}, z_{\text{nom},k})$ for each linear coil ID $k = (r-1) \cdot 8 + c$ (where $r, c \in \{1, \ldots, 8\}$ are the row and column indices) are initialized relative to the module center as:

$$x_{\text{nom},k} = (c - 4.5)P, \tag{4.1}$$

$$y_{\text{nom},k} = (4.5 - r)P, \tag{4.2}$$

$$z_{\text{nom},k} = 0. \tag{4.3}$$

These coordinates are stored in a persistent 64×3 coordinate map variable and are only updated if the user commands a calibration with valid marker settings.

Calibration Points and Coordinate Mapping

For calibration, a subset of the 64 coils, indexed by their linear IDs, is selected as calibration points. At the center of each chosen coil, a framed Vicon marker is temporarily added to record the coil's physical position. The number of selected calibration points, M, is chosen to comply with the inequality equation $4 \leq M \leq 16$. This is done to ensure that, at minimum, a calibration of a stator square area is done. The measured coordinates $\mathbf{x}_{\text{marker},n} = [x_{\text{marker},n}, y_{\text{marker},n}, z_{\text{marker},n}]^\top$ for each marker $n \in \{1, \ldots, M\}$ are corrected by subtracting the vertical marker stand (fixture) height $z_{\text{stand}} = 13.5\,\text{mm}$ to retrieve the actual coil winding centers:

$$x_{d,n} = x_{\text{marker},n}, \quad y_{d,n} = y_{\text{marker},n}, \quad z_{d,n} = z_{\text{marker},n} - z_{\text{stand}}. \tag{4.4}$$

These adjusted target coordinates $\mathbf{x}_{d,n} = [x_{d,n}, y_{d,n}, z_{d,n}]^\top$ are paired with their corresponding nominal coordinates $(x_{s,n}, y_{s,n}) = (x_{\text{nom},k}, y_{\text{nom},k})$, where k is the linear coil ID associated with marker n. Figure 4.1 showcases the new graphical interface in the user GUI to report the found marker coordinates.

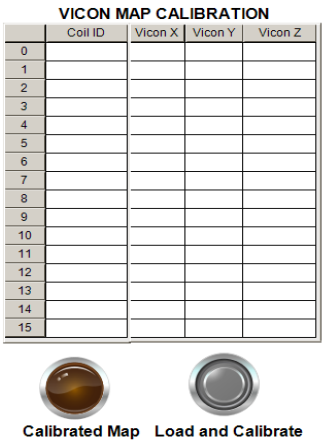


Figure 4.1: Interface for stator coil coordinates calibration.

Method A: Bilinear Corner Warp ($M = 4$)

When the minimum number of markers ($M = 4$) is placed at the corners of the module, a bilinear mapping is computed analytically. This method models the spatial deformation

as a hyperbolic paraboloid [6], mapping normalized nominal coordinates $u = x_{\text{nom}}/P$ and $v = y_{\text{nom}}/P$ to the calibrated coordinates:

$$x_{\text{cal}} = a_0 + a_1 u + a_2 v + a_3 uv, \tag{4.5}$$
$$y_{\text{cal}} = b_0 + b_1 u + b_2 v + b_3 uv, \tag{4.6}$$
$$z_{\text{cal}} = c_0 + c_1 u + c_2 v + c_3 uv, \tag{4.7}$$

where $\mathbf{a} = [a_0, a_1, a_2, a_3]^\top$, $\mathbf{b} = [b_0, b_1, b_2, b_3]^\top$, and $\mathbf{c} = [c_0, c_1, c_2, c_3]^\top$ are coefficient vectors. Normalizing the coordinates by the coil pitch P prevents numerical matrix conflicts during inversion. The 4×4 system matrix $\mathbf{A}_4$ is constructed from the normalized coordinates of the four corners:

$$\mathbf{A}_4 = \begin{bmatrix} 1 & u_1 & v_1 & u_1 v_1 \\ 1 & u_2 & v_2 & u_2 v_2 \\ 1 & u_3 & v_3 & u_3 v_3 \\ 1 & u_4 & v_4 & u_4 v_4 \end{bmatrix}. \tag{4.8}$$

The coefficients are solved independently, as with the inverse of matrix A [10]:

$$\mathbf{a} = \mathbf{A}_4^{-1}\mathbf{x}_d, \quad \mathbf{b} = \mathbf{A}_4^{-1}\mathbf{y}_d, \quad \mathbf{c} = \mathbf{A}_4^{-1}\mathbf{z}_d, \tag{4.9}$$

where $\mathbf{x}_d, \mathbf{y}_d, \mathbf{z}_d \in \mathbb{R}^4$ represent the target destination coordinate vectors of the corners. The calibrated positions for all 64 coils are then evaluated using:

$$\mathbf{X}_{\text{cal}} = \mathbf{A}_{\text{all}}\mathbf{a}, \quad \mathbf{Y}_{\text{cal}} = \mathbf{A}_{\text{all}}\mathbf{b}, \quad \mathbf{Z}_{\text{cal}} = \mathbf{A}_{\text{all}}\mathbf{c}, \tag{4.10}$$

where $\mathbf{A}_{\text{all}} \in \mathbb{R}^{64\times 4}$ contains the rows $[1, u_k, v_k, u_k v_k]$ evaluated for all 64 nominal coil positions.

Method B: Thin-Plate Spline ($5 \leq M \leq 16$)

If five or more markers are available, the Thin-Plate Spline (TPS) method is utilized. TPS is a radial basis function (RBF) interpolation technique that maps the nominal grid to the measured coordinates while minimizing the integral bending energy [5] . The mapping for any target dimension $v \in \{x, y, z\}$ is formulated as:

$$v_{\text{cal}}(x, y) = \sum_{j=1}^{M} w_{v,j} U(r_j) + a_{v,0} + a_{v,1} x + a_{v,2} y, \tag{4.11}$$

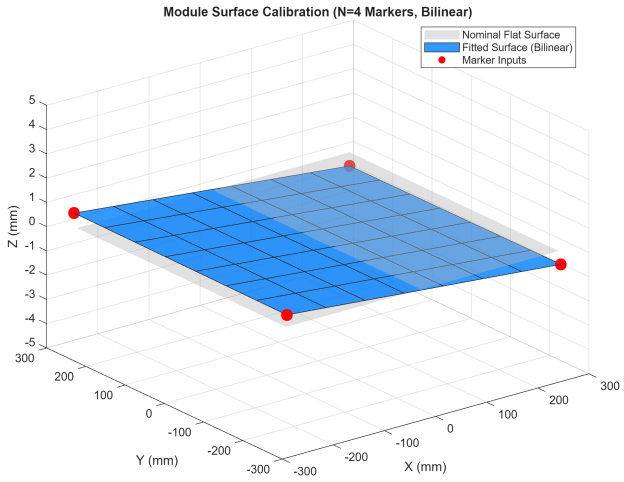


Figure 4.2: Bilinear mapping of the Testbench in the case of N=4 markers. The difference in control points is exaggerated for visual purposes.

where $U(r) = r^2 \ln(r)$ is the biharmonic RBF kernel, $r_j = \sqrt{(x - x_{s,j})^2 + (y - y_{s,j})^2}$ is the Euclidean distance to source marker j, $\mathbf{w}_v \in \mathbb{R}^M$ is the kernel weight vector, and $\mathbf{a}_v = [a_{v,0}, a_{v,1}, a_{v,2}]^\top$ represents the affine polynomial coefficients.

To determine the weights and polynomial coefficients, the system solves the following block matrix linear system:

$$\begin{bmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w}_v \\ \mathbf{a}_v \end{bmatrix} = \begin{bmatrix} \mathbf{v}_d \\ \mathbf{0} \end{bmatrix}, \tag{4.12}$$

where $\mathbf{v}_d \in \mathbb{R}^M$ is the measured destination coordinate vector. The kernel matrix $\mathbf{K} \in \mathbb{R}^{M \times M}$ and the polynomial matrix $\mathbf{P} \in \mathbb{R}^{M \times 3}$ are defined as:

$$K_{i,j} = \begin{cases} r_{i,j}^2 \ln(r_{i,j}) & \text{if } r_{i,j} > 0 \\ 0 & \text{if } r_{i,j} = 0 \end{cases}, \tag{4.13}$$

$$\mathbf{P} = \begin{bmatrix} 1 & x_{s,1} & y_{s,1} \\ 1 & x_{s,2} & y_{s,2} \\ \vdots & \vdots & \vdots \\ 1 & x_{s,M} & y_{s,M} \end{bmatrix}, \tag{4.14}$$

with $r_{i,j} = \sqrt{(x_{s,i} - x_{s,j})^2 + (y_{s,i} - y_{s,j})^2}$.

The linear system of dimension $(M+3) \times (M+3)$ is solved directly for each coordinate

direction to yield the weight and polynomial coefficients. Once computed, these parameters are evaluated at all 64 nominal coil coordinates $(x_{\text{nom},q}, y_{\text{nom},q})$ to generate the complete calibrated coil coordinate map.

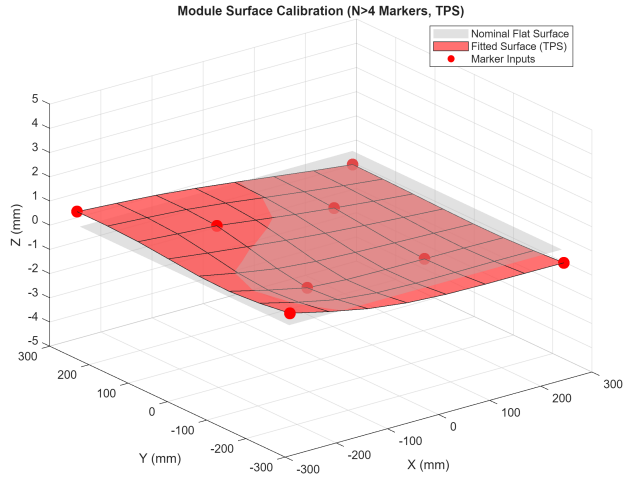


Figure 4.3: TPS surface fitting in the case of N>4 available coil markers. The difference in control points is exaggerated for visual purposes.

4.1.2 Kalman Filter State Estimation

To improve the robustness of the control loop, the Testbench device relies on continuous, accurate 6-DOF pose feedback from the Vicon camera system. However, optical tracking systems are inherently susceptible to occasional dropped frames, communication jitter, or temporary marker occlusions, which degrade pose accuracy. As such, in the original DMM implementation [55], a basic state estimation in conjunction with a PID closed-loop control was integrated. Velocity estimation was performed using simple finite backward difference with a low-pass filter that blended the current velocity with the previous frame estimate. While this helped the system to overcome camera missed frames, the controller blindly propagated the last known velocity. This approach is highly sensitive to measurement noise due to the lack of a physical model limitation, leading to erratic derivative control efforts as seen in Figure 4.4, potentially destabilizing the levitation during tracking gaps.

To resolve the discrete derivative issues, a Kalman filter (KF) was introduced to overcome the previous state estimation inconsistencies. The KF provides a robust, predictor-corrector framework that optimally fuses the noisy optical measurements with a dynamic

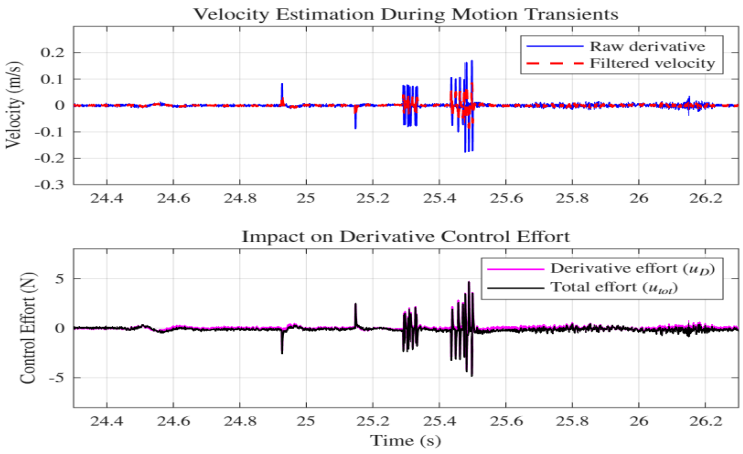


Figure 4.4: Original velocity estimation. The raw derivative is the current frame discrete velocity, while the filtered signal is after the low-pass filter. Note that even with the low-pass filter, since a frame was missed, a derivative-kick control effort generated a spike.

physical model of the mover [52]. The state vector is defined as a 12-dimensional vector $\mathbf{x} = [\mathbf{p}^\top, \mathbf{v}^\top]^\top \in \mathbb{R}^{12}$, comprising the 6-DOF pose $\mathbf{p} = [x, y, z, \theta_x, \theta_y, \theta_z]^\top$ and velocity $\mathbf{v} = \dot{\mathbf{p}}$ of the levitating DMM. The selection of the simple KF follows the requirement of an under 1 ms wrench-current cycle, and that the long trajectory model currently implemented adheres to a trapezoidal velocity curve.

During a long stroke motion, the DMM translates with a constant acceleration. Therefore, the wrench actuating on the mover follows:

$$\mathbf{W_{mover}} = \mathbf{M}\dot{\mathbf{v}},$$

where $\mathbf{M}$ is the mover mass and inertia in matrix form $\mathbf{M} = \text{diag}(m, m, m, I_{xx}, I_{yy}, I_{zz})$. For the studied DMM, the mass and inertia parameters were provided in Table 3.1. Given that the control input is the mover wrench, $\mathbf{W_{mover}} \rightarrow \mathbf{u} \in \mathbb{R}^6$, the model equation simply follows the solution of the model acceleration as,

$$\dot{\mathbf{v}} = \mathbf{M^{-1}u(t)}, \tag{4.15}$$

writing equation 4.15 in the common state-space representation,

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c(\mathbf{u}(t) + w_r), \tag{4.16}$$

where $w_r \sim \mathcal{N}(0, \mathbf{Q})$ is the process noise and the matrices $\mathbf{A}_c$ and $\mathbf{B}_c$ being represented by:

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{0}_{6\times6} & \mathbf{I}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} \mathbf{0}_{6\times6} \\ \mathbf{M}^{-1} \end{bmatrix}. \tag{4.17}$$

The prediction step of the filter leverages these known physical properties of the mover and the active control efforts. By discretizing this continuous-time model using a Zero-Order Hold (ZOH) over the constant sample time T_s, the discrete-time state transition matrix $\mathbf{A} \in \mathbb{R}^{12\times12}$ and the control input matrix $\mathbf{B} \in \mathbb{R}^{12\times6}$ are obtained:

$$\mathbf{A} = e^{\mathbf{A}_c T_s} = \begin{bmatrix} \mathbf{I}_6 & T_s\mathbf{I}_6 \\ \mathbf{0}_{6\times6} & \mathbf{I}_6 \end{bmatrix}, \tag{4.18}$$

$$\mathbf{B} = \left(\int_0^{T_s} e^{\mathbf{A}_c \tau} d\tau \right) \mathbf{B}_c = \begin{bmatrix} \frac{1}{2}T_s^2\mathbf{M}^{-1} \\ T_s\mathbf{M}^{-1} \end{bmatrix}. \tag{4.19}$$

To model the now discretized covariance matrix $\mathbf{Q} \in \mathbb{R}^{12\times12}$, the wrench uncertainty covariance, described as $\mathbf{\Sigma}_f = \text{diag}(\boldsymbol{\sigma}^2_{\text{model}})$ is used to derive $\mathbf{Q} = \mathbf{B}\mathbf{\Sigma}_f\mathbf{B}^\top$. In physical continuous kinematic systems, process noise typically enters through unmodeled forces or external disturbances acting on the accelerations. When discretized, this continuous noise must be mapped correctly into the discrete state space. Rather than using an arbitrary diagonal matrix that incorrectly assumes position and velocity errors as independent states, the $\mathbf{B}\mathbf{\Sigma}_f\mathbf{B}^\top$ formulation explicitly couples the disturbances of unmodeled accelerations and how they are integrated over the sample time T_s to simultaneously affect both velocity and position [2].

For the corrector step, the measurement model relies on the Vicon system, which provides positional feedback. The measurement equation is:

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_r, \tag{4.20}$$

where $\mathbf{C} = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0}_{6\times6} \end{bmatrix} \in \mathbb{R}^{6\times12}$ extracts the pose from the state vector, and $\mathbf{v}_r \sim \mathcal{N}(0, \mathbf{R})$ is the measurement noise with a diagonal covariance matrix $\mathbf{R} \in \mathbb{R}^{6\times6}$ containing the translational and rotational sensing variances.

The predictor-corrector nature of the Kalman Filter offers an elegant solution to the missing frame problem. The algorithm continuously monitors the incoming Vicon frame index. When a new, unique frame is received, the filter executes its corrector step. It computes the measurement residual and optimally updates the state estimate using the Kalman gain, while the state covariance is updated by the commonly known posteriori

error equation [62]. In the event of a dropped frame, the corrector step is bypassed. The system instead seamlessly carries forward the dynamic prediction ($\hat{\mathbf{x}} = \mathbf{x}_{\text{pred}}$), maintaining a continuous and physically consistent state estimate until visual feedback is restored.

The system process and signal covariance matrices, $\mathbf{Q}$ and $\mathbf{R}$, were tuned with the following procedure for the MLPA system. Acknowledging that $\mathbf{R}$ represents the covariance of the measurement noise of the visual tracking system, a levitation procedure is done where the tracked mover is held in an idle levitation pose for a determined amount of time. The resulting time series of measured position and orientation were treated as repeated measurements of a constant state. Under the assumption the noise is constant during levitation and the camera system does not suffer from any significant drift during recording, the sample variances of the positional components and the sample variances of the orientation components were computed from the recorded data and assigned to the corresponding diagonal entries of $\mathbf{R}$, where the $\sigma^2_{x,y,z}$ and $\sigma^2_{roll,pitch,yaw}$ correspond to those variances. Using the real sensor-recorded variance is common practice for KF parameter tuning [21]. In contrast to the measurement noise $\mathbf{R}$, the process noise covariance $\mathbf{Q}$ cannot be measured explicitly. The covariance values $\mathbf{Q}$ were initialized from an initial guess and then tuned empirically by analyzing the filter innovations on representative trajectories. The nominal $\mathbf{Q}$ was increased or decreased iteratively so that the innovation sequence approached zero-mean white noise. It is important to note that, in this specific maglev device, it is preferred to heavily trust the sensor recordings since the Vicon cameras are well-known calibrated and robust sensing systems. Table 4.1 reports the final Kalman Filter covariance values applied during the experiment data collection.

Table 4.1:
MLPA Kalman Filter Covariances

Axis	$\mathbf{Q}$	$\mathbf{R}$
X	3.2	7.25×10^{-10}
Y	3.2	7.25×10^{-10}
Z	8	7.25×10^{-10}
Roll (ϕ)	0.5	1.26×10^{-11}
Pitch (θ)	0.5	1.26×10^{-11}
Yaw (ψ)	1.0	1.26×10^{-11}

With this architecture improvement, not only were the prediction and control logic decoupled algorithmically, but a model-aware prediction was implemented in the system, increasing the robustness of the overall system, as seen in Figure 4.5, where the DMM was conducted through the same motion without the derivative term kicks.

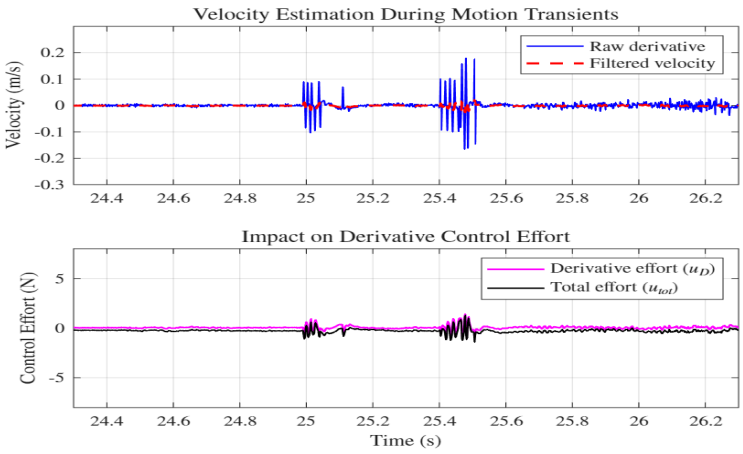


Figure 4.5: Kalman filter state estimation during the same long motion. Note the reduction of both the derivative response and the control effort during a missed frame.

4.2 Wrench implementation methodology

As the interaction capabilities of the MLPA are extended by operating in cooperation with a robotic manipulator, the underlying magnetic wrench calculation during levitation must adapt to the increased operational range. The initial rigid array lookup table (LUT) utilized during independent levitation control presents hard limitations when deployed in extended or cooperative dynamic tasks. Therefore, in this section, a dual deep neural network (DNN) system is introduced to mitigate these limitations and augment wrench modeling fidelity across the extended interaction workspace.

4.2.1 Limitations of the Rigid Array Lookup Table

The previous implementation of the MLPA controller relied on an analytical LUT that pre-calculated the wrench output for a fixed 16-coil (4×4) matrix acting on the mover array [57]. While this approach effectively reduced the real-time computation burden by grouping all coils into a single block and exploiting the underlying repetitive pattern of the DMM and stator, it suffered from a strictly limited range of action and a lack of accuracy near the borders of the electromagnet array. For instance, the maximum calculated wrench values for airgap and tilting angles were bounded to ranges of 16 mm $\leq Z \leq$ 30 mm and $-10° \leq \text{tilt}(\text{roll}, \text{pitch}) \leq 10°$, respectively. Figure 4.6 illustrates the spatial range of the LUT with respect to the fixed stator pattern.

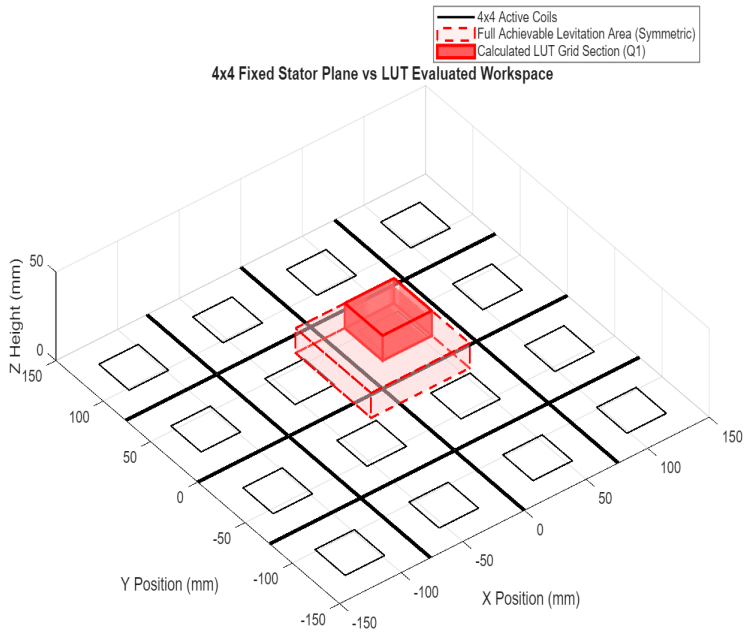


Figure 4.6: LUT range boundaries. Only the spatial coordinates are represented.

When the mover interacts with the end-effector of the robot arm, it frequently operates outside these pre-established boundaries. Although the closed-loop PID controller attempts to compensate for missing model data through position error feedback, the feedforward wrench output becomes capped at the maximum or minimum bounds stored in the table. This artificial saturation inherently degrades tracking performance during complex interaction trajectories.

Addressing this issue by simply expanding the spatial range of the rigid LUT proved impractical. Generating a grid-based table large enough to encompass the full dynamic workspace of the interaction model would have necessitated an enormous amount of offline computation. Furthermore, the resulting data structure would yield an excessive file size, leading to memory loading times that are prohibitively long at each levitation startup. These limitations motivated a shift from rigid, grid-based modeling toward data-driven surrogate models.

4.2.2 Data Generation via Sampling

To overcome the LUT limitations, a dual DNN setup was proposed. A fundamental shift in this approach is the modeling of a *mover–single coil* interaction rather than the rigid 16-coil

structure. By mapping a relative configuration pose to the wrench generated by a single coil, the control architecture decouples coil selection from magnetic wrench evaluation, enabling a dynamic selection of active coils that respond to the current mover pose.

To train the DNNs, a highly optimized dataset was required. Rather than using a uniform rigid grid, which rapidly suffers from the curse of dimensionality as the workspace is extended, the training set was designed to fill the configuration space as uniformly as possible while preserving an underlying structure that can be easily expanded in the future. Given these requirements, the training data were generated using a quasi-random Sobol sequence, spanning the 6-DOF continuous mover configuration space.

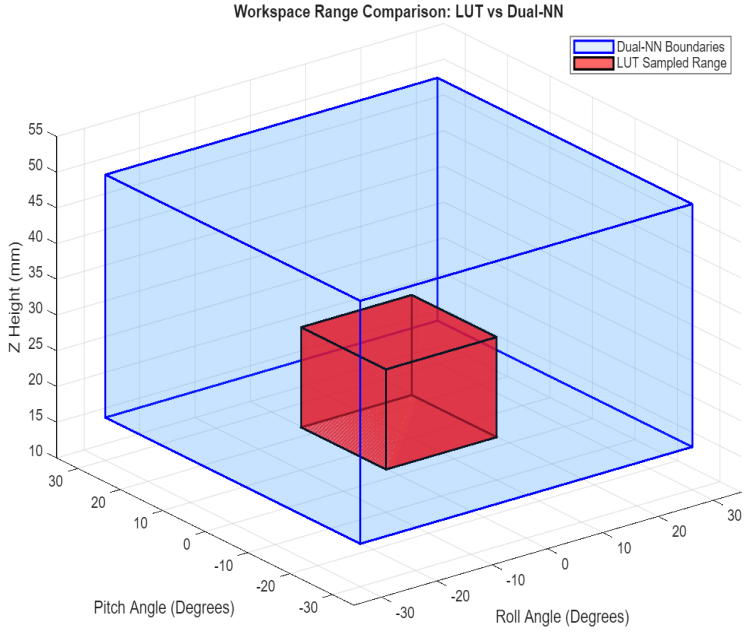


Figure 4.7: LUT range boundaries against the dual-NN setup for Z, roll, and pitch. The remaining DOFs maintain the same ranges between the comparisons.

Sobol sequences belong to the family of low-discrepancy, quasi-random sampling methods and are specifically designed to provide a nearly uniform coverage of high-dimensional parameter spaces while avoiding clustering and large empty regions [8]. In surrogate modeling and engineering simulations, such sequences have been shown to improve the accuracy and robustness of neural networks and response surface models compared to standard random or grid-based sampling, especially when the number of training samples is limited [8, 20]. By leveraging Sobol sampling, the proposed dataset achieves dense coverage of

the relevant wrench configurations while keeping the total number of simulated poses computationally tractable. Figure 4.7 illustrates the extended range of the dual-NN setup in comparison to the original LUT.

From Equations [2.1, 2.3], it is known that the magnetic field, and thus the generated wrench, becomes stronger as the magnets move closer to the stator array. As such, the dataset was generated in a split manner, where half of the samples obtained from the Sobol sequence were assigned to the lower airgap zone, concentrated in the range of 16 mm $\leq Z \leq$ 28 mm, where precise force modeling is most critical. The remaining 50% of the dataset was assigned to the high airgap zone, where the magnetic forces become more uniform, as seen in Figure 4.8. In addition, a collision filter was applied to the generated poses, guaranteeing that no samples corresponded to physically colliding configurations and preventing wasted computation during data generation and neural network training.

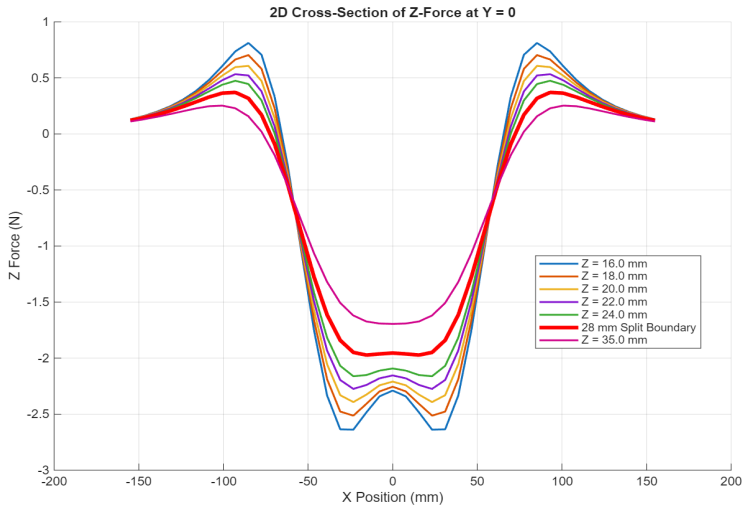


Figure 4.8: Change of the Z force on the mover with respect to the air gap. It is illustrated that at higher air gaps, the force pattern becomes nearly constant. The pattern begins to shift significantly around $Z = 28$ mm.

4.2.3 Dual Neural Network Architecture and Training

Due to the vast difference in magnetic field gradients, the full workspace dataset was divided across the Z-axis to train a dual network architecture. This architecture is composed of Residual Multi-Layer Perceptrons (ResMLPs) utilizing hyperbolic tangent (`tanh`) activation functions, and is further developed in Appendix A. The `tanh` activations were

deliberately chosen to provide smooth, bounded, and continuously differentiable force outputs, which are essential properties for stable real-time feedback control.

- **Near Network ($Z \leq 28$ mm):** Tasked with learning the sharp, highly non-linear force peaks occurring close to the stator.
- **Far Network ($Z > 28$ mm):** Focused on the smoother exponential decay of the magnetic field at larger distances.

Since the mover exhibits symmetry with respect to the stator array, the dual-NN setup is trained on a single spatial quadrant ($X \geq 0$, $Y \geq 0$), employing the same symmetry mirroring technique as the LUT implementation. Wrench evaluations in the remaining three quadrants are resolved dynamically by applying deterministic physical symmetry mapping rules during real-time levitation.

4.2.4 Controller Integration and Validation

By shifting to the dual neural network methodology, the parameter weights loaded by the PLC during system initialization were reduced from a required 7-minute idle loading time to $\sim 1\ ms$. In contrast with the LUT methodology, the dual-NN model requires an increased average execution time of 398 μs during levitation and control loop execution, which is an increase of ~ 7 times the duration of the LUT cycle (52 μs). While the neural network evaluations require a longer cycle time than the rigid LUT lookup, the proposed method still complies with the sub-millisecond real-time requirement (1000 μs) and, in turn, provides an extended workspace volume for the mover for a fraction of the required system memory storage if a similar-sized LUT was applied.

An often overlooked benefit of neural architectures is their ability to extrapolate, which is inherently impossible for a LUT approach. This was clearly shown during the validation step of the methodologies when compared to the analytical solution used on both LUT and dual DNN models, as shown in Table 4.2. Given the symmetry of both the mover and stator, the metrics for lateral forces (F_x, F_y) and tilt torques (T_x, T_y) exhibit similar statistical behavior. Consequently, these metrics were grouped as F_{xy} and T_{xy} for the analysis. To ensure a fair comparison between both methods, the sample range is divided into two sectors, bounded by the LUT's original limits. Thus, the sectors are *Inside LUT Range* (Z : 16–30 mm, Tilts: $\pm 10°$) and *Outside LUT Range* (Z : 30–60 mm, Tilts: $\pm 30°$).

The LUT model exhibited marginally better wrench predictions than the dual neural network setup during the small air gap zone. The LUT method is expected to yield

slightly better wrench predictions in this range, given the highly sampled population used to construct it, in contrast to the proposed neural network sampled database, which has to take into account the increased operational range. While this is true, the dual neural network showcased a competitive standard deviation metric, demonstrating that the method learned the mover's non-linear wrench values through this region.

Although both methods produced similar results at the sampled low air-gap boundaries, the limitations of the fixed LUT structure became evident as the planar actuator moved into the *Outside LUT Range*. The dual-NN outperforms the LUT at this volume, reducing all the axis-calculated RMSE metrics, the 3σ variance for F_z from 1.217 N to 0.968 N and nearly halving the T_{xy} 3σ error. Figure 4.9 further illustrates this behavior, visually showing the RMSE values across both sectors. The grouped bar charts show that both methods maintain safe control tolerances and similar relative error to the analytical version in the lower air-gap range, but the dual-NN clearly improves over the LUT method during high air-gap motions, especially for the calculated torques, showing that the proposed method supports the extended tilting ranges required for the manipulator cooperation.

Table 4.2: Total 16-Coil Error Comparison Relative to Analytical Baseline

	Inside LUT Range[1]		**Outside LUT Range[2]**	
Metric	**LUT**	**Dual-NN**	**LUT**	**Dual-NN**
RMSE F_{xy}	0.110 N	0.153 N	0.310 N	0.260 N
RMSE F_z	0.201 N	0.216 N	0.433 N	0.314 N
RMSE T_{xy}	5.8 mNm	7.4 mNm	26.9 mNm	13.2 mNm
RMSE T_z	5.0 mNm	4.3 mNm	12.5 mNm	7.8 mNm
3σ F_{xy}	0.324 N	0.459 N	0.876 N	0.806 N
3σ F_z	0.497 N	0.653 N	1.217 N	0.968 N
3σ T_{xy}	15.8 mNm	22.3 mNm	75.1 mNm	40.9 mNm
3σ T_z	14.2 mNm	13.2 mNm	35.3 mNm	24.4 mNm

[1] $Z \in [16, 30]$ mm, Tilt $\in [-10°, 10°]$.
[2] $Z \in (30, 60]$ mm, Tilt $\in [-30°, 30°]$.
Note: RMSE values are also visualized graphically in Fig. 4.9.

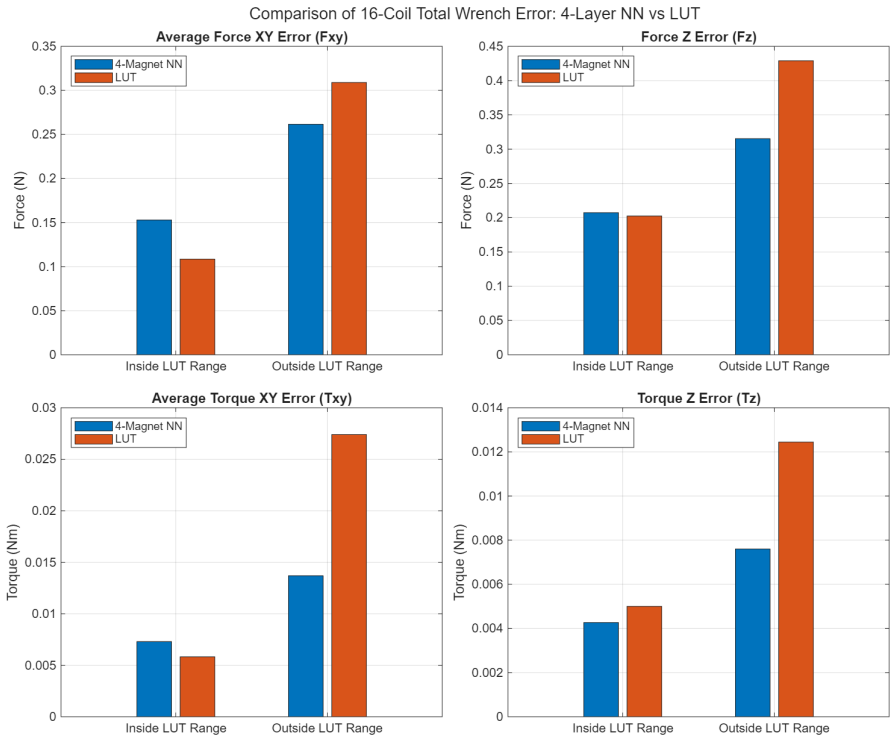


Figure 4.9: RMSE comparison across the entire workspace range

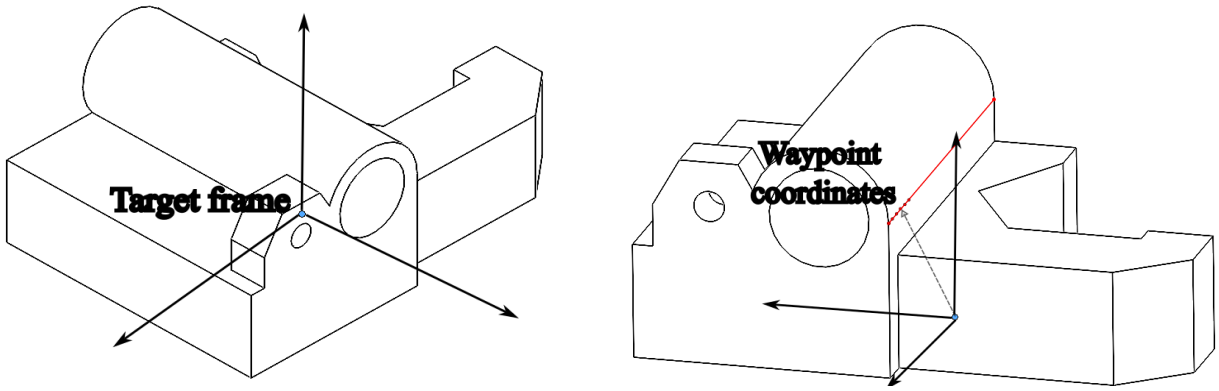
4.3 Interaction Path Generation

Any robotics or path planning algorithm requires the definition of the objective task, whether it be the parking of a robot or the manipulation of an object. For the MLPA interaction with an external manipulator, the goal defined conveys the completion of a 5-DOF trajectory by the end effector of the robot arm, which is a process typically developed for manufacturing procedures such as welding, painting, part inspection, and assemblies. In this section, the generation of the task trajectory is developed.

4.3.1 Mesh-Based Trajectory Extraction

Prior to generating the task trajectory, the object involved in the interaction between the DMM and the manipulator, referred to as the **target**, must be defined. Assuming the target's mesh model is available, as is standard in the manufacturing industry,

a local coordinate frame ${}^{tgt}T$ can be assigned to it. Within this local frame, a discretized trajectory S_p is constructed, comprising a finite set of N_w waypoints such that $S_p = \{{}^{tgt}T_{w_1}, {}^{tgt}T_{w_2}, \ldots, {}^{tgt}T_{w_{N_w}}\}$, where ${}^{tgt}T_{w_k}$ represents the transformation pose of the k-th waypoint. As an example, Figure 4.10 illustrates a target object assigned a discretized straight-line path defined within its local frame. The target mesh and the trajectory



(a) Target frame definition on the local geometry center.

(b) Relative waypoint coordinates to the local target frame.

Figure 4.10: Definition of the waypoint coordinates with respect to the target frame.

waypoints are defined using the CAD software SolidWorks. These discretized waypoints establish the spatial coordinates that the manipulator's end effector must reach to execute the task.

However, a single trajectory only specifies the positional coordinates within the target frame; it is insufficient to fully define the 5-DOF pose, as it lacks the required end-effector orientation (i.e., the angle of attack). To address this, an auxiliary orientation trajectory is introduced. While this secondary trajectory can assume an arbitrary shape, it must contain the same number of waypoints (N_w) as the primary path to fully constrain the task. As shown in Figure 4.11, this orientation trajectory dictates the angle of attack. Conceptually, the orientation trajectory defines the origin of the approach vector, while the primary trajectory dictates its destination and the required contact coordinates.

At this stage, the task trajectory is defined exclusively within the target's local frame. To express these waypoints in the global coordinate frame, to enable communication with the manipulator, the following kinematic chain transformation is applied:

$$T_{w_k} = T_m \, {}^{m}T_{tgt} \, {}^{tgt}T_{w_k}, \tag{4.21}$$

where T_m denotes the global position of the DMM's geometric center, ${}^{m}T_{tgt}$ represents the target's offset relative to the mover (determined by the physical fixture), and ${}^{tgt}T_{w_k}$ is the discretized path waypoint relative to the target. Figure 4.12 depicts the complete assembly of the mover, target, and task trajectory mapped into the global frame. The target mesh and both trajectories are exported as STL and CSV format from SolidWorks, respectively. This is done to generate each waypoint ${}^{tgt}T_{w_k}$ matrix and for the collision verification of the robot arm-target-mover assembly.

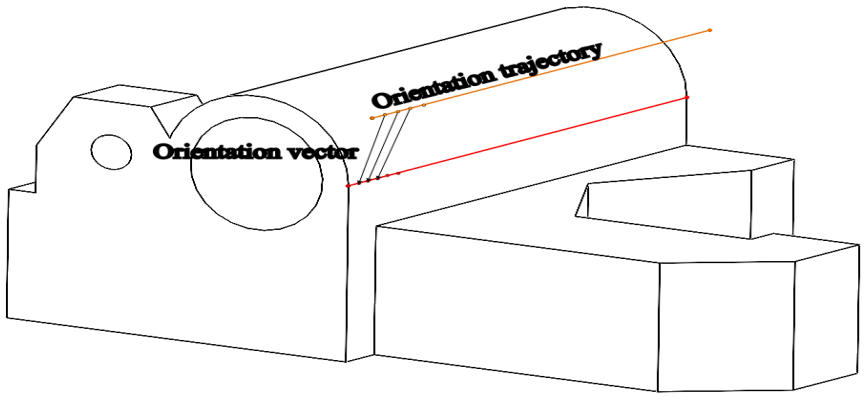


Figure 4.11: Definition of the orientation trajectory. Note the direction of the vector obtained from using the orientation and the coordinate trajectory.

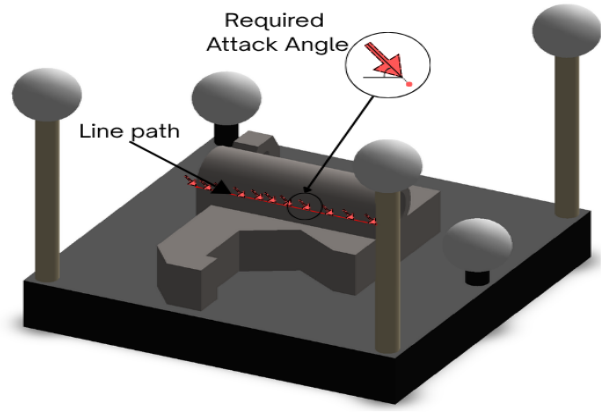


Figure 4.12: Assembly of mover, target and trajectory path representation.

4.4 Cooperative Path Planning and Workspace Filtering

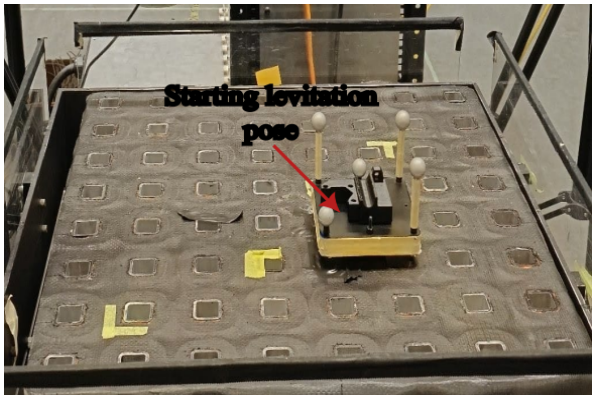
With the physical trajectory defined, the core contribution of this work lies in generating an optimal and collision-free path for both the maglev mover and the robotic arm. In this section, the continuous robot arm workspace is discretized with the models reviewed in Section 3.2.2, with an introduced condition on the workspace filtering provided by the trajectory objective. Finally, a Jacobian-based path planner is proposed to generate the cooperative configuration chain between both mover and robotic arm systems.

4.4.1 Definition of the Interaction Process

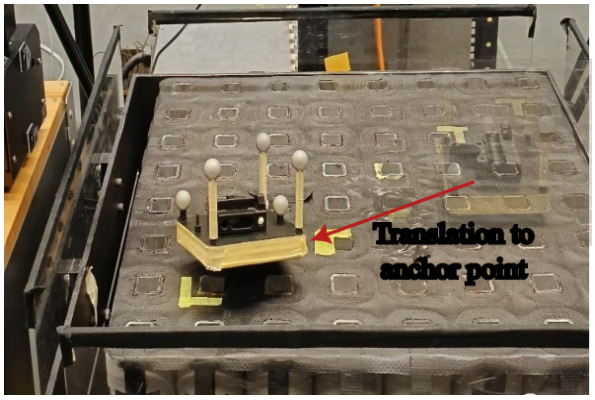
In industrial manufacturing settings, the utilization of external positioning devices typically follows a three-step procedure: an initial object motion step where the part is delivered to the fixed workspace of the robot arm; an interaction step where the manipulator's tool executes the assigned task based on its capabilities; and a final delivery step where the finished product is moved to the next station in the production line. To explore the omnidirectional and flexible orientation capabilities provided by the maglev addition, a comparable integration methodology is proposed with the following objectives, as visualized in Figure 4.13.

First is an object motion step, where the mover, levitating at an arbitrary starting point (Figure 4.13a), reaches a designated global coordinate position within the Testbench and anchors itself (Figure 4.13b). Once anchored, the mover's spatial translation is fixed, leaving only rotational motions available. Second, the interaction step where the robotic arm, with a fixed base, and the mover cooperate in parallel (Figure 4.13c). During this stage, the arm's end effector tracks the required Cartesian coordinates of the task, while the mover simultaneously rotates to fulfill the orientation requirements of the 5-DOF target trajectory, all while optimizing the interaction time. Third, the return step where the arm retracts to its home position and the mover returns to its original starting pose (Figure 4.13d). The overall time evaluated and optimized by the planner is the combination of these three consecutive steps.

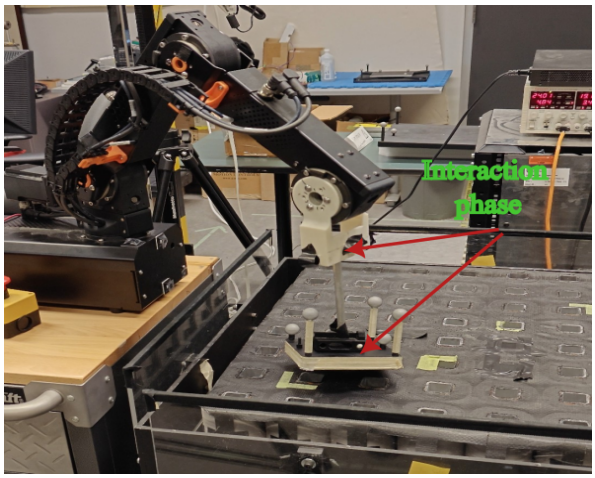
Constraining the mover's spatial position upon the initiation of the interaction serves to establish a robust methodological baseline, acts as a critical safety measure, and provides a decoupling strategy to systematically reduce the dimensionality of the overall configuration space.



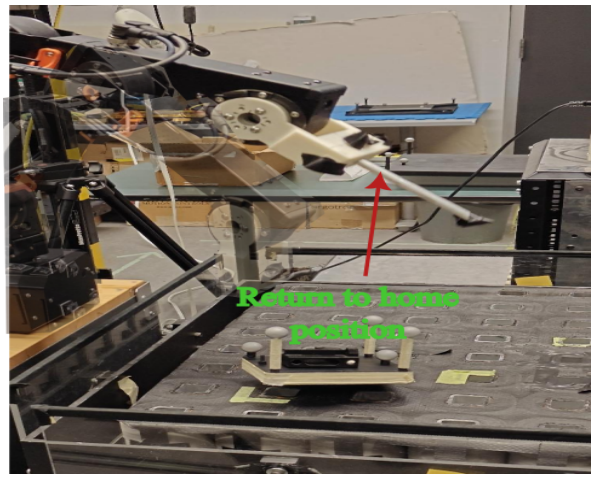
(a) Starting mover pose



(b) Translation to anchor



(c) Interaction phase at anchor



(d) Return to home

Figure 4.13: The cooperative interaction sequence outlining the initial state, translation, interaction, and return phases.

4.4.2 Workspace Filtering and Testbench Constraints

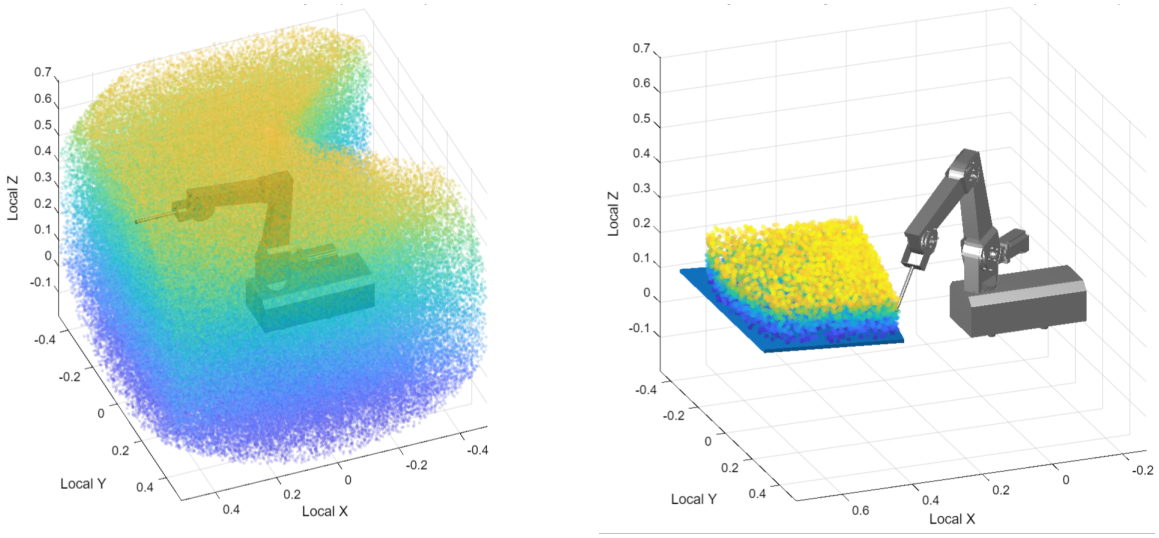
As discussed in Section 2.2.2, a computationally efficient way to represent the reachable workspace of a manipulator is through a randomized sampling discretization using a Monte-Carlo drawing method. The joint configuration q_{test} is generated by sampling each joint variable θ_i from a continuous uniform distribution bounded by its mechanical and self-collision limits $[\theta_{i,min}, \theta_{i,max}]$. The corresponding end effector pose is then calculated using the developed analytical forward kinematics model.

However, for the cooperative interaction task, the raw generated workspace must be restricted. Each sampled pose is subjected to two distinct filters. The first is a collision check that verifies the entire arm structure against the static environmental obstacles of the Testbench. The second is a geometric boundary filter that discards any end effector positions falling outside the allowable levitation area of the MLPA and a safety boundary range to eliminate possible singular configurations produced by the manipulator. This ensures that the arm only attempts interactions within the spatial domain where the mover can safely operate, and the mathematical pipeline developed (geometrical inverse kinematics and the path-planner solver) does not fail during the exploration of the space due to the loss of matrix dimensions at any point. This sampling and filtering loop is iterated until a sufficiently dense set of N_{test} valid configurations is obtained, representing the feasible cooperative workspace. Figure 4.14 showcases this iterative process, where a final number of possible manipulator configurations are left.

4.4.3 Anchor Generation and Selection

With the arm's workspace discretized and filtered, the cooperative interaction task requires the system to track the geometric path S_p inscribed on the target object. To successfully complete S_p, the combined motions of the robot arm and the mover must satisfy the kinematic matching constraint such that the global end-effector transformation aligns with every target waypoint along the trajectory.

To define the mover anchor pose in the global frame, subject to the feasibility check of completing S_p, an initial screening over all the surviving q_{test} configurations is performed as follows. Let the end-effector global transformation matrix be T_a, obtained through the kinematic chain $T_a = T_b {}^bT_a$, where T_b is the global pose of the fixed robot arm base and ${}^bT_a$ is the relative end-effector pose to the base. By applying the manipulator forward kinematics (FK) model with input q_{test} to find ${}^bT_a$, the required mover pose T_m to comply with any waypoint k is given by Equation 4.22, where the relative mover pose ${}^{w_k}T_m$ is



(a) Unfiltered robot arm workspace

(b) Filtered robot arm workspace

Figure 4.14: Levitation area arm workspace filtering. Dimensions in meters

found by ${}^{w_k}T_m = ({}^{m}T_{tgt}\,{}^{tgt}T_{w_k})^{-1}$.

$$T_m = T_a\,{}^{w_k}T_m. \tag{4.22}$$

Instead of selecting individual pathway points as anchors and blindly testing the feasibility of each one, an optimal average path pose ${}^{tgt}T_{w_{avg}}$ is defined to minimize the aggregate spatial deviations from all N_w waypoints. Decomposing this optimal state into a Cartesian position vector $\mathbf{p}_{avg}$ and an orientation quaternion $\mathbf{Q}_{avg}$, the formulation is decoupled into a translational and a rotational optimization problem, given by

$$\mathbf{p}_{avg} = \arg\min_{\mathbf{p}\in\mathbb{R}^3} \sum_{k=1}^{N_w} \|\mathbf{p} - \mathbf{p}_{w,k}\|^2, \text{ and} \tag{4.23}$$

$$\mathbf{Q}_{avg} = \arg\min_{\mathbf{Q}\in S^3} \sum_{k=1}^{N_w} \|A(\mathbf{Q}) - A(\mathbf{Q}_{w,k})\|_F^2, \tag{4.24}$$

where $\mathbf{p}_{w,k}$ and $\mathbf{Q}_{w,k}$ are the position and quaternion representations of ${}^{tgt}T_{w_k}$, $A(\mathbf{Q})$ denotes the rotation matrix corresponding to the quaternion $\mathbf{Q}$, and $\|\cdot\|_F$ is the Frobenius norm [41]. Figure 4.15 illustrates the average path pose for a circular path on a DMM with a start target object. It is this averaged pose that is tested for the initial anchor screening feasibility.

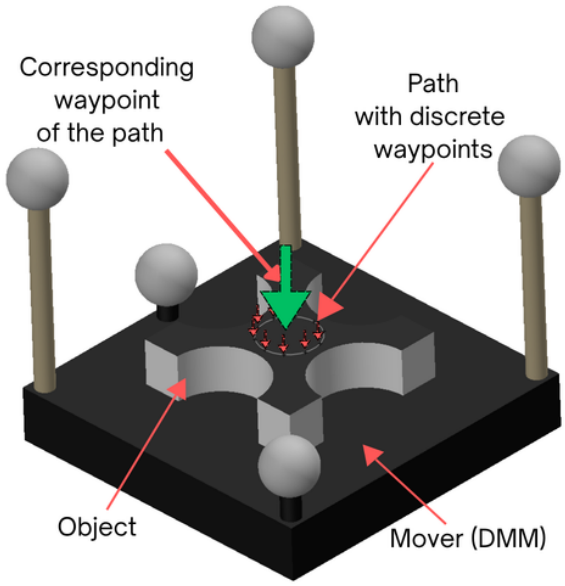


Figure 4.15: DMM and object with assigned circular path. Dots represent the coordinate position of the waypoints and the red arrow the desired interaction orientation.

Once the optimal average path pose ${}^{tgt}T_{w_{avg}}$ is found, the corresponding optimal mover anchor pose T_f is calculated by substituting ${}^{tgt}T_{w_{avg}}$ into Equations 4.21 and 4.22. Because the anchored mover functionally acts as a spherical wrist restricted to pure rotation, minimizing the spatial and rotational deviations of the path inherently minimizes the required angular effort of the mover's joints α_i for all $i \in \{1, 2, 3\}$, which represent the mover's roll, pitch, and yaw, respectively.

Each of the N_{test} surviving q_{test} configurations yields a possible global anchor pose T_f, which is then tested for collision and kinematic limit boundaries. The resultant valid T_f configurations are used as the fixed global coordinate constraints for the time-optimal path completion algorithm. Any non-compliant anchor candidate is discarded, since a failure implies a collision for at least one of the waypoints w_k in the static obstacle configuration space.

4.4.4 Optimization and Trajectory Generation

With a set of surviving anchor candidates, the core contribution lies in generating a time-optimal, collision-free configuration chain. The interaction model, once the mover is an-

chored, operates as an overactuated system. Following standard Jacobian-based redundancy resolution methodologies for redundant manipulators revised in Section 2.2.2, the algorithm is divided into two coupled stages.

First, a local Jacobian-based solver generates multiple feasible configurations at each waypoint by exploiting the redundancy of the 5-DOF point-of-contact task. For a fixed anchor position, the local optimization variables are defined as $\mathbf{z} = [\alpha_1, \alpha_2, \alpha_3, \phi]^\top$, where α_i are the mover orientation angles and ϕ is the arm pitch variable. At waypoint k, the residual error vector $\mathbf{e}$ is defined as

$$\mathbf{e} = \begin{bmatrix} \mathbf{p}_a - \mathbf{p}_{w_k} \\ \mathbf{r}_\perp \end{bmatrix}, \qquad \mathbf{r}_\perp = (\mathbf{r}_{\text{axang}} - \mathbf{r}_{\text{axang}}^\top \hat{\mathbf{z}}_a)\hat{\mathbf{z}}_a, \tag{4.25}$$

where $\mathbf{p}_a$ and $\mathbf{p}_{w_k}$ are the tool tip and waypoint positions, $\mathbf{r}_{\text{axang}}$ is the axis-angle orientation error between the tool frame and the waypoint frame, and $\hat{\mathbf{z}}_a$ is the tool's local Z-axis. The projection $\mathbf{r}_\perp$ removes the orientation error component corresponding to the free twist about the tool axis, reducing the fully constrained 6-DOF task to a 5-DOF point-of-contact task.

A finite difference Jacobian $\mathbb{J} = \frac{\partial \mathbf{e}}{\partial \mathbf{z}}$ is computed with respect to the variables in $\mathbf{z}$. The local iterative update combines a primary correction of the residual with a secondary objective projected into the task's null space, similar to the null-space projection introduced for coil currents in Equation 2.11. This update is given by

$$\Delta \mathbf{z} = -\mathbb{J}^{\#}\mathbf{e} + (\mathbf{I} - \mathbb{J}^{\dagger}\mathbb{J})\nabla\Phi, \tag{4.26}$$

where $\mathbb{J}^{\#}$ denotes the damped pseudoinverse and $\mathbb{J}^{\dagger}$ forms the null-space projector. The secondary objective gradient $\nabla\Phi$ utilizes a weighting matrix $\mathbf{W}$ to penalize deviations from a preferred state $\mathbf{z}_{\text{pref}}$:

$$\nabla\Phi = \mathbf{W}(\mathbf{z}_{\text{pref}} - \mathbf{z}). \tag{4.27}$$

The weighting matrix penalizes deviations in the mover's roll and pitch more strongly than its yaw, encouraging the mover to remain close to its preferred orientation while satisfying the interaction constraints.

Each converged solution is checked for collisions and stored as a feasible candidate. To capture the full redundancy and avoid collapsing into a single local solution, additional nearby candidates are generated by perturbing $\mathbf{z}$ along the 2D null space $\mathcal{N}(\mathbb{J})$.

Second, these feasible candidates are organized into a layered search graph. Each layer corresponds to one trajectory waypoint, and each node represents a collision-free configuration for that waypoint. Edges connect configurations in consecutive layers, with

weights assigned based on the maximum parallel execution time required for both the arm and the mover to transition between those states. The graph is augmented with a start node (accounting for the initial mover approach time t_s) and an end node (accounting for the return-to-home time t_h).

Finally, Dijkstra's algorithm is applied to compute the shortest path through this graph, selecting the sequence of waypoint configurations that minimizes the total overall completion time t_c. This yields a fully synchronized and optimal cooperative trajectory ready for execution by the master-slave controller. The complete procedure is summarized in Algorithm 2.

Algorithm 2 Null-Space Configuration Search for Anchored Mover

Require: Candidate anchors $\mathcal{A}$, waypoints $\{w_k\}_{k=1}^{N_w}$, tolerance ε
Ensure: Optimal anchor f^* and configuration sequence $\mathcal{Q}^*$
$t_c^* \leftarrow \infty$, $\mathcal{Q}^* \leftarrow \emptyset$
for all $f \in \mathcal{A}$ **do**
  Init graph $\mathcal{G}_f$ and seeds $\mathcal{S}_1$ for $\mathbf{z} = [\alpha_1, \alpha_2, \alpha_3, \phi]^\top$
  for $k = 1, \ldots, N_w$ **do**
    $\mathcal{V}_k \leftarrow \emptyset$
    for all $\mathbf{z}_0 \in \mathcal{S}_k$ **do**
      $\mathbf{z} \leftarrow \mathbf{z}_0$
      repeat
        Recover IK $\boldsymbol{\theta}$; compute $\mathbf{e} = [\mathbf{p}_a - \mathbf{p}_{w_k}, \mathbf{r}_\perp]^\top$ and $\mathbb{J} = \partial \mathbf{e} / \partial \mathbf{z}$
        $\Delta \mathbf{z} \leftarrow -\mathbb{J}^{\#}\mathbf{e} + (\mathbf{I} - \mathbb{J}^{\dagger}\mathbb{J})\mathbf{W}(\mathbf{z}_{\text{pref}} - \mathbf{z})$; $\mathbf{z} \leftarrow \mathbf{z} + \Delta \mathbf{z}$
      until $\|\mathbf{e}\| < \varepsilon$ or max iterations reached
      if $\|\mathbf{e}\| < \varepsilon$ and collision free **then**
        Add $(\boldsymbol{\theta}, \boldsymbol{\alpha})$ as node $v \in \mathcal{V}_k$
      end if
    end for
    if $\mathcal{V}_k = \emptyset$ **then**
      Discard anchor f and **continue**
    end if
    Gen. $\mathcal{S}_{k+1}$ by perturbing $\mathcal{V}_k$ along $\mathcal{N}(\mathbb{J})$; Connect $\mathcal{V}_{k-1} \to \mathcal{V}_k$ with cost c_{ij}^{k-1}
  end for
  Add start/home edges; Compute shortest path $\mathcal{Q}_f$ in $\mathcal{G}_f$ (Dijkstra)
  if $t_c(\mathcal{Q}_f) < t_c^*$ **then**
    $t_c^* \leftarrow t_c(\mathcal{Q}_f)$, $\mathcal{Q}^* \leftarrow \mathcal{Q}_f$, $f^* \leftarrow f$
  end if
end for
return $f^*, \mathcal{Q}^*$

4.4.5 Arm-only Baseline Configuration Search

For comparison during experimental validation, an arm-only baseline algorithm is defined by imposing the additional constraint that the mover remains fixed at the selected anchor pose throughout the interaction phase. Under this strict restriction, the coordinate transformation for each trajectory waypoint must be generated directly from the constant anchor pose, and the arm configuration is recovered independently via the analytical IK model using the corresponding waypoint position and required planar pitch. The same layered graph for optimization is applied as before. The baseline procedure is detailed in Algorithm 3.

Algorithm 3 Configuration Search for Arm-only Baseline

Require: Candidate anchors $\mathcal{A}$, waypoints $\{w_k\}_{k=1}^{N_w}$
Ensure: Optimal anchor f^* and configuration sequence $\mathcal{Q}^*$
$t_c^* \leftarrow \infty$, $\mathcal{Q}^* \leftarrow \emptyset$
for all $f \in \mathcal{A}$ **do**
  Initialize layered graph $\mathcal{G}_f$
  Extract fixed anchor pose $(\mathbf{p}_f, \boldsymbol{\alpha}_f)$
  for $k = 1, \ldots, N_w$ **do**
    $\mathcal{V}_k \leftarrow \emptyset$
    Extract target position $\mathbf{p}_{w_k}$ and planar pitch ϕ_k
    if target approach direction is incompatible **then**
      continue
    end if
    Recover arm joints $\boldsymbol{\theta}_k$ from analytic IK using $(\mathbf{p}_{w_k}, \phi_k)$
    if IK succeeds and configuration is collision-free **then**
      Add $(\boldsymbol{\theta}_k, \boldsymbol{\alpha}_f)$ as node $v \in \mathcal{V}_k$
    end if
    if $\mathcal{V}_k = \emptyset$ **then**
      Discard anchor f and **continue** to next anchor
    end if
    if $k > 1$ **then**
      Connect $\mathcal{V}_{k-1}$ to $\mathcal{V}_k$ with cost $c_{ij}^{k-1} = t_{\text{arm}}$
    end if
  end for
  Add start and home-return edges to account for t_s and t_h
  Compute shortest path $\mathcal{Q}_f$ in $\mathcal{G}_f$ using Dijkstra's algorithm
  if $t_c(\mathcal{Q}_f) < t_c^*$ **then**
    $t_c^* \leftarrow t_c(\mathcal{Q}_f)$, $\mathcal{Q}^* \leftarrow \mathcal{Q}_f$, $f^* \leftarrow f$
  end if
end for
return $f^*, \mathcal{Q}^*$

Chapter 5

Experimental Results and Discussion

This chapter details the experimental validations conducted to evaluate the performance of the proposed methodologies. The first section analyzes the improvements in the mover's state estimation and control using the developed predictive Kalman Filter mapping, comparing it against the original rigid table system. The second section focuses on the cooperative path planning pipeline, deploying the fully integrated robotic arm and MLPA system to complete complex trajectory tasks, comparing the cooperative approach against the arm-only baseline.

5.1 Mover Model Calibration and State Estimation

Before evaluating the cooperative path planner, the core enhancements to the mover's state estimation, wrench calculation, and control loop were validated. The developed model, which incorporates the calibration routine for active coil mapping, the new wrench modeling, and a revamped prediction step using a Kalman Filter, was compared against the original LUT model. To comprehensively assess improvements in tracking performance and stability, the system was subjected to two distinct long-stroke paths: a circle and a rectangle at two different air gaps.

The geometric and dynamic parameters defining these two experimental paths are detailed in Table 5.1 and Table 5.2. For both paths, the tilting angles and mover Z coordinate were maintained at a constant 0° and 24 mm, respectively, throughout the entire trajectory.

Table 5.1: Parameters of the Circular Long Stroke Path

Parameter	**Value**
Starting Pose (X, Y)	[-80, 0] mm
Radius	170 mm
Speed	1000.0 mm/s
Acceleration	2500 mm/s^2

Table 5.2: Parameters of the Rectangular Long Stroke Path

Parameter	**Value**
Starting Pose (X, Y)	[-180, -180] mm
Side Length (X)	360 mm
Side Length (Y)	360 mm
Speed	1000.0 mm/s
Acceleration	2500 mm/s^2

5.1.1 Path Tracking Performance

The tracking performance of both the original and improved system models was evaluated while executing the defined paths. Figure 5.1 displays the experimental results for one of the recorded circular paths at a low air gap. Both models tracked the circle with relatively similar positional accuracy; however, the enhanced prediction stability of the modified model produced a noticeably smoother trajectory with significantly less jitter, which is particularly evident in the Z-axis tracking.

A similar conclusion can be drawn for one of the single runs recorded for the rectangular path shown in Figure 5.2. While the modified model generally maintained smoother tracking, it exhibited slightly higher tracking errors in certain localized regions of the path. This behavior is theorized to be a byproduct of reusing the identical controller gains from the original LUT model, which had been explicitly tuned for a restricted 4×4 coil module configuration rather than the full extended workspace. Nonetheless, it is clear that when the mover accelerated out of the corners, the original model suffered from control effort spikes due to the noisy nature of its raw velocity predictor, generating significantly higher tracking deviations compared to the smooth response of the improved model.

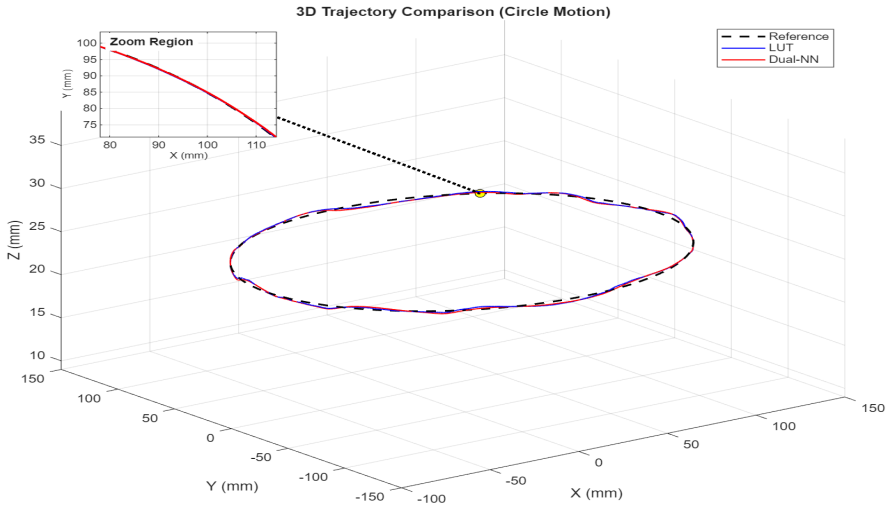


Figure 5.1: Experimental tracking results for the circle path, comparing the original model against the developed model.

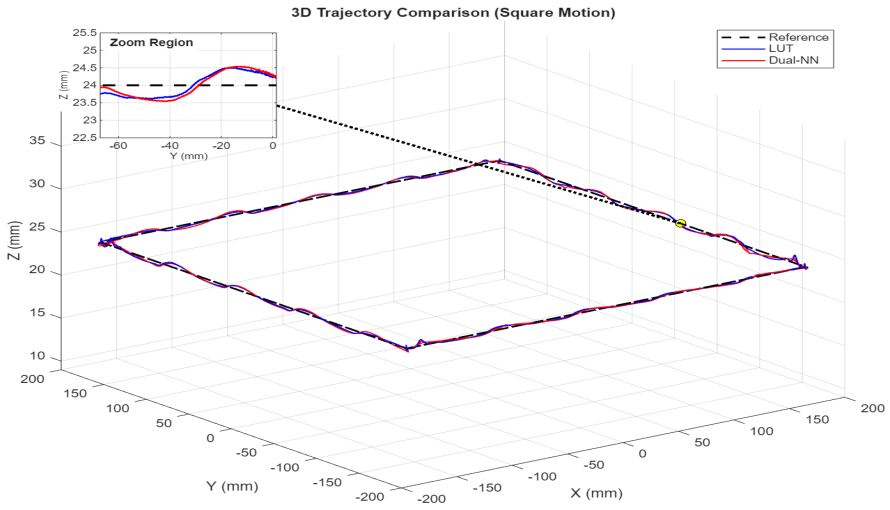


Figure 5.2: Experimental tracking results for the rectangle path, comparing the original model against the developed model.

The same paths, long stroke, and mover parameters are applied to a higher air gap application outside of the LUT construction range, with the intention of analyzing the new dual-NN performance. Figures 5.4 and 5.5 show the global path of the DMM. For both paths, the original LUT model failed to complete the total path, failing to reach the starting pose and only being able to complete $\sim 8\%$ of the rectangular and circular paths, respectively. While it is expected that the wrench values diminish and flatten with higher global Z positions, the wrench value change during motion of the mover is not linear, as approximated by the original implemented interpolation method of the LUT. With the additional controller effort required to reduce the missing pose-range values, the system is unable to correct all accumulated wrench offsets to the real pose, causing levitation failure. In contrast, the dual-NN setup not only has information about the expanded range, but it also intrinsically learns the curvature figure of the wrench values at different $X-Y$ poses of the stator.

To investigate the dual neural network capacities for large-tilt operations during cooperative interaction, an in-place rotation motion was conducted. Figure 5.3 demonstrates that stable and accurate tilting motions can be obtained with the implemented neural network model, expanding the available regions where the robotic manipulator and mover can interact without either of the systems failing due to modeling issues.

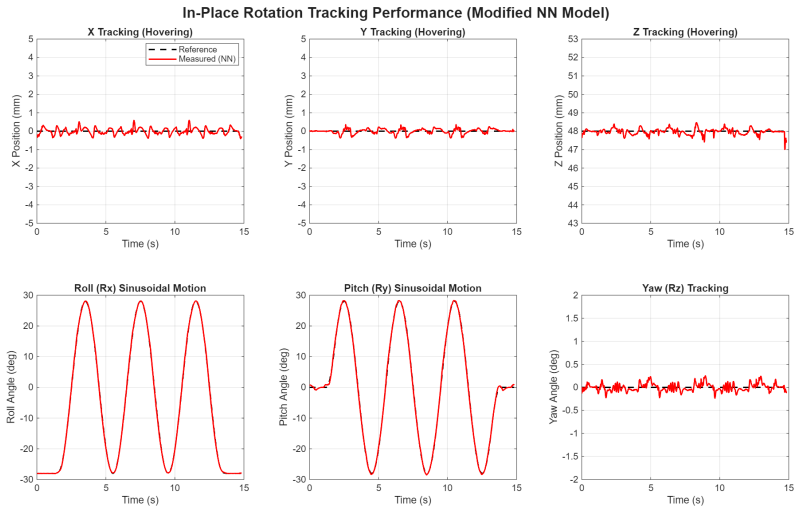


Figure 5.3: Experimental tracking results for an in-place rotation with high tilting for the DMM. The original model is not showcased since it failed to achieve the starting 25° tilting.

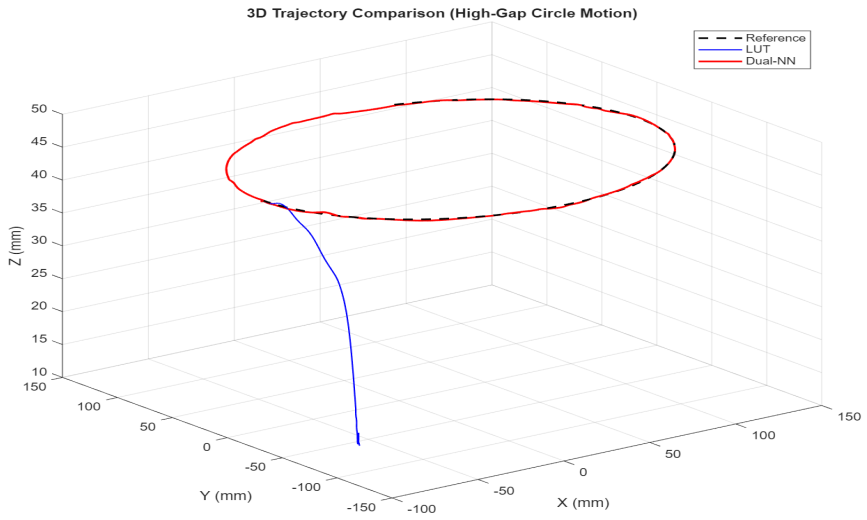


Figure 5.4: Experimental tracking results for the circle path for high Z positions. The original model results exhibit the mover failing to the flat array.

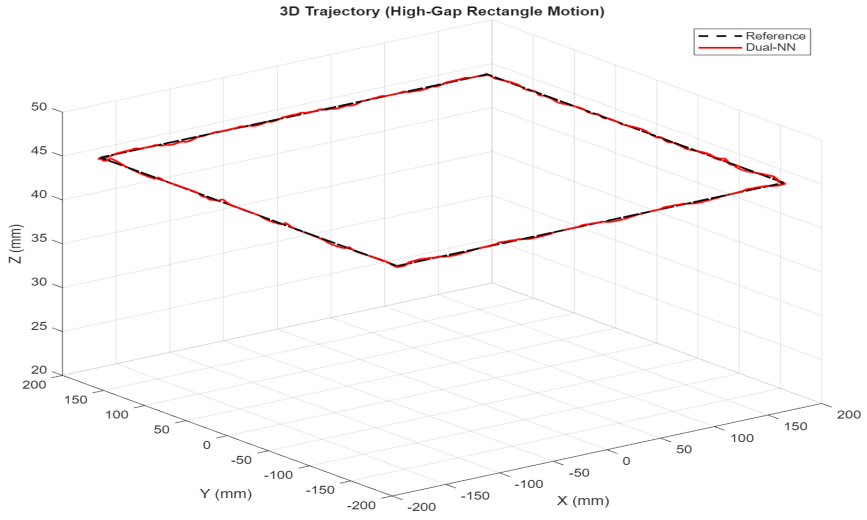


Figure 5.5: Experimental tracking results for the rectangle path, for high Z positions. The original model is not shown since it previously failed to levitate and reach the starting pose.

5.1.2 Tracking Error Metrics

To quantitatively summarize these improvements, the tracking error metrics for all paths and models are presented in Table 5.3. The Root Mean Square Error and Mean Absolute Error are reported for each Cartesian axis (in mm) and each orientation angle (in degrees).

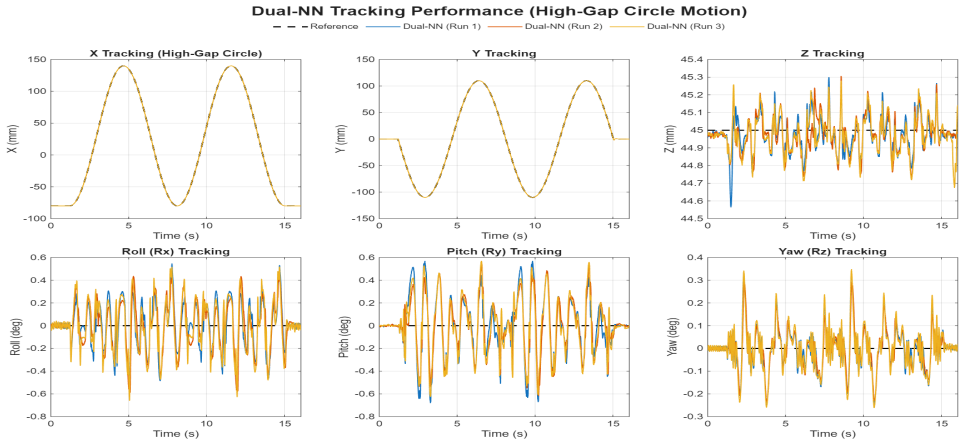
Table 5.3: Tracking Error Metrics Comparison

Path	Model	X-Axis (mm)		Y-Axis (mm)		Z-Axis (mm)		Roll (°)		Pitch (°)		Yaw (°)	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Low Air Gap (Z = 24mm)													
Circle	LUT	0.092	0.066	0.250	0.119	0.200	0.150	0.398	0.275	0.481	0.331	0.128	0.093
Circle	Dual NN	0.078	0.057	0.233	0.128	0.204	0.158	0.247	0.177	0.326	0.225	0.105	0.080
Rectangle	LUT	0.211	0.100	0.220	0.114	0.204	0.154	0.429	0.261	0.335	0.223	0.107	0.079
Rectangle	Dual NN	0.233	0.108	0.249	0.123	0.232	0.181	0.363	0.244	0.300	0.205	0.096	0.070
High Air Gap (Z = 45mm)													
Circle	Dual NN	0.076	0.059	0.177	0.096	0.109	0.085	0.219	0.175	0.248	0.190	0.096	0.069
Rectangle	Dual NN	0.215	0.097	0.220	0.101	0.106	0.082	0.322	0.229	0.177	0.122	0.093	0.066
In-Place Rotation (Z = 45mm)													
Rotation	Dual NN	0.173	0.140	0.133	0.102	0.163	0.118	0.279	0.202	0.413	0.331	0.082	0.064

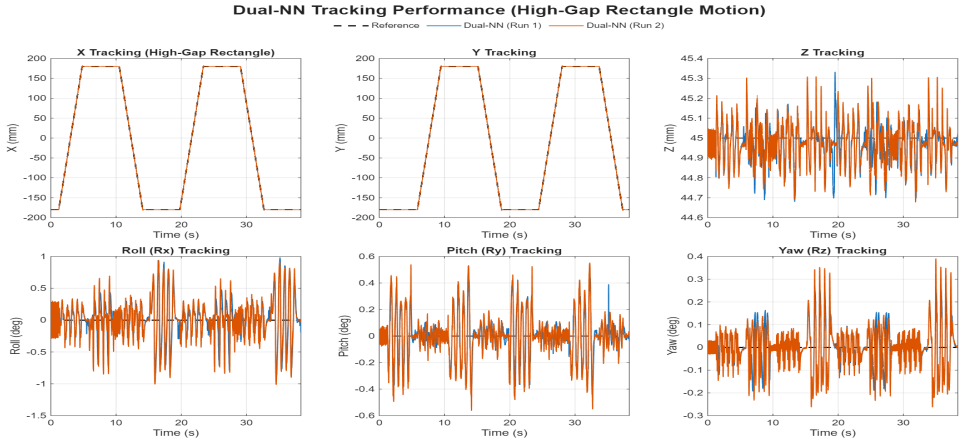
As shown in the tracking metrics, the developed dual neural network model demonstrated an overall improvement in the recorded mover angles over the original LUT method. While a drastic reduction in positional tracking error was not observed across all axes, with even small increased errors in certain paths, it must be noted that the extended operational range of the dual neural network mapping required the system to generalize over a much larger workspace. Consequently, the improved model prioritizes reducing the overall average error across all sampled poses. In contrast, and as explained previously, the newly implemented dual-NN setup was able to obtain long stroke motions during the high air gap zone in comparison to the original LUT model. The results show that the introduced model can compete with the original while also increasing the levitation workspace by ~ 22 times, given the increased tilting and Z axis range.

To demonstrate the repeatability of the newly implemented model, multiple runs were performed on both the rectangular and circular test cases at different air gaps. Figures 5.6a and 5.6b show single-axis pose-tracking results from multiple runs in the high-air-gap case. The standard deviation (σ) per axis was computed by generating a dataset of all instantaneous errors (tracked pose against commanded pose per frame) for every recorded run, including both high- and low-air-gap cases. This approach provides a generalized performance measure rather than a case-specific metric, yielding the following values per translational axis: 0.179, 0.224, and 0.217 mm for the $x, y,$ and z axes, respectively. The

global variance for orientation is as follows: 0.328°, 0.352°, and 0.089° for the roll, pitch, and yaw tracked axes. Although only rectangular and circular paths were tested, the consistency across multiple runs indicates strong repeatability of the magnetic-levitated device on the two trajectories (curves and lines) that make up most of the required motions during industrial automation.



(a) Individual axes tracking for a circular path.



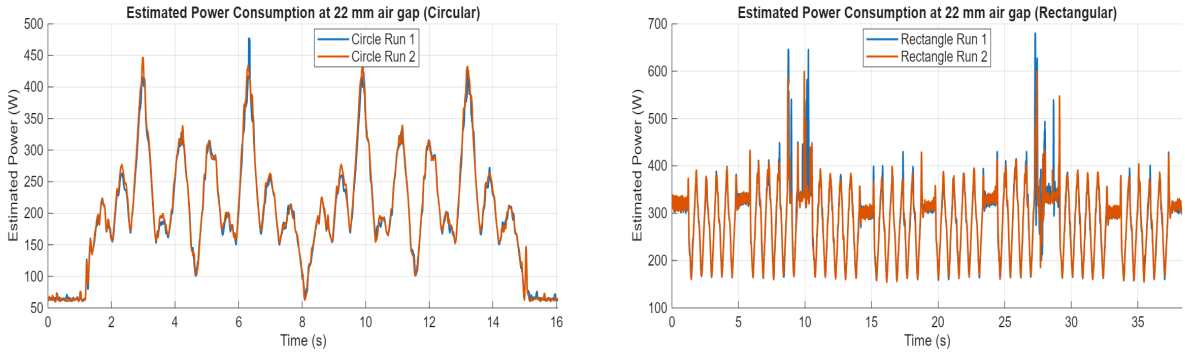
(b) Individual axes tracking for a rectangular path.

Figure 5.6: Multiple recorded circular and rectangular paths during high air gap motion.

5.1.3 Power Consumption Analysis

One of the most important aspects analyzed by engineers in manufacturing is the system's total power efficiency. Since the MLPAs task is the transport and positioning of the workpiece, it is necessary to determine the overall power consumption during levitation. This value not only defines the efficiency of transportation but also determines the energy available to the mover to counteract external loads, such as the contact forces encountered during painting, assembly, or welding. To determine the remaining power available for manipulator interaction with the positioned workpiece, the rectangular and circular long-stroke motions are grouped, and the average mover power consumption is obtained. The parameters used for data collection are the same as in Tables 5.1 and 5.2.

In Figure 5.7, the power consumption during the circular and rectangular paths is shown during an air gap of 22 mm. The studied mover achieved competitive power consumption, with an average of 298 W across both tasks. For the power analysis, two runs of the circular and rectangular trajectories were selected, with each run following the same path twice. The mover's power consumption remains well within the Testbench's power limits, even during consumption spikes, providing a large range of available power for external load compensation.



(a) Power consumption during the circular path.

(b) Power consumption during the rectangular path.

Figure 5.7: Power consumption during motion trajectories.

5.2 Cooperative Path Planning Evaluation

With the system's foundational control validated, the cooperative path planning pipeline was experimentally tested. The following sections detail the interaction cases, the execution methodology, and the performance analysis comparing the proposed 5-DOF cooperative planner against the arm-only baseline constraint algorithm described in Section 4.4.5.

5.2.1 Interaction Cases and Experimental Execution

The evaluations were conducted over three distinct geometric interaction paths to thoroughly examine the system under varying levels of dynamic complexity. For each experimental test, the current arm joint angles and mover pose were used as the initial starting node for both the cooperative and the baseline planners. Each planner was tasked with generating the full sequence of configuration waypoints.

Once the feasible trajectory sequence was found, the trajectories were loaded and executed. A supervisory master-slave framework was initialized to synchronize execution, where the robotic arm acted as the master and the DMM followed as the slave. As mentioned in Section 3.3, a global planner PC ran a path progression controller monitoring the arm's progress, continuously updating the desired mover pose so that both systems reached each waypoint within a prescribed tolerance.

To account for real-world discrepancies, a pre-operation calibration was performed. The arm was moved through various configurations, and the end effector pose was tracked using Vicon markers to generate the required geometric transformations. Consequently, the tracked trajectories represent physically measured motion rather than merely idealized commanded poses. Appendix B details the end-effector calibration setup of the manipulator in previous experiments.

5.2.2 Path 1: Circle Trajectory

The first evaluated interaction path is a circular trajectory, conceptually similar to the circular paths often required in continuous-seam welding or adhesive dispensing. For this evaluation, the specific circular geometry referenced as an example in Figure 4.15 was tracked. The experimental tracking results for a single recorded run for this path are reported in Figure 5.8, comparing the cooperative and baseline executions directly.

Both the arm-only baseline planner and the developed cooperative path planner successfully generated feasible waypoint sequences to complete the circular trajectory. During the planning phase, it was observed that the arm-only baseline planner required considerably less computation time than the proposed cooperative planner. This is expected; when a trajectory is fully reachable by the arm alone, the redundant cooperative system yields a near-infinite number of valid configuration chains for each anchor, forcing the planner to evaluate a vastly expanded search space. Nonetheless, computation time can be reduced by user preference during the planner call via an additional input that limits exploration of additional branches by a limiting factor, similar to the well-known beam-search greedy modification applied to the breadth-first search algorithm. [18].

During execution, however, the cooperative planner demonstrated distinct interaction advantages. The arm-only baseline completed the trajectory in 82.61 seconds, with maximum positional tracking errors of $(0.87, 0.86, 0.11)$ mm and RMSE values of $(0.31, 0.32, 0.03)$ mm in the order of axis $X, Y,$ and Z, respectively. In contrast, the cooperative strategy completed the same task significantly faster, in 73.87 seconds, while simultaneously reducing the positional MAE to $(0.73, 0.83, 0.07)$ mm and RMSE values to $(0.27, 0.30, 0.02)$ mm. Both methods achieved identical orientation accuracy, maintaining a maximum pitch error of $0.04°$ and a pitch RMSE of $0.01°$. It is important to note that the reduction in the error metrics does not derive from the usage of the cooperative algorithm, but is a side effect of the single-arm planned path generating unfavorable combined poses where tracking occlusion from the Vicon system could occur.

Interestingly, despite the faster completion time, the cooperative planner did not generate a trajectory that assigned the mover a higher yaw motion to accomplish the circular path, as one might expect. This is because the cooperative planner inherently optimizes for time efficiency; it naturally selects final orientation poses that closely align with the starting pose, effectively eliminating the need for large yaw realignments during the homing phase. In addition to the need for angle realignment to the starting pose after interaction, the planner also takes into consideration the kinematics limits of both systems during the graph search, which also influences the motion requirements of the mover since it calculates a faster motion by the arm only, generating a path mostly dominated by the manipulator using the cooperative path planner.

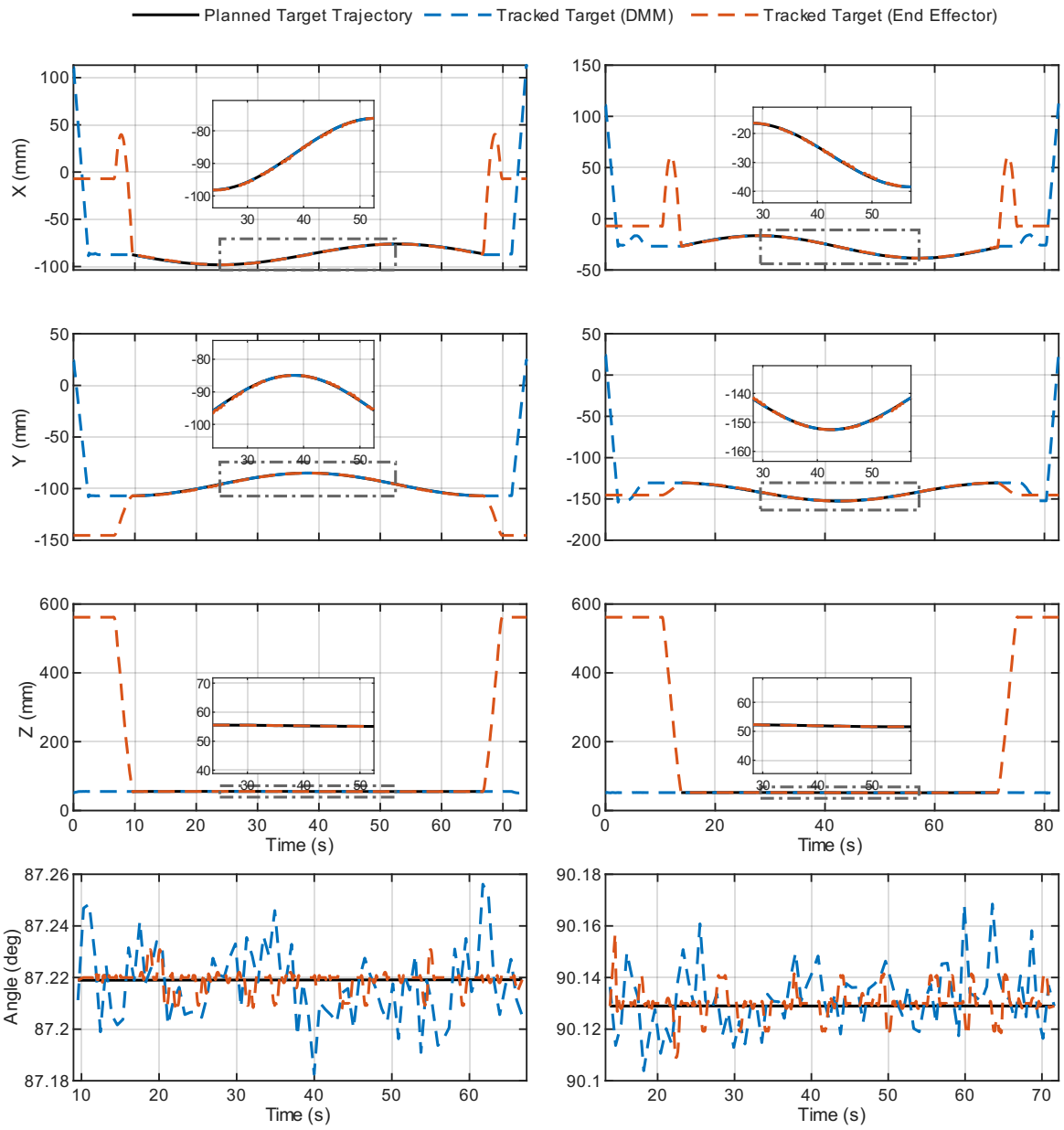


Figure 5.8: Experimental tracking results for the circle trajectory (Path 1). Left column: Cooperative execution. Right column: Arm-only baseline.

5.2.3 Path 2: Line Trajectory

The second evaluated interaction path is a conventional straight-line trajectory, characteristic of traditional assembling and welding production lines. The proposed path is the same as shown in Figure 4.12, showcased in the previous section. The experimental tracking results for the line path are reported in Figure 5.9, comparing the cooperative and baseline execution directly for a single run. In the arm-only baseline case, the trajectory was completed in 87.85 seconds. The maximum tracking errors recorded were $(0.65, 0.08, 0.06)$ mm for the positional axis, with RMSE values of $(0.25, 0.04, 0.02)$ mm. The orientation control yielded a maximum pitch error of $0.07°$ and a pitch RMSE of $0.02°$.

Under the proposed cooperative strategy, the same task path was completed in only 71.21 seconds. The cooperative tracking demonstrated maximum errors of $(0.07, 1.02, 0.06)$ mm and RMSE values of $(0.03, 0.20, 0.02)$ mm. The maximum pitch error was significantly reduced to $0.03°$, with a pitch RMSE of $0.01°$.

While there is no correlation between the two planned trajectories, for this specific line path on the target, a clear degradation in tracking accuracy on one of the positional axes (X or Y error) occurs during the two configurations. Since the same error is shared during both interactions, it can be attributed to the tracked path requiring a mover-arm tilted interaction on a single plane with constant occlusion of the Vicon trackers. The reduction in completion time continues to prove the anticipated benefits of active cooperation. By not remaining fixed at a static anchor orientation, the mover dynamically assists the arm by closing the spatial gaps between waypoints. Furthermore, the baseline algorithm was forced to select a suboptimal anchor that required significant static tilting to comply with the waypoints' attack-angle constraint, thereby significantly increasing the approach and home-return times (t_s and t_h). This is an important effect from trying to complete a 5-DOF task without mover motion, since, from all the possible anchors being checked, only a $\sim 5\%$ of them will be able to fully accomplish the proposed task, with a high probability of them requiring very large rotation angles.

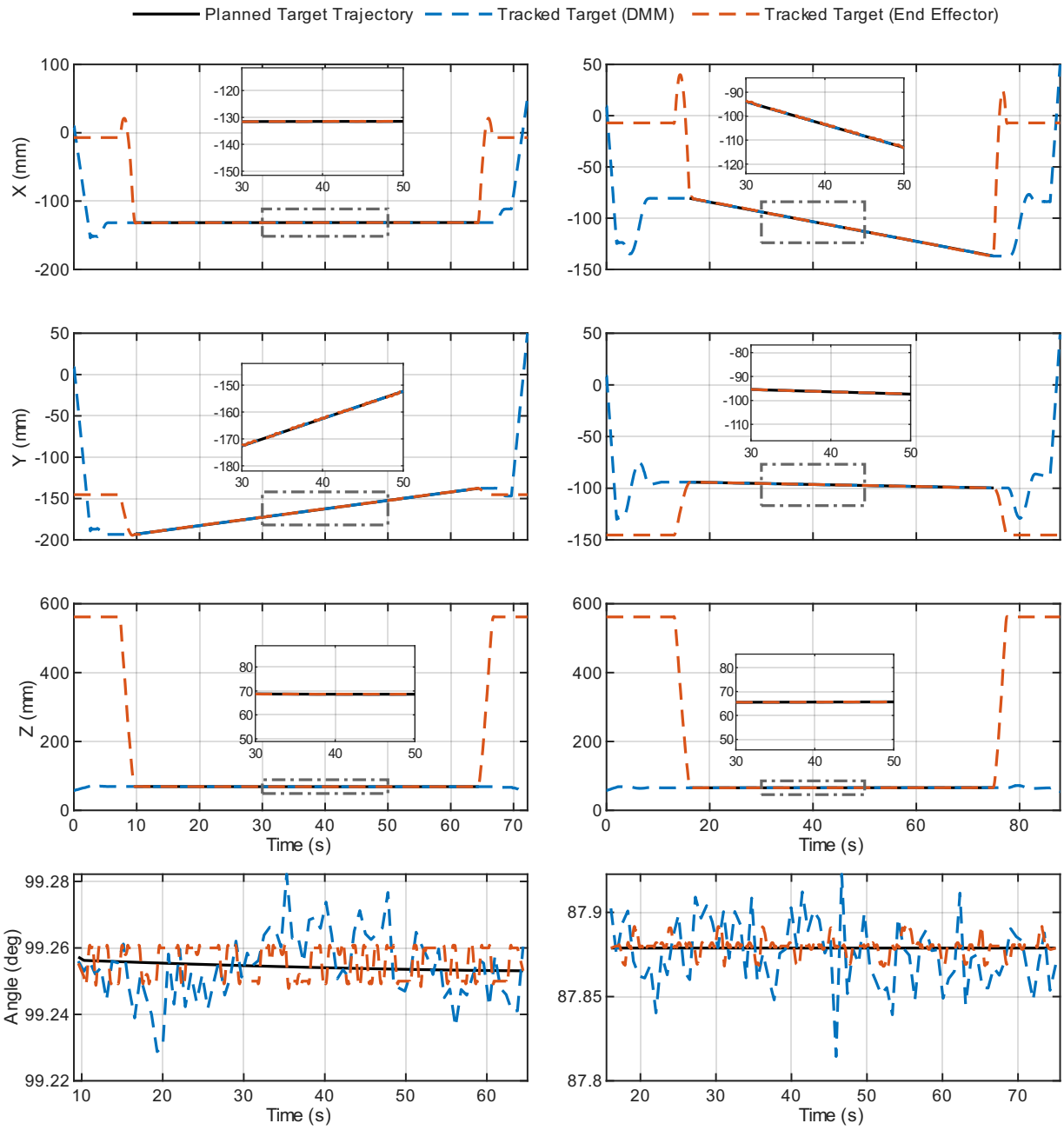


Figure 5.9: Experimental tracking results for the line trajectory (Path 2). Left column: Cooperative execution. Right column: Arm-only baseline. Shows global coordinates, expanded interaction phases, and attack angle tracking.

5.2.4 Path 3: Arc Trajectory

The final evaluation utilized a continuously changing orientation arc path. This curve demands simultaneous cooperation between the arm and mover due to the aggressive shifting required in the end effector's Z-axis attack angle. Figure 5.10 illustrates the target and path positional and orientation requirements, which change depending on the path distance.

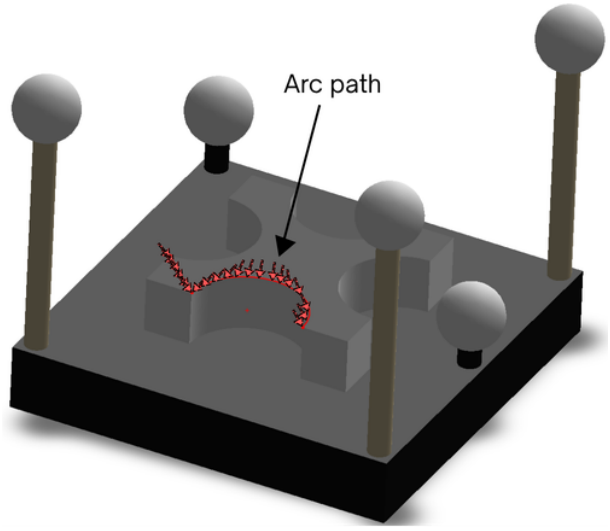


Figure 5.10: Proposed curve path for cooperative interaction.

For this complex curve, only the cooperative planner was successful in finding a feasible sequence. The constrained arm-only planner failed to find any fixed anchor that could satisfy the required end-effector position and orientation for all of the waypoints over the entire arc. The trajectory tracked by the cooperative algorithm is shown in Figure 5.11. The complete path was finished in 127 seconds, yielding maximum tracking errors of $(1.01, 1.62, 1.12)$ mm and RMSE values of $(0.32, 0.52, 0.39)$ mm. The maximum pitch error recorded was $0.76°$, with an RMSE of $0.32°$. While tracking errors marginally increased compared to the simple line path, primarily due to dynamic tilting and intermittent optical occlusions of the Vicon markers, this experiment conclusively demonstrates the critical necessity of the cooperative framework for enabling lower-DOF manipulators to complete otherwise infeasible tool paths.

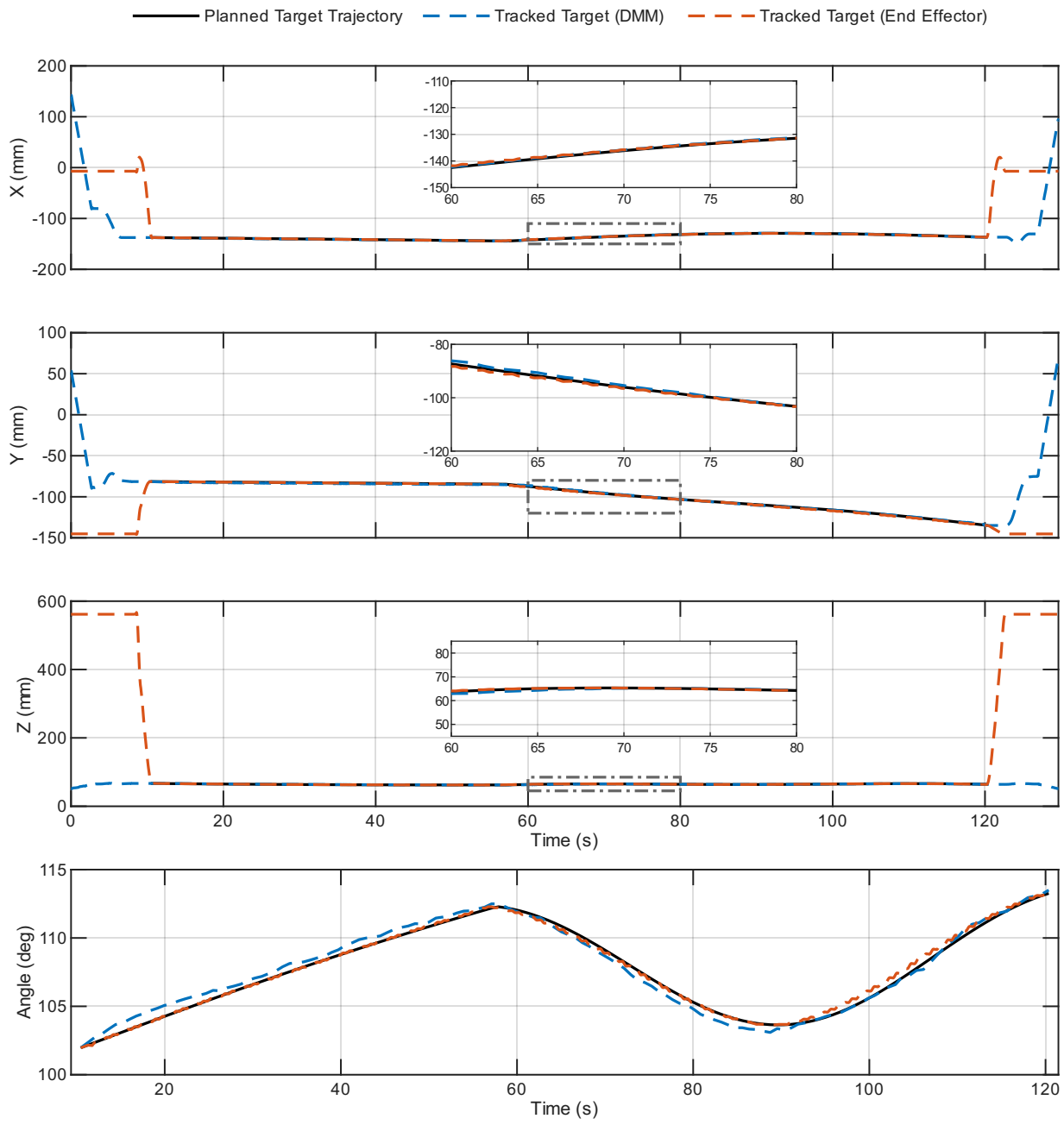


Figure 5.11: Experimental tracking results for the arc trajectory (Path 3) utilizing the cooperative planner. The arm-only baseline failed to find a valid solution for this path.

5.2.5 Individual Tracking Performance

As previously done with the experimental results from the comparison between the dual neural network and look-up table modeling, multiple algorithm runs were performed to calculate the repeatability of each individual system. One important factor to consider is that the path planner algorithm, being naturally sample-based, can produce different kinematic sequences for both the manipulator and the mover, which in turn produce different outcomes and task times. Nonetheless, if the same dataset of sampled manipulator joints is used, the path planner will always reach the same kinematic chain because the algorithm's initial seed will be the same (the mover anchor's pose). As such, the following analysis aims to represent the accuracy achieved across multiple planned paths for each individual device (mover and manipulator) to identify the robustness of each local controller during cooperative interaction.

By providing and validating the same proposed paths from the previous section, three successful runs of each path for the cooperative path planner are recorded and included in the statistical analysis, using a procedure similar to that in Section 5.1.2. Table 5.4 presents the accumulated metrics from a total of 9 runs. While the overall motion is completed with the precision found from the mover analysis alone, it is clear that during cooperative motions the mover tracking performance is affected for short periods, producing the observed peaks of maximum errors.

Table 5.4: Cooperative Execution Tracking Accuracy (Aggregated Runs Dataset)

Mover Accuracy						
Metric	**X (mm)**	**Y (mm)**	**Z (mm)**	R_x **(deg)**	R_y **(deg)**	R_z **(deg)**
RMSE	0.0375	0.0345	0.0446	0.1620	0.0987	0.0761
Max error	0.7419	0.5654	0.8644	1.2573	1.3983	1.4493

Arm/Manipulator Accuracy				
Metric	**Joint 1 (deg)**	**Joint 2 (deg)**	**Joint 3 (deg)**	**Joint 4 (deg)**
RMSE	0.1020	0.0707	0.2421	0.3642
Max error	0.9630	0.4234	1.0773	1.7162

Chapter 6

Conclusions

This thesis presented a comprehensive framework for the integration, control, and cooperative path planning of a magnetic levitation planar actuator operating alongside a conventional 4-DOF robotic manipulator. To accomplish this, the project was divided into two main objectives: expanding the DMM levitation workspace due to the new cooperative air-gap requirements and constructing a path planner capable of generating trajectories for both the mover and the manipulator.

In this project, the enhancement over the initial control cycle model was implemented in the form of a calibration map by exploiting the availability of the Vicon markers for correct physical coil localization and the integration of a Kalman Filter into the model. It was found that adding the Kalman Filter provided realistic system estimation, eliminating the original implementation's derivative spikes due to random tracking pose desynchronization. The enhancements were validated experimentally in conjunction with a novel dual neural network setup, designed to expand the range of a restricted look-up table without increasing the required system memory and parameter loading time. The proposed model in this project obtained a real-time execution of 398 μs, which complies with the system timing constraint of 1000 μs, while obtaining similar tracking performance to the look-up table on the low air gap boundary and significantly reducing the mover errors during high air gap motions, with analyzed cases where only the developed neural network configuration could achieve levitation.

A new path planner algorithm was proposed for the cooperative interaction between the mover and the manipulator. The path planner was designed to validate and utilize a discretized sampled manipulator workspace for the generation of possible mover anchors. Then, the constructed planner systematically explored the null-space of the system to

localize possible arm-mover configurations for a given task trajectory. The planner was validated with a real magnetically levitated device for three different paths. It was experimentally verified that only the cooperative planner can achieve certain trajectories, while providing faster completion times in comparison to having the arm single-handedly complete two of the proposed paths. This required designing and integrating a new communication pipeline between the MLPA, a master PC with the path-planning logic, and the robotic arm.

With the integrations provided in this project, the initial steps toward intelligent cooperation between external and magnetically levitated devices were taken. The potential of magnetic levitation technology for manufacturing was showcased by achieving task trajectories that a conventional production line could not achieve without increased difficulty due to positioning, space, and available manipulators.

6.1 Future Works

While this research successfully bridged the operational gap between an MLPA and an external manipulator, the proposed cooperation algorithm opens several possibilities for future improvements across diverse sections:

1. **Enhancement of robustness of the null-space exploration:** In the proposed path planner, the original seed used for the solver as a starting guess is the calculated average pose of the interaction track, given by the anchored position of the mover and target relationship. When a successful anchor first waypoint is found, subsequent layers of nodes are populated by perturbing the null-space of the solution of the previous layer. This is done to find solutions from a family of configurations where the mover is at the center of the rotation range of the interaction path, and the subsequent calculated layers are close to the solved previous configurations, preventing sudden changes of joint state. While practical and proven, the solver does not explore possible configurations far away from the initial family of configurations. Sample-based or seed-restart techniques could be explored to enhance the found optimal solutions.

2. **Energy-Aware Path Optimization:** Currently, the only variable optimized in the local step of the cooperative path planner is task completion time. While time efficiency inherently correlates with total energy consumption, explicitly incorporating an energy optimization parameter into the cost function could yield significantly more efficient interactions, particularly for highly repetitive industrial tasks.

3. **Advanced Interaction Control Architecture:** Although not the primary focus of this thesis, replacing the current unidirectional master-slave observer controller would greatly improve the interaction's accuracy. An advanced bidirectional control architecture would allow the mover to apply small, dynamic positional displacements from its anchor pose to actively correct any mechanical deviations or vibrations originating from the manipulator.

4. **Mover as a Force-Sensing Device:** Beyond its primary role as a purely positional actuation device, the MLPA mover can theoretically be utilized as an additional force-feedback sensor. By analyzing the displacements and control effort required to maintain a pose, the system can measure the external forces applied by the robotic arm's tool. This supplementary force feedback could be used to increase the overall precision and performance of contact-based tasks across the entire interaction sequence.

References

[1] F Bancel. Magnetic nodes. *Journal of Physics D: Applied Physics*, 32(17):2155, sep 1999.

[2] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, New York, NY, 2001.

[3] Beckhoff Automation GmbH & Co. KG. Xplanar planar motor system. https://www.beckhoff.com/en-en/products/motion/xplanar-planar-motor-system/, 2026. Accessed: 2026-06-10.

[4] Peter Berkelman and Steven Kang. Multiple magnet independent levitation and motion control using a single coil array. pages 537–542, 2023.

[5] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[6] Alan C. Bovik. Chapter 3 - basic gray level image processing. In Al Bovik, editor, *The Essential Guide to Image Processing*, pages 43–68. Academic Press, Boston, 2009.

[7] B&R Industrial Automation GmbH. Acopos 6d – software-driven planar motor transport system. https://www.br-automation.com/en/products/mechatronic-systems/acopos-6d/, 2026. Accessed: 2026-06-10.

[8] Sebastian Burhenne, Dirk Jacob, and Gregor P. Henze. Sampling based on sobol0 sequences for monte carlo techniques applied to building simulations. In *Proceedings of Building Simulation 2011*, 2011.

[9] Samuel R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse, and damped least squares methods. https://mathweb.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf, 2009. Unpublished survey, accessed 2026-06-15.

[10] Kenneth R. Castleman. Chapter four - geometric transformations. In Fatima A. Merchant and Kenneth R. Castleman, editors, *Microscope Image Processing (Second Edition)*, pages 47–54. Academic Press, second edition edition, 2023.

[11] Yanlin Chen, Xianmin Zhang, Yanjiang Huang, Yanbin Wu, and Jun Ota. Kinematics optimization of a novel 7-dof redundant manipulator. *Robotics and Autonomous Systems*, 163:104377, 2023.

[12] Commonplace Robotics GmbH. Cri ethernet interface. Accessed: June 25, 2026.

[13] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2005.

[14] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22(2):215–221, 06 2021.

[15] Rosen Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2010.

[16] Tilo Espenhahn and Ruben Hühne. Levitation characteristics of a fully superconducting magnetic bearing. *IEEE Transactions on Applied Superconductivity*, 36(5):1–5, 2026.

[17] Fabrizio Flacco, Alessandro De Luca, and Oussama Khatib. Prioritized multi-task motion control of redundant robots under hard joint constraints. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3970–3977, 2012.

[18] David Furcy and Sven Koenig. Limited discrepancy beam search. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 125–131, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[19] Edward P. Furlani. *Permanent Magnet and Electromechanical Devices: Materials, Analysis, and Applications*. Academic Press, 2001.

[20] Sushant S. Garud, Iftekhar A. Karimi, George P. E. Brownbridge, and Markus Kraft. Evaluating smart sampling for constructing multidimensional surrogate models. *Computers & Chemical Engineering*, 109:274–292, 2018.

[21] John Gilbertson. Introduction to the kalman filter and tuning. *arXiv preprint arXiv:1503.04313*, 2015.

[22] Mikell P. Groover. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Pearson, 2020.

[23] Yisheng Guan and Kazuhito Yokoi. Reachable space generation of a humanoid robot using the monte carlo method. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1984–1989, 2006.

[24] Lounell B. Gueta, Ryosuke Chiba, Tamio Arai, Tsuyoshi Ueyama, and Jun Ota. Practical point-to-point multiple-goal task realization in a robot arm with a rotating table. *Advanced Robotics*, 25(6-7):717–738, 2011.

[25] Igorovsyanko. Industry metal stainless steel, 2017. Accessed: June 10, 2026. Pixabay License.

[26] igus GmbH. robolink® d articulated robot arm (model: Rl-d-rbt-3322-bc). Hardware Component, 2018. 4 DOF standard robotic arm with motor encoders and INI-switches for zero-point definition.

[27] igus GmbH. *igus® Robot Control (iRC) User Guide (Version 14)*. igus GmbH, 2022. Accessed: June 25, 2026.

[28] J. W. Jansen, C. M. M. van Lierop, Elena A. Lomonova, and AndrÉ J. A. Vandenput. Magnetically levitated planar actuator with moving magnets. *IEEE Transactions on Industry Applications*, 44(4):1108–1115, 2008.

[29] Won jong Kim and David L. Trumper. High-precision magnetic levitation stage for photolithography. *Precision Engineering*, 22(2):66–77, 1998.

[30] Afarin Khabbazian, Lauren Kwong, Aaron Lewis, Erica Liu, Noura Abdelrazec, Anna C. Bakenecker, Nil Fontanals, Guillem Lopez, Samuel Sánchez, Juan Manuel Lopez, Brian Carrillo, Monica Farcas, Chris Kallweit, Alfred C. H. Yu, Mir Behrad Khamesee, and Veronika Magdanz. Kidney stone dissolution by tetherless, enzyme-loaded, soft magnetic miniature robots. *Advanced Healthcare Materials*, 14(23):2403423, 2025.

[31] Won-Jong Kim, D.L. Trumper, and J.H. Lang. Modeling and vector control of planar magnetic levitator. *IEEE Transactions on Industry Applications*, 34(6):1254–1262, 1998.

[32] Lukas Kramer, Teun van den Dool, and Gert Witvoet. Demonstrator for nano-precision multi-agent maglev positioning platform for high throughput metrology. *IFAC-PapersOnLine*, 52(15):471–476, 2019. 8th IFAC Symposium on Mechatronic Systems Mechatronics.

[33] Parichit Kumar, Mazyar Ansari, Ehsan Toyserkani, and Mir Behrad Khamesee. Experimental implementation of a magnetic levitation system for laser-directed energy deposition via powder feeding additive manufacturing applications. *Actuators*, 12(6), 2023.

[34] Parichit Kumar and Mir Behrad Khamesee. Development and analysis of a novel magnetic levitation system with a feedback controller for additive manufacturing applications. *Actuators*, 11(12), 2022.

[35] Steffan Lloyd, Rishad A. Irani, and Mojtaba Ahmadi. Fast and robust inverse kinematics of serial robots using halley's method. *IEEE Transactions on Robotics*, 38(5):2768–2780, 2022.

[36] Xiaodong Lu and Irfan ur-rab Usman. 6d direct-drive technology for planar motion stages. *CIRP Annals*, 61(1):359–362, 2012.

[37] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, Cambridge, United Kingdom, 2017.

[38] Maglev Microrobotics Laboratory. Welcome to Maglev Microrobotics Laboratory. https://uwaterloo.ca/maglev-microrobotics-laboratory/, 2026. University of Waterloo; Accessed: 2026-06-17.

[39] Kashif Mahmood, Tauno Otto, and Arnab Chakraborty. Layout planning and analysis of a flexible manufacturing system based on 3d simulation and virtual reality. *Procedia CIRP*, 120:201–206, 2023. 56th CIRP International Conference on Manufacturing Systems 2023.

[40] Vicky Manthou and Maro Vlachopoulou. Agile manufacturing strategic options. In *Agile Manufacturing: The 21st Century Competitive Strategy*, pages 685–702. Elsevier Science Ltd, Oxford, 2001.

[41] F. Landis Markley, Yang Cheng, John L. Crassidis, and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007.

[42] Koichi Matsuda and Jie Luo. Design and control of compact 6-dof maglev linear motor. In *7th International Conference on Electrical, Control and Instrumentation Engineering*, pages 52–57, 2025.

[43] Joseph Nofech and Mir Behrad Khamesee. Machine learning for automation of 3-dof control of magnetically-levitated microrobots. *Mechatronics*, 110:103356, 2025.

[44] Rajesh Patel, Amit Kumar, and Vikas Sharma. A comprehensive review on mechanical conveyor systems: Performance, energy efficiency, and design considerations. *International Journal of Mechanical and Production Engineering Research*, 11(2):45–58, 2023.

[45] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955.

[46] Pexels Contributor. A robotic device. https://www.pexels.com/photo/a-robotic-device-8438977/, 2021. Accessed: 2026-06-10.

[47] Planar Motor Incorporated. Planar motor, 2026. Accessed: 2026-08-07.

[48] Arthur Quaid and Ralph Hollis. Cooperative 2-dof robots for precision assembly. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2188–2193. IEEE, 1996.

[49] R. Ravaud, G. Lemarquand, S. Babic, V. Lemarquand, and C. Akyel. Cylindrical magnets and coils: Fields, forces, and inductances. *IEEE Transactions on Magnetics*, 46(9):3585–3590, 2010.

[50] Razor Robotics. UNIMATE PUMA 500 Robot Arm. https://www.flickr.com/photos/razorrobotics/6202074072, October 2011. Fotografía compartida en Flickr bajo licencia Creative Commons. Tomada originalmente en agosto de 2010. Accedido el 12 de junio de 2026.

[51] Mulukuta S. Sarma and Akira Yamamura. Nonlinear analysis of magnetic bearings for space technology. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15(1):134–140, 1979.

[52] Dan Simon. *The discrete-time Kalman filter*, chapter 5, pages 121–148. John Wiley Sons, Ltd, 2006.

[53] C. Stewart. Implementation of a modular magnetic levitation planar motor. Master's thesis, University of Waterloo, 2024.

[54] Tongxin Sun. A unified control framework for magnetic manipulation: Integrating a robotic arm and an electromagnetic system with simulation verification. *Magnetic Medicine*, 1(4):100053, 2025.

[55] Chanuphon Trakarnchaiyo. *Design and Implementation of Magnetic Levitation Based Modular Omnidirectional Conveyor System*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, aug 2023. Accessed: 2026-06-17.

[56] Chanuphon Trakarnchaiyo and Mir Behrad Khamesee. A novel wrench–current decoupling strategy to extend the use of small lookup data for a long-range maglev planar motor. *Actuators*, 12(9), 2023.

[57] Chanuphon Trakarnchaiyo, Yang Wang, and Mir Behrad Khamesee. Design of a compact planar magnetic levitation system with wrench–current decoupling enhancement. *Applied Sciences*, 13(4), 2023.

[58] Vicon Motion Systems Ltd. *Vicon Tracker User Guide*. Oxford Metrics plc, Oxford, UK, 2020. Technical manual for real-time 6DoF object tracking.

[59] Yang Wang and Mir Behrad Khamesee. Deep learning-based wrench model for magnetically levitated actuators. *IEEE Transactions on Industrial Electronics*, 71(11):14663–14672, 2024.

[60] Yang Wang and Mir Behrad Khamesee. Magfloor: a universal magnetic levitation platform for flexible manufacturing. In *2024 7th International Conference on Mechatronics, Robotics and Automation (ICMRA)*, pages 152–156, 2024.

[61] Yang Wang, Zhenchuan Xu, and Mir Behrad Khamesee. Magnet-coil role-switching real-time wrench model for magnetic levitated motors with extendable motion range. *IEEE Transactions on Industrial Electronics*, 72(3):2863–2871, 2025.

[62] Greg Welch and Gary Bishop. An introduction to the Kalman filter. Technical Report TR 95-041, University of North Carolina at Chapel Hill, Chapel Hill, NC, 2006. Updated version, originally published 1995.

[63] WZ Magnetics. What is Halbach Array? https://www.wzmagnetics.com/what-is-halbach-array/, 2026. Accessed: 2026-06-17; Image showing Halbach array configuration.

[64] Hongjun Xing, Ruixiang Huang, Yuqi Yang, Jinbao Chen, Weihua Li, Chen Yao, Chuang Shi, and Liang Ding. Inverse kinematics and redundancy resolution in manipulators: Methods, optimization objectives, and future trends. *Biomimetic Intelligence and Robotics*, 6(2):100339, 2026.

[65] Zhenchuan Xu, Chanuphon Trakarnchaiyo, Curtis Stewart, and Mir Behrad Khamesee. Modular maglev: Design and implementation of a modular magnetic levitation system to levitate a 2d halbach array. *Mechatronics*, 99:103148, 2024.

[66] Zhenchuan Xu, Yang Wang, Jie Wang, and Mir Behrad Khamesee. Predictor-based model predictive control for maglev planar motor with a 2d halbach array mover. *Control Engineering Practice*, 141:105731, 2024.

[67] Zhenchuan Xu, Xiaodong Zhang, and Mir Behrad Khamesee. Real-time data-driven force and torque modeling on a 2-d halbach array by a symmetric coil considering end effect. *IEEE Transactions on Magnetics*, 58(11):1–10, 2022.

[68] Luguang Yan. Development and application of the maglev transportation system. *IEEE Transactions on Applied Superconductivity*, 18(2):92–99, 2008.

[69] Qingfei Zeng, Xuemei Liu, Chengrong Qiu, and Aiping Li. Kinematic characteristics analysis of cooperative welding robot with multiple manipulators. In *4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pages 1–5, 2019.

[70] Qiang Zheng, Mark Wang, Xiaoming Liu, and Yu Chen. Overview of the development of planar motor technology. *Energies*, 17(16):3955, 2024.

[71] Tong Zheng, Xianze Xu, Xing Lu, and Fengqiu Xu. Analysis and modeling for magnetically levitated actuator with end-effects. In *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 667–673, 2020.

[72] Haiyue Zhu, Tat Joo Teo, and Chee Khiang Pang. Design and modeling of a six-degree-of-freedom magnetically levitated positioner using square coils and 1-d halbach arrays. *IEEE Transactions on Industrial Electronics*, 64(1):440–450, 2017.

[73] Gerlof Zuidema, Dave C. J. Krop, and Elena A. Lomonova. Stator magnet analysis for a novel magnetic levitation actuator. *IEEE Transactions on Magnetics*, 59(11):1–4, 2023.

APPENDICES

Appendix A

Dual Neural Network Structure and Wrench Modeling

This appendix provides an overview of the methodologies applied to design, train, and deploy the dual Neural Network (NN) architecture used for real-time magnetic wrench evaluation. The process encompasses dataset generation, network topology selection, real-time symmetry transformations, and boundary extrapolation testing.

A.1 Dataset Generation

As introduced in Section 4.2.2, the training dataset for the neural network must densely cover the 6-DOF interaction workspace without suffering from the curse of dimensionality inherent to uniform grid sampling. To achieve this, a quasi-random Sobol sequence was employed. Sobol sequences ensure a close uniform distribution while preventing the clustering of sample points, thus providing a robust training foundation for the neural network setup.

To prioritize the high magnetic field gradients occurring at closer proximity to the stator, the dataset is split. The lower air gap zone ($16\text{ mm} \leq Z \leq 28\text{ mm}$) contains 50% of the samples, while the remaining 50% spans the upper air gap zone. Furthermore, a collision filter evaluates each generated pose to ensure the mover does not physically intersect the stator plane before calculating the analytical wrench. Figure A.1 illustrates the distribution of the generated dataset across the 6-DOF configuration space using the Sobol sequence.

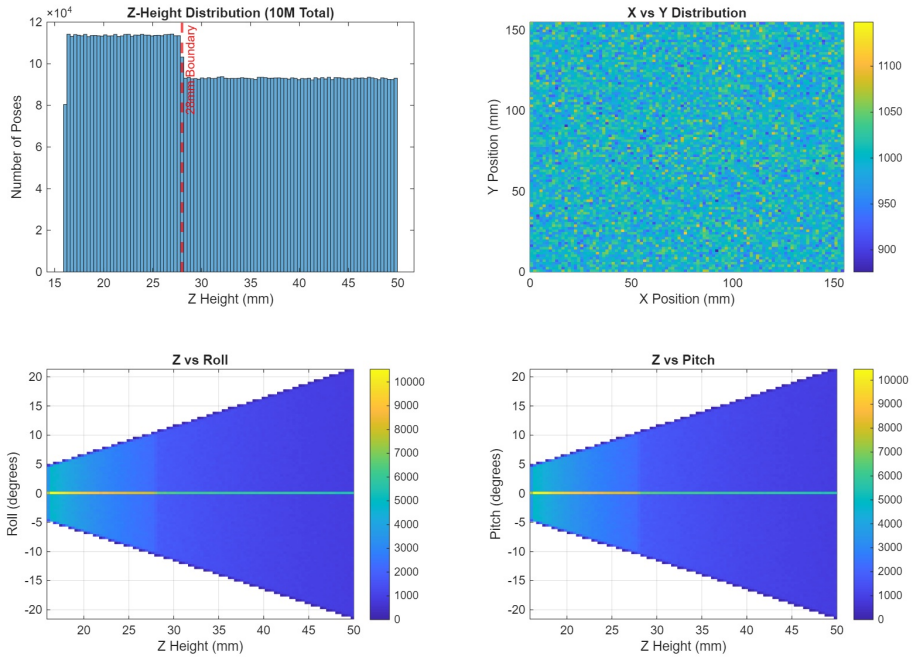
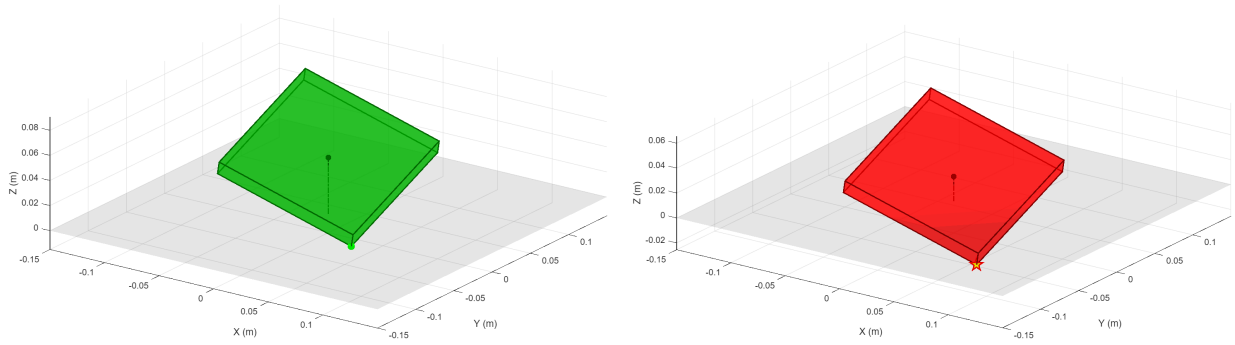


Figure A.1: Distribution of the generated dataset across the 6-DOF configuration space using the Sobol sequence. Note the higher sample rate in the lower air gap region. Also, the collision filter removes tilting poses that produce a collision, generating the funnel pattern on the Z vs tilt plots.



(a) Collision Passed (b) Collision Filtered

Figure A.2: Visual representation of the collision filter boundary conditions. The left image shows a safe tilt angle at a higher air gap (passed), whereas the right image shows a collision at the same tilt angle but a lower air gap (filtered).

A.2 Wrench Quarter Mapping Logic

Instead of calculating the entire $X-Y$ range of action between the electromagnet and the DMM, multiple researchers utilize the intrinsic symmetry of the stator to reduce the overall pose and wrench datasets [59, 60, 57]. This same strategy is implemented to increase the capacity of the trained neural networks to learn the intrinsic wrench changes on a specified area. By applying the symmetry logic shown in Algorithm 4 during the PLC implementation, the wrench modeling system can seamlessly calculate the wrenches for any mover pose.

First, the input to the algorithm is the relative pose of the mover with respect to the center of the active coil. Using this input pose, the quadrant currently occupied by the mover is determined. The corresponding symmetry transformations are then applied to map the pose into the first quadrant, where the neural network evaluates the magnetic wrench. Because the computed wrench corresponds to the first-quadrant pose, a final unwrapping step is required to translate the force and torque vectors back to the mover's original quadrant. Figure A.3 illustrates the wrapping logic used to determine the equivalent first-quadrant pose.

Because the mover arrangement comprises four identical disc permanent magnets, it possesses a 90-degree rotational symmetry. This property allows the yaw angle to be safely wrapped into a $[0, \pi/2]$ domain without loss of physical accuracy, simplifying the neural network's required mapping space even further.

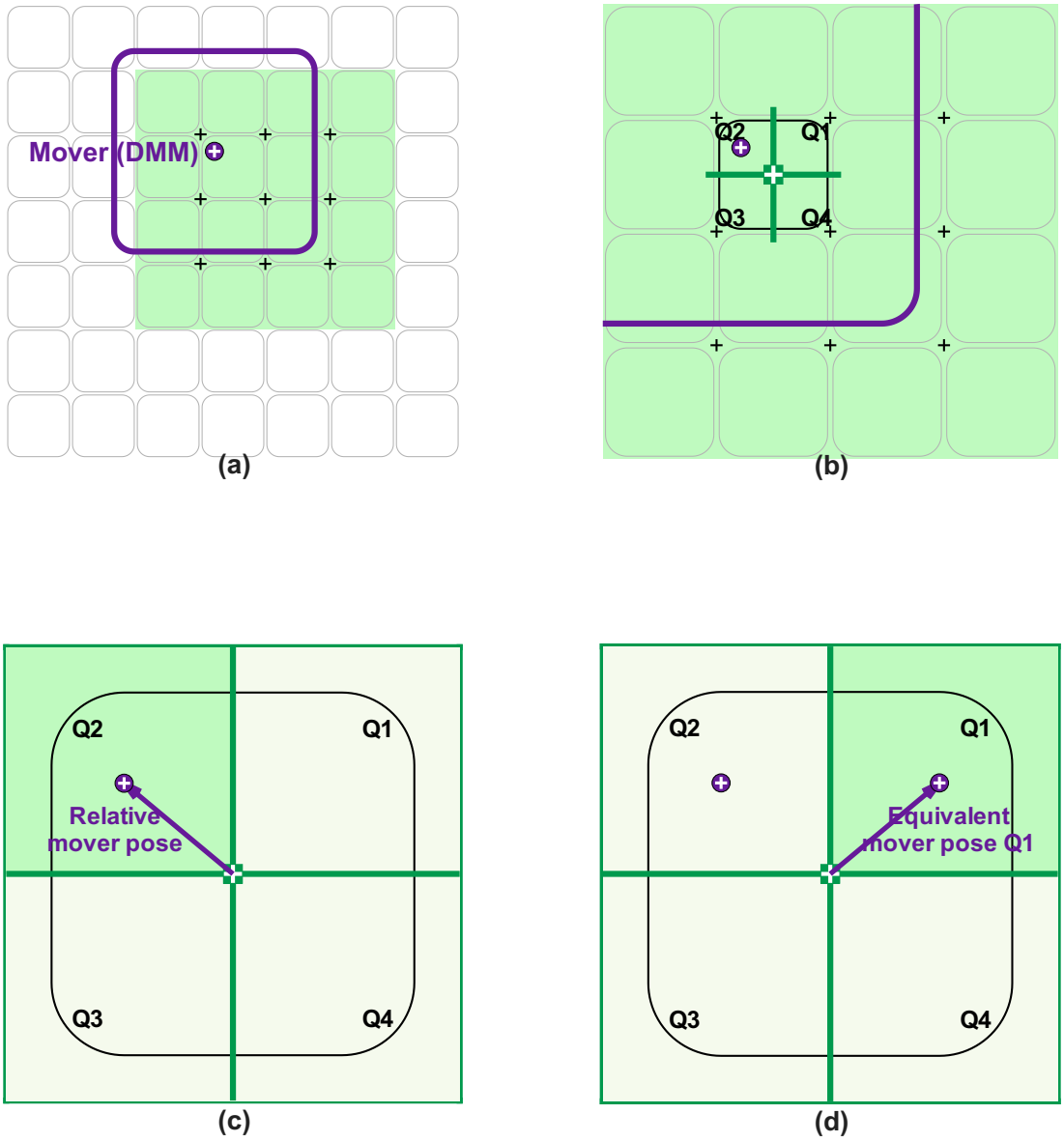


Figure A.3: Mover pose wrapping logic: (a) Find the subset of active stator coils; (b) Iterate through each coil to calculate the relative mover pose; (c) Identify the current mover-coil quadrant; (d) Map to the equivalent mover pose in the first quadrant.

Algorithm 4 Wrench Quarter Mapping and Yaw Symmetry Logic

Require: Input pose $\mathbf{x} = [x, y, z, \phi, \theta, \psi]^\top$
Ensure: Output Wrench $\mathbf{W}_{\text{out}} = [F_x, F_y, F_z, \tau_x, \tau_y, \tau_z]^\top$

$\mathbf{x}_{q1} \leftarrow \mathbf{x}$
if $x < 0$ and $y \geq 0$ **then** ▷ Quadrant 2
    $\mathbf{x}_{q1}(x) \leftarrow -x, \quad \mathbf{x}_{q1}(\theta) \leftarrow -\theta, \quad \mathbf{x}_{q1}(\psi) \leftarrow -\psi$
else if $x < 0$ and $y < 0$ **then** ▷ Quadrant 3
    $\mathbf{x}_{q1}(x) \leftarrow -x, \quad \mathbf{x}_{q1}(y) \leftarrow -y, \quad \mathbf{x}_{q1}(\phi) \leftarrow -\phi, \quad \mathbf{x}_{q1}(\theta) \leftarrow -\theta$
else if $x \geq 0$ and $y < 0$ **then** ▷ Quadrant 4
    $\mathbf{x}_{q1}(y) \leftarrow -y, \quad \mathbf{x}_{q1}(\phi) \leftarrow -\phi, \quad \mathbf{x}_{q1}(\psi) \leftarrow -\psi$
end if
$\mathbf{x}_{q1}(\psi) \leftarrow \mathbf{x}_{q1}(\psi) \bmod (\pi/2)$ ▷ Yaw 90-degree symmetry
$\mathbf{W}_{\text{out}} \leftarrow$ **Dual-NN call** ▷ The Q1 wrench is calculated
if $x < 0$ and $y \geq 0$ **then** ▷ Quadrant 2
    $\mathbf{W}_{\text{out}}(F_x) \leftarrow -F_x, \quad \mathbf{W}_{\text{out}}(\tau_y) \leftarrow -\tau_y, \quad \mathbf{W}_{\text{out}}(\tau_z) \leftarrow -\tau_z$
else if $x < 0$ and $y < 0$ **then** ▷ Quadrant 3
    $\mathbf{W}_{\text{out}}(F_x) \leftarrow -F_x, \quad \mathbf{W}_{\text{out}}(F_y) \leftarrow -F_y, \quad \mathbf{W}_{\text{out}}(\tau_x) \leftarrow -\tau_x, \quad \mathbf{W}_{\text{out}}(\tau_y) \leftarrow -\tau_y$
else if $x \geq 0$ and $y < 0$ **then** ▷ Quadrant 4
    $\mathbf{W}_{\text{out}}(F_y) \leftarrow -F_y, \quad \mathbf{W}_{\text{out}}(\tau_x) \leftarrow -\tau_x, \quad \mathbf{W}_{\text{out}}(\tau_z) \leftarrow -\tau_z$
end if
return $\mathbf{W}_{\text{out}}$

A.3 Neural Network Structure and Topology Selection

To identify the optimal balance between computational execution speed (to comply with the sub-millisecond real-time requirements of the PLC controller) and prediction accuracy, multiple neural network topologies were evaluated. The configurations tested included single large networks and various, different-sized dual network setups using Residual Multi-Layer Perceptrons (ResMLPs) with hyperbolic tangent (`tanh`) and regular linear activation functions.

While the split at the $Z = 28$ mm corresponds to a change in the levitation force Z behavior, it is also applied to circumvent the PLC computational power limitations by specializing the networks on different stages of levitation. This is easily seen in Table A.1 for Model A, which represents a singular deep neural network composed of a three-layer setup and an increased neuron capacity. While similar error metrics were obtained, the calculation time exceeds the 1 ms PLC constraint. Models B through D are structured with the same number of layers, but with different capacities to find the saturation point of learning the dataset. Therefore, they represent layers with a capacity of 16, 48, and 64 neurons, respectively. Model E introduces an extra layer. Finally, Model F is configured with a similar structure as Model C, but a different activation function is used (a rectified linear unit, ReLU).

The metrics selected to evaluate the error against the analytical model include the maximum force and torque discrepancies, the Frobenius norm, and the Root Mean Square Error (RMSE). The Frobenius norm, $\|\mathbf{K}\|_F$, is particularly intuitive as it compares the overall structure of the predicted wrench vector against the analytical one, providing a single scalar value that represents the total divergence between the two.

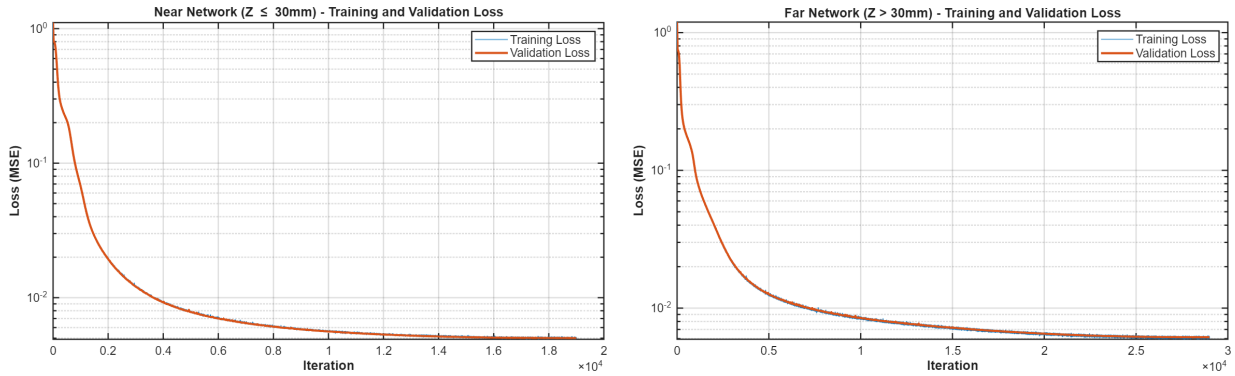
Table A.1: Comparison of Evaluated Neural Network Topologies (Single Coil vs Analytical)

Model	Layers	Neurons	Time (μs)	Max F (N)	Max τ (Nm)	Max $\|K\|_F$	RMSE $F_{x,y,z}$ (10^{-2} N)	RMSE $\tau_{x,y,z}$ (10^{-3} Nm)
A	3	128	1982	1.251	0.059	1.446	1.95, 1.86, 3.10	1.11, 1.22, 0.59
B	3	16	110	1.548	0.066	1.752	4.77, 3.87, 5.82	2.24, 2.19, 1.17
C	3	48	277	1.116	0.051	1.275	2.41, 2.30, 3.82	1.25, 1.40, 0.61
D	3	64	627	1.153	0.053	1.251	2.21, 2.12, 3.37	1.28, 1.28, 0.64
E	4	48	398	1.275	0.053	1.322	1.84, 1.75, 2.74	0.99, 1.07, 0.52
F	3	48	265	1.241	0.052	1.352	2.18, 2.09, 3.34	1.28, 1.55, 0.68

Based on the collected data, Model E represents the optimal balance for the dual-NN implementation. Although it exhibits a roughly 43% increase in execution time compared to Model C, it achieves an almost 30% improvement across the RMSE metrics, presenting

a highly favorable trade-off. Interestingly, Model D—despite possessing more neurons per layer—does not match the performance of Model E. This indicates that the wrench prediction task benefits more from a deeper network architecture rather than simply widening the layers, a characteristic advantage commonly observed in residual network structures.

Furthermore, while even larger networks could theoretically be employed to further reduce the error, the real-time execution of the dual-NN setup within the Simulink control loop must share computational resources with the Kalman Filter, the active coil mapping interpolation, and other essential control components. Therefore, selecting a lightweight structure like Model E maintains a critical margin of computational power on the PLC, ensuring that the system remains flexible and responsive enough to accommodate future algorithmic additions to the control loop.



(a) Near Network Loss

(b) Far Network Loss

Figure A.4: Training and validation loss values for the selected dual-NN setup (Model E). The dual architecture requires independent evaluations for the near and far air gap regions.

A.4 Extrapolation Capabilities and LUT Comparison

One of the primary limitations of the previous rigid Lookup Table (LUT) was its strict spatial boundaries. Whenever the interaction trajectory forced the mover beyond the precalculated LUT grid, the output wrench was artificially capped to the boundary values, drastically degrading control performance.

In contrast, the dual-NN setup possesses intrinsic function approximation capabilities that allow it to safely extrapolate slightly beyond the sampled domain without immediately saturating. To validate this, tests were conducted comparing the wrench estimation of the

NN against the full analytical model for configurations placed outside the valid boundaries of the original LUT.

Table A.2 showcases the quantitative results of evaluating both models within this extrapolated region. As observed, the dual-NN setup achieves almost double the accuracy in terms of the RMSE across both forces and torques when compared to the clamped LUT during numerical analysis. While the maximum absolute force discrepancies are relatively similar between the two methodologies, this behavior is mathematically expected. The magnetic field intensity naturally decays with increased distance (such as air gaps above 30 mm), meaning the absolute force magnitudes are inherently smaller; thus, the clamped LUT values do not diverge as aggressively as they would closer to the stator. Nonetheless, as seen by the experiments conducted and shown in Section 5.1, the original model was unable to achieve the high air gap motions due to spontaneous change of mover control effort required to maintain the air gap.

Table A.2: Extrapolation Performance Comparison Outside LUT Range ($Z \in [30, 60]$ mm, Tilts $\pm 25°$)

Method	**Max F (N)**	**Max τ (Nm)**	**Max $\|K\|_F$**	**RMSE $F_{x,y,z}$ (10^{-1} N)**	**RMSE $\tau_{x,y,z}$ (10^{-3} Nm)**
Per-Coil K-Matrix					
Current LUT	2.670	0.136	3.037	-	-
Dual-NN	2.791	0.104	2.944	-	-
Total 16-Coil Assembly					
Current LUT	2.411	0.118	2.615	2.38, 2.46, 3.61	19.66, 20.58, 9.98
Dual-NN	2.816	0.135	2.569	1.40, 1.45, 2.24	8.39, 8.43, 4.27

Appendix B

Robot Manipulator Vicon Calibration

This appendix details the calibration procedures performed on the 4-DOF robotic manipulator to accurately track its base and end-effector within the Vicon motion capture system. These steps are necessary to obtain the transformation matrix of the end-effector and to account for any forward kinematics (FK) or manufacturing errors resulting from miscalculated manipulator link lengths.

B.1 Robot Arm Base Vicon Object

The first step in integrating the manipulator with the Vicon tracking system is defining the coordinate frame of the robot base. Because the robot arm is designed to be attached to a mobile platform in future iterations of this project, static physical measurements relative to the MLPA testbench are insufficient. Therefore, reflective markers were physically attached directly to the base of the robot arm.

Within the Vicon software, these markers were used to define a rigid body object representing the base of the arm. By creating this object, the system can continuously compute the global transformation matrix of the arm's base at any given time, regardless of its position on the testbench or future mobile platforms. Figure B.1 illustrates the physical placement of the markers and the corresponding object definition within the tracking software, alongside the TwinCAT interface section utilized to record the real-time base pose data.

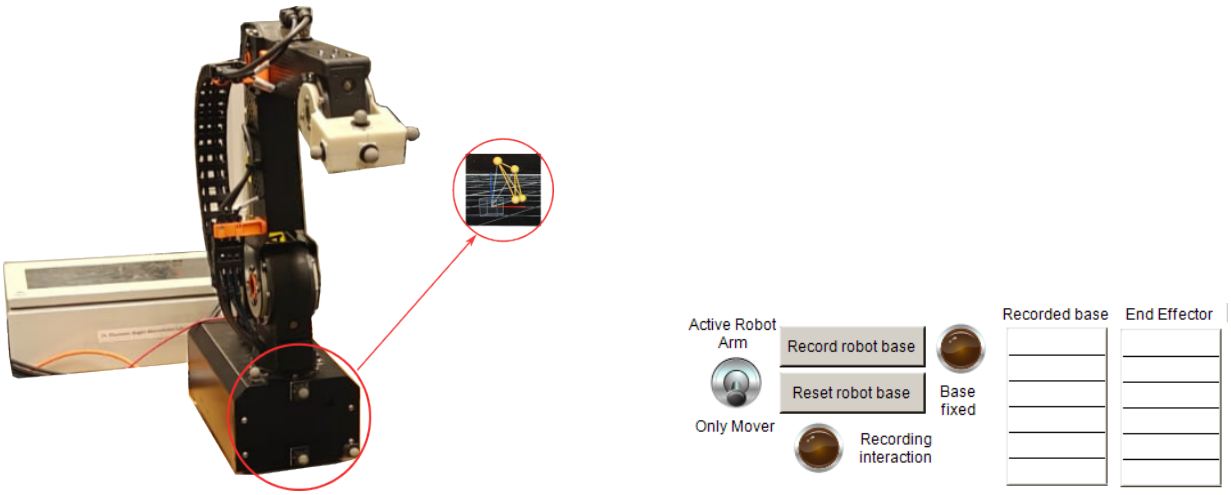


Figure B.1: Reflective markers attached to the robot arm base and the TwinCAT interface utilized for tracking the base pose.

Algorithm 5 Vicon UDP Multi-Object Parsing and Assignment Logic

Require: Received UDP packet msg containing $TotalObjects$
Ensure: Decoded poses for the mover, end-effector, and robot base

for $nObj = 0$ **to** $TotalObjects - 1$ **do**
  $ItemOffset \leftarrow 5 + (nObj \times 75)$ ▷ Each object data packet is 75 bytes
  if $(ItemOffset + 75) \leq 256$ **then**
    $ObjName \leftarrow$ Extract 24 bytes starting at $msg[ItemOffset + 3]$
    $x_{\text{bytes}}, y_{\text{bytes}}, z_{\text{bytes}} \leftarrow msg[\dots]$ ▷ Extract position bytes
    $Rx_{\text{bytes}}, Ry_{\text{bytes}}, Rz_{\text{bytes}} \leftarrow msg[\dots]$ ▷ Extract rotation bytes
    Convert byte arrays to double-precision real values (x, y, z, Rx, Ry, Rz)
    Convert rotations from radians to degrees
    if $ObjName ==$ 'mover' **then**
      Assign values to mover pose variables
    else if $ObjName ==$ 'end_effector' **then**
      Assign values to end-effector pose variables
    else if $ObjName ==$ 'arm_base' **then**
      Assign values to arm base pose variables
    end if
  end if
end for

Initially, the system was configured to track only the mover object stream from the Vicon PC. However, an additional modification was made to the TwinCAT PLC receiver program to continuously stream the poses of both the robot arm base and the end-effector. Leveraging the Vicon UDP protocol structure, up to three object poses can be extracted and collected in every cycle stream package. To ensure that the decoding sequence assigns the correct pose to each object, regardless of the selection order in the Vicon tracker program, the unique name string identifier (ID) of each object is used to map the corresponding pose data. Algorithm 5 illustrates how the poses of the mover, the end-effector, and the robot arm base are simultaneously parsed and assigned.

B.2 End-Effector Forward Kinematics Calibration

With the base transformation T_b continuously tracked via the established Vicon base object, the next step is the end-effector calibration. This process corrects the theoretical Forward Kinematics (FK) calculations to account for real-world deviations resulting from physical attachment offsets. To accomplish this, a separate set of markers was attached directly to the end-effector, creating a distinct object in the Vicon tracker yielding its global pose T_{ee}. As an important assumption, the generated correction homogeneous matrix, ${}^{ee_{FK}}T_{ee}$, is expected to be constant during the entire interaction process. This translates to an unchanging end-effector recording (or small variations) between samples at the same pose.

Since the global base position is reliably handled by direct Vicon tracking, the end-effector calibration focuses entirely on identifying the local transformation offset ${}^{ee_{FK}}T_{ee}$ between the theoretical FK frame and the true physical end-effector frame. The procedure is fully automated. A script commands the manipulator to move through a series of predefined target poses within the camera's Field of View (FOV).

At each configuration, the script stabilizes the arm and performs a robust sampling of the end-effector's global pose T_{ee} alongside the base's global pose T_b. The theoretical local FK pose ${}^{b}T_{ee_{FK}}$ is simultaneously derived from the current joint configuration. Because the base transformation is precisely known, the offset can be isolated algebraically. By rearranging the kinematic chain $T_{ee} = T_b {}^{b}T_{ee_{FK}} {}^{ee_{FK}}T_{ee}$, the offset at each sampled pose is directly extracted as ${}^{ee_{FK}}T_{ee} = (T_b {}^{b}T_{ee_{FK}})^{-1} T_{ee}$. By averaging these computed offsets across all valid target poses using rotational quaternion averaging and translational mean filtering, a highly accurate, rigid end-effector offset matrix is obtained. Algorithm 6 summarizes this direct offset extraction process. This calculated offset is then injected back into the control loop to ensure high-fidelity spatial accuracy during all subsequent interaction tasks.

Algorithm 6 End-Effector Offset Calibration via Direct Base Tracking

Require: Set of N target joint configurations $\mathbf{q}_{\text{targets}}$
Ensure: Rigid End-Effector offset transformation matrix ${}^{ee_{FK}}T_{ee}$

Initialize empty arrays for translations $\mathbf{t}_{\text{offsets}}$ and quaternions $\mathbf{Q}_{\text{offsets}}$
for $i = 1$ **to** N **do**
    Command arm to target configuration $\mathbf{q}_{\text{targets}}[i]$
    Wait for arm stabilization
    Sample Vicon base pose T_b and Vicon EE pose T_{ee} over M frames
    Apply spatial median filter to find representative T_b and T_{ee}
    Compute theoretical FK local pose ${}^{b}T_{ee_{FK}}$ using current joint angles
    ${}^{ee_{FK}}T_{ee}^{(i)} \leftarrow \left(T_b {}^{b}T_{ee_{FK}}\right)^{-1} T_{ee}$
    Extract translation vector $\mathbf{t}^{(i)}$ from ${}^{ee_{FK}}T_{ee}^{(i)}$
    Extract rotation matrix $\mathbf{R}^{(i)}$ from ${}^{ee_{FK}}T_{ee}^{(i)}$ and convert to quaternion $\mathbf{Q}^{(i)}$
    Store $\mathbf{t}^{(i)}$ in $\mathbf{t}_{\text{offsets}}$ and $\mathbf{Q}^{(i)}$ in $\mathbf{Q}_{\text{offsets}}$
end for
$\mathbf{t}_{\text{avg}} \leftarrow \text{mean}(\mathbf{t}_{\text{offsets}})$
$\mathbf{Q}_{\text{avg}} \leftarrow \text{meanrot}(\mathbf{Q}_{\text{offsets}})$ ▷ Average quaternions
Construct final offset matrix ${}^{ee_{FK}}T_{ee}$ using $\mathbf{t}_{\text{avg}}$ and $\mathbf{Q}_{\text{avg}}$
return ${}^{ee_{FK}}T_{ee}$