Automated 3-DoF Control Development for Magnetically-Levitated Microrobots Using Machine Learning

by

Joseph Nofech

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

Waterloo, Ontario, Canada, 2024

© Joseph Nofech 2024

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

This study presents a novel methodology for achieving three-degree-of-freedom (3-DoF) control for an attractive-type magnetically-levitated (maglev) microrobot using machine learning. Traditional micromanipulation methods face challenges associated with friction and maintenance requirements; particularly in applications such as cell injection, where current devices tend to be limited by their high cost and maintenance requirements. The precise and low-maintenance nature of attractive-type levitation makes it a viable alternative to traditional micromanipulation methods, but a primary challenge lies in the difficulty of achieving precise 3-DoF control for such systems due to the complexity in the magnetic fields they generate. This research addresses this challenge by introducing a machine learning-based methodology that automates the learning of levitation dynamics across the workspace.

Our presented approach introduces and incorporates an automated system for generating training data with minimal human intervention, enabling a machine learning model to learn how the levitated microrobot responds to system inputs. This information is then used to establish 3-DoF position control of the levitated microrobot. Our automated methodology simplifies the setup process for new and newly-modified attractive-type levitation platforms, and is demonstrated to improve performance over conventional methods by accounting for observed variations in the levitation dynamics throughout the workspace; achieving up to a 20% reduction in root mean square error during trajectory tracking and up to a 36% reduction in step response settling times.

The results demonstrate the ability of our automated methodology to significantly reduce the accessibility barriers associated with establishing and modifying attractive-type maglev platforms; effectively replacing the usual methods of finite element simulation, precise magnetic field measurements, and/or analytical calculations while providing enhanced levitation control over traditional methods. This advancement contributes to the field of micromanipulation and microforce sensing by offering a more accessible and efficient approach to achieving precise control in attractive-type maglev systems.

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Mir Behrad Khamesee, for his expertise in acquainting me with his field of study, for his guidance in helping me complete this research, and for his patience in putting up with my shenanigans.

This work was supported by the Canada Foundation for Innovation (CFI) and by the Natural Sciences and Engineering Research Council of Canada (NSERC).

"Please use a different quote."
- My thesis committee (paraphrased)

Table of Contents

List of Figures ix

List of Tables xii

1 Introduction 1

1.1 Introduction to Micromanipulation 1

1.2 Introduction to Magnetic Levitation in Micromanipulation 6

1.3 Contribution of This Thesis 9

1.4 Objective & Thesis Outline 10

2 Magnetic Levitation System 12

2.1 Theory of Magnetic Forces and Torques 12

2.2 Principles of Attractive-Type Magnetic Levitation 16

2.2.1 Magnetic Coils 17

2.2.2 Magnetic Circuit Principles 19

2.2.3 Simplification of force equations 24

2.2.4 Pole Pieces 25

2.2.5 General Assembly of Levitation Apparatus 27

2.3 The Setup 28

2.3.1 Responsiveness of system 34

3 Control of the Microrobot **37**
3.1 Current State-of-the-Art . . . 37
3.1.1 Magnetic Force Approximation . . . 37
3.1.2 Alternate Embodiments of the State-of-the-Art . . . 40
3.1.3 Signal Processing During Levitation . . . 40
3.1.4 Limitations and Proposed Solutions . . . 42
3.2 Modified Control Algorithm . . . 45

4 Automated Training Data Generation **49**
4.1 Initial Calibrations . . . 51
4.2 Initial Hardware Considerations . . . 52
4.3 Initial Software Considerations . . . 54
4.4 Data Generation Loop . . . 55

5 Machine Learning Implementation for Horizontal Position Control **63**
5.1 Initial Data Processing . . . 63
5.2 Machine Learning-Assisted Position Control . . . 66
5.2.1 Motivation & Goals . . . 67
5.2.2 Generation & Usage of Wrench Matrices . . . 69
5.3 Final Parameters . . . 77
5.4 Alternative Methods . . . 78

6 Performance & Discussion of ML-Assisted Methodology **80**
6.1 Performance of Final Model . . . 80
6.1.1 Qualitative Results . . . 81
6.1.2 Quantitative Performance Analysis . . . 90
6.2 Future Steps for Performance Enhancement . . . 91
6.2.1 Hardware Configuration . . . 95

6.2.2 Position Control 96
6.2.3 Training Data Enhancements 97
6.3 Recommendations for Future Directions 99
6.3.1 Force Control 99
6.3.2 Open-Loop Position Control 102

7 Conclusion **104**

References **107**

List of Figures

1.1 Micro-probing for failure analysis of semiconductors [1] 2

1.2 Example of cell injection of a mouse embryo, approximately 55 μm in diameter [2] 3

1.3 Example physical model describing the known mechanical properties of cells, for use in force detection through visual servoing [3] 4

1.4 Conventional microforce sensor technologies 5

1.5 Example implementations of attractive- and repulsive-type maglev configurations 7

2.1 Illustration of Lorentz Force Law on a current-carrying wire 14

2.2 Illustration of the Biot-Savart Law 18

2.3 Illustration of thick cylindrical solenoid (see Equation 2.32) 20

2.4 Simulation of $\vec{B}$-field generated by a thick cylindrical coil 21

2.5 Magnetic field line confinement in magnetic circuit 22

2.6 Flow of information during closed-loop control of levitation. Analog signals are indicated with dashed arrows 28

2.7 Photograph of the final ATMA configuration 29

2.8 Diagram of the final ATMA configuration 31

2.9 Laser sensor detection range for Keyence LS-5041 33

2.10 Sample recording of "spikes" in laser sensor readings when object exceeds detection range 33

3.1 Current SOTA of 3DOF position control; see Equation 3.1 for $\mathbf{A}_{\text{SOTA}}$ (Adapted from [4]) . . . 39
3.2 Sample laser scan micrometer readings of a perfectly stationary object . . . 41
3.3 Feedforward + feedback vertical position controller, used in the modified algorithm . . . 48

4.1 Probability density cloud for the per-coil current perturbation selection, visualized with "offset" = 0 . . . 57
4.2 Flowchart of data generation loop . . . 61
4.3 Heatmap of all generated training data (specifically, "start" and "end" points associated with a new CCR). Note the overrepresentation of data points near the center of the workspace . . . 62

5.1 Power spectrum of unmodified, unfiltered laser sensor readings of a perfectly stationary object . . . 66
5.2 Visualization of the three distinct "subregions" defined in this research, each exhibiting distinct levitation dynamics . . . 70
5.3 Visual summary of shallow neural network structure (fully connected) . . . 71
5.4 Flowchart for generating the wrench matrix for a single "subregion". Note the x and y components using separate networks! . . . 74
5.5 Summary of final 3-DoF position control model . . . 75
5.6 Visualization of how the final horizontal control model selects (and interpolates) between the available wrench matrices based on microrobot position (in the 1D case) . . . 76
5.7 Example of the "true" B_{max} point (black dashed line) of a CCR "step" being extrapolated from raw data (blue), assuming either an overdamped (red) or underdamped (green) oscillation . . . 79

6.1 3D trajectory tracking comparison ("figure eight") . . . 82
6.2 3D trajectory tracking comparison ("rectangle") . . . 82
6.3 3D trajectory tracking comparison ("vertical helix") . . . 83
6.4 3D trajectory tracking comparison ("horizontal helix") . . . 83

6.5 3D trajectory tracking comparison ("oval") 84
6.6 Trajectory tracking comparison, for "figure eight" path (RMSE in mm) . . 85
6.7 Trajectory tracking comparison, for "rectangle" path (RMSE in mm) . . . 86
6.8 Trajectory tracking comparison, for "vertical helix" path (RMSE in mm) . 87
6.9 Trajectory tracking comparison, for "horizontal helix" path (RMSE in mm) 88
6.10 Trajectory tracking comparison, for "oval" path (RMSE in mm) 89
6.11 Transient response along the x-direction, within each of the "BASE", "HIGH Y", and "LOW Y" subregions 92
6.12 Transient response along the y-direction, both within the "BASE" subregion and in transitions between the "BASE" and "HIGH Y" / "LOW Y" subregions 93
6.13 Relative energy cost across the workspace for levitation at $z_d = 2.0$mm above the platform, during the randomized data generation process 97

List of Tables

5.1 Optimal hyperparameters for each subregion 78

6.1 Comparison of "wrench matrix selection" methods for final PID controllers, for x movement in the defined subregions 94

6.2 Comparison of "wrench matrix selection" methods for final PID controllers, for y movement *across* the defined subregions 94

Chapter 1

Introduction

1.1 Introduction to Micromanipulation

Magnetically levitated (maglev) microrobots have been of particular interest in the field of micromanipulation over the past several decades, offering unique advantages such as fully frictionless and untethered actuation that can address the limitations of traditional mechanical methods. The unique advantages of maglev micromanipulation altogether represent a significant advancement in micromanipulation technologies, including highly precise and delicate working environments such as interactions with living tissue on a cellular or subcellular level.

Micromanipulation broadly refers to the technologies and techniques for precisely interacting with objects at very small scales, such that achieving these tasks with the unaided human hand would be difficult or impossible. Many aspects of modern day-to-day life are enabled in part by the industrial automation of micromanipulation methods, such as the micromachining-enabled assembly of small gyroscopes embedded into modern handheld electronics or accelerometers used to deploy airbags in vehicles. Micromanipulation is also commonly used for research applications in areas such as testing the physical properties of micro- or nano-scale materials, or in mechanical probing for failure analysis of very small integrated electronics [1] (see Figure 1.1). The ability to interact with objects on very small scales is especially important when it comes to many biomedical applications, and one of the earliest such applications was robotic micromanipulation and injection of living cells [2].

Biomedical applications of micromanipulation can include manipulation, injection, cutting, fusing, etc. of living tissue within both the macro and micro domains, including for



Figure 1.1: Micro-probing for failure analysis of semiconductors [1]

applications in cell injection or minimally-invasive microsurgery [5]. These applications tend to require a level of precision beyond other micromanipulation applications (relative to their physical scale) due to the delicate nature of living tissue; for example, mechanically stitching a cut on a human arm, suturing a blood vessel that has been surgically incised or transected, or performing injection into an individual cell all require the living tissue to be punctured without actually destroying it. Cell manipulation and cell injection in particular offer a wide variety of applications for medicine and biomedical research; such as for injecting material into cells for use in gene therapy or in-vitro fertilization, testing the effects of new treatments on healthy or cancerous cells, or even simply characterizing the mechanical properties of cell membranes. However, these tasks require a particularly precise level of both position and force control of the end effector due to the small scale and delicate nature of individual cells.

The most traditional and broadly applicable methods of micromanipulation tend to be performed using motorized robotic micromanipulators and end effectors [6] that physically contact the objects, with the end effectors including devices such as microgrippers and micropipettes as seen in Figure 1.2. Many of these applications estimate applied forces with visual servoing techniques, although this method typically requires a detailed mechanical model of the manipulated object as seen in Figure 1.3 [3], which can sometimes be impractical in the concept of cell manipulation due to variation between the properties of individual cells of the same type [7]. For more general usage scenarios, force detection on such devices is typically performed by integrating some form of force sensor into the micromanipulator design, with the most common microforce-sensing methods including:

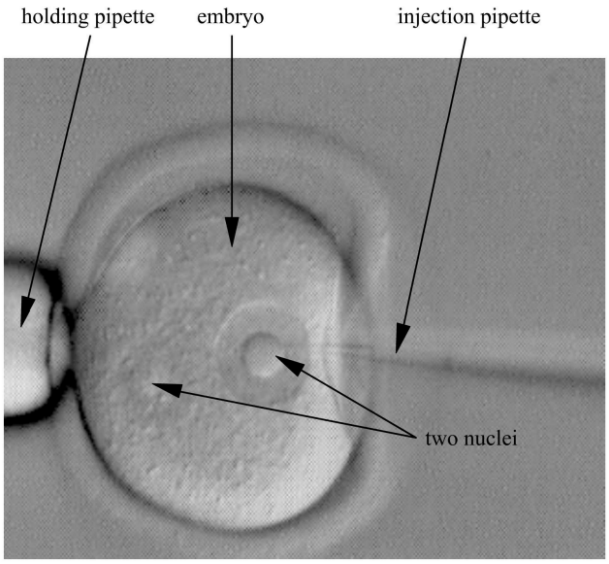


Figure 1.2: Example of cell injection of a mouse embryo, approximately 55 μm in diameter [2]

- Piezoresistive sensors (which are the most common option due to their low cost [8], and utilize the piezoresistive effect to produce a measurable change in resistance based on the strain deformation of the sensor)
- Piezoelectric sensors (which similarly operate using the piezoelectric effect, where the sensor material generates surface charges in response to changes in applied force; but notably do not respond to static forces)
- Capacitive sensors (which produce a measurable change in capacitance in response to physical deformation)
- Atomic force microscopy (AFM) probes (which measure forces with nN-scale precision and high reliability by optically measuring minute deflections of a cantilever with an extremely sharp tip that comes in contact with the object, but are restricted to an extremely narrow measurement range and tend to be extremely expensive)

However, simply affixing force sensors to the end effectors tends to be difficult in the context of micromanipulation, largely due to the small size and/or high fragility of the end effector. This makes such force detection methods ill-suited for many applications; especially in the context of cell manipulation, where clogging and breakage are common for the

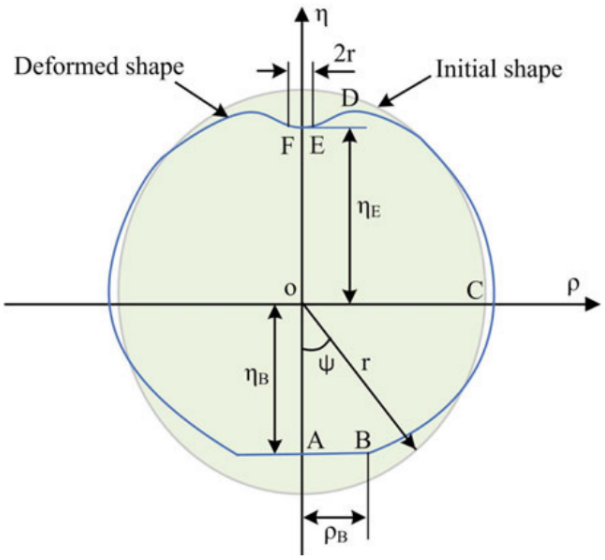
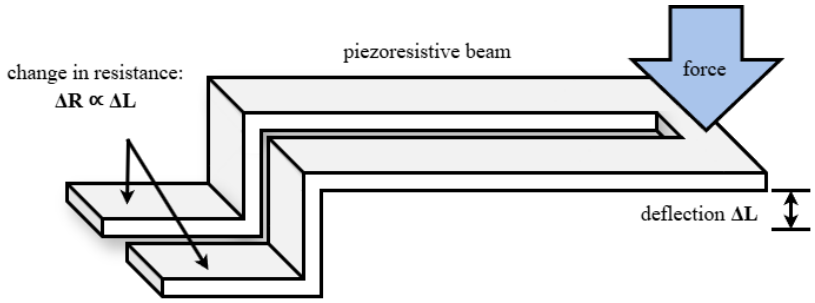


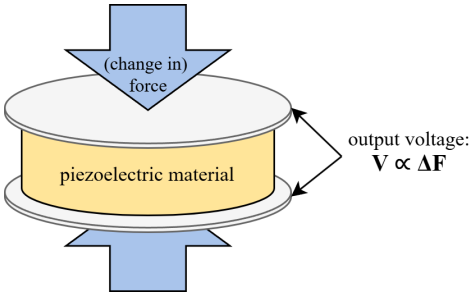
Figure 1.3: Example physical model describing the known mechanical properties of cells, for use in force detection through visual servoing [3]

micropipettes used in holding or injecting the cells [9] (see Figure 1.2 for an illustrative example). In addition to the difficulties associated with installing microforce sensors onto small (and sometimes highly fragile) end effectors, these mechanical micromanipulation technologies also tend to be susceptible to imprecisions due to step loss and gear backlash. Many such applications are further complicated by the adhesion effects between end effector and object/environment that become far more dominant at small scales [10], which are also an important reason for the relative rarity of automation in micromanipulation systems [11]. Furthermore, mechanical methods are susceptible to both friction and eventual deterioration over time; and maintenance or replacement of damaged/missing components can incur unwanted expenses.

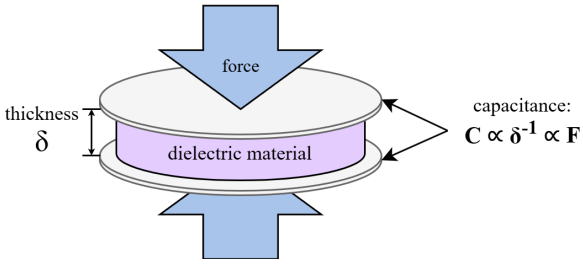
These are flaws to which traditional methods of contact micromanipulation are generally susceptible. However, a purely magnetically levitated "end effector" can alleviate many of these issues due to its completely untethered and frictionless nature.



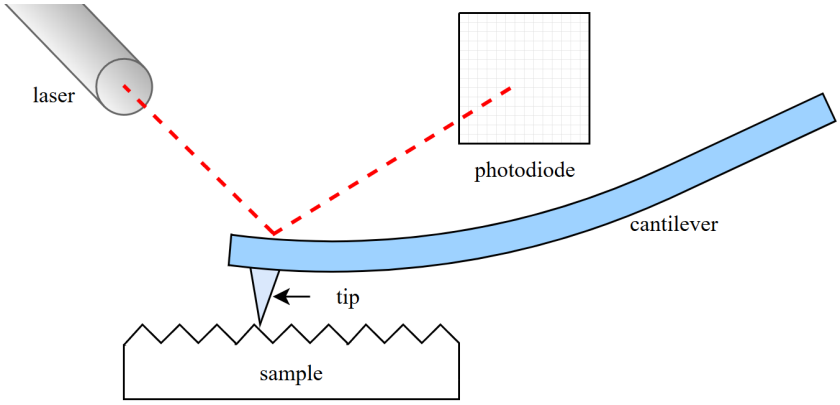
(a) Piezoresistive force sensor



(b) Piezoelectric force sensor



(c) Capacitive force sensor



(d) Atomic force microscopy (AFM) probe

Figure 1.4: Conventional microforce sensor technologies

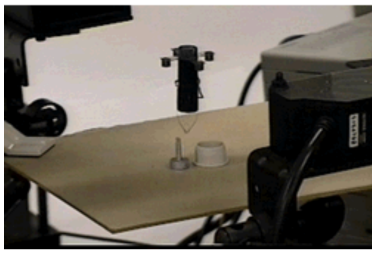
1.2 Introduction to Magnetic Levitation in Micromanipulation

Magnetic levitation contains a variety of unique properties that can make it a viable solution to many of the common caveats associated with traditional micromanipulation methods.

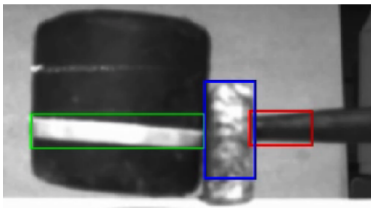
By generating and controlling magnetic fields from outside the workspace using an array of solenoids, it is possible to perform micromanipulation tasks quickly and precisely using a fully untethered magnetically levitated microrobot. In the context of traditional methods of contact micromanipulation, maglev microrobots offer the significant advantages of being completely frictionless and not being reliant on mechanical parts that are susceptible to wear and tear, while also being well-suited for automated or teleoperated micromanipulation of matter in sealed environments such as in chemical or biomedical applications (where there is a need to reduce the possible environmental contaminants interacting with the material, and/or it is necessary to hazardous substances a safe distance away from the operator).

Many fundamentally different maglev implementations have been proposed and implemented over the past several decades; with recent examples including a fully untethered "claw" that can perform pick-and-place operations [12], a levitated microrobot that can perform cooperative manipulation of objects in coordination with a robotic micromanipulator [4], and a planar magnetically levitated Halbach array platform that can rapidly move across a wide workspace while carrying payloads [13] (see Figure 1.5). However, when it comes to moving an untethered "end effector" across a reasonably wide and open workspace with control over at least 3 degrees of freedom for the purpose of interacting with objects, common configurations include those where a magnetized carrier is levitated via repulsive magnetic forces from below (hereafter "repulsive-type maglev apparatus", or "RTMA"), or those where a magnetized carrier is levitated via attractive magnetic forces from above (hereafter "attractive-type maglev apparatus", or "ATMA"). A key difference between the two configurations is the inherent stability of the translational magnetic forces, which can be understood intuitively by holding a permanent bar magnet in each hand and trying to balance their mutual attractive/repulsive force against gravity.

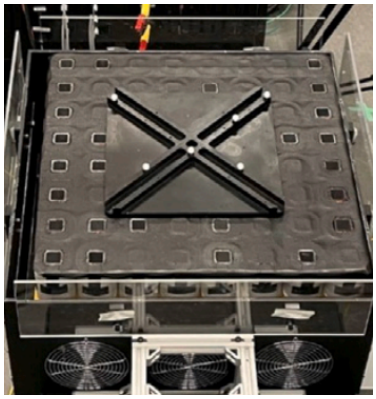
Magnetic repulsion tends to become stronger the closer the magnetic field sources are to each other, which is why the vertical magnetic forces associated with repulsive-type setups are inherently stable. However, the magnetic torques are distinctly *unstable* due to the far end of the magnet being strongly attracted to the source of the "repulsive" force. As such, the levitated devices in repulsive-type configurations tend to be very wide



(a) "Attractive-type" maglev pick-and-place system [12]



(b) "Attractive-type" maglev cooperative manipulator [4]



(c) "Repulsive-type" levitated platform with a Halbach array carrier[13]

Figure 1.5: Example implementations of attractive- and repulsive-type maglev configurations

in construction (whether as a wide single-magnet carrier or as a combination of magnets spread out over a wide area) relative to the coils producing the driving force, so as to facilitate closed-loop control of their tilt angle. These wider structures, as well as the tendency for repulsive levitation force to passively increase with closer proximity with the coils, make the repulsive-type configuration significantly better suited for carrying heavy payloads across wide workspaces (e.g., as a load-bearing conveyor platform in microfactory applications [14][15]).

In contrast, magnetic attraction can levitate a device from above by balancing the magnetic force against gravity, where the tendency of the magnetic force to increase with closer proximity to the coils leads to vertical magnetic force being inherently *unstable* in these configurations. Likewise, the magnetic torque applied to the levitated device is stable, as the "attracted" pole of the levitated device is already oriented to be closer to the electrified coils than the repelled end (although the exact details of this interaction will be described in greater detail in Section 2.1). The instability of the vertical force makes this method less suitable for carrying heavy payloads; however, notably, the *horizontal* magnetic force applied by an attractive-type configuration is also stable (provided that the vertical force is controlled such that the levitation height is constant). This means that the horizontal position of the ATMA-levitated magnet can be controlled with relatively little effort even if the levitated magnet is much smaller than the coils themselves [4] [14], which means that attractive-type setups tend to naturally be more suitable for miniaturization

into the micro-domain.

The design of the levitated device in attractive-type maglev configurations is relatively unconstrained in terms of width and minimum allowed size. This makes it so that ATMA-levitated microrobots can be composed of a simple cylindrical magnet, which can be as small as necessary for the desired application and subsequently extremely inexpensive to replace if lost or damaged [16]. The low replacement cost, low maintenance requirements (due to the absence of friction), and relatively low investment cost (compared to commercially available contact micromanipulators for use in cell injection or microsurgery, for example) give attractive-type maglev setups the potential to be a particularly economical option for a wide variety of micromanipulation tasks; capable of solving many of the problems associated with traditional contact micromanipulation methods with a lower associated financial barrier.

Many significant advancements have been made towards the usage of ATMAs in micromanipulation. For example, previous research has demonstrated the viability of the untethered levitated microrobot being controlled in 3-DoF with both position control and haptic force feedback via teleoperation [17], as well as the ability for the microrobot to be used in tandem with a mechanical micromanipulator for cooperative manipulation of objects [4]. Prior research has also demonstrated the viability of equipping the microrobot with a piezoelectrically-actuated gripper for use in pick-and-place operations [12]. These prior experiments serve as valuable indicators of the vast and varied potential applications of ATMAs, highlighting the numerous potential opportunities for innovation across multiple fields.

However, as with any micromanipulator, establishing a method to control the position of the levitated microrobot is necessary, and the problem herein lies in the complexity of this process. In the current state-of-the-art (or "SOTA") for attractive-type setups, establishing the control model typically involves obtaining an accurate model for the magnetic fields generated by each coil throughout the workspace (which directly corresponds to the magnetic forces and torques experienced by a magnetic dipole, as will be explained in Chapter 3). Establishing such a model is not straightforward, and the process typically involves a combination of intensive analytical calculations, numerous empirical measurements of the magnetic fields produced by each coil using expensive precision instruments, and/or simulation of these magnetic fields in often-expensive proprietary software. Additionally, these steps will typically need to be redone if there are any significant changes to the configuration such as addition, removal, and/or repositioning of the electromagnetic coils.

All of this can form a steep accessibility barrier in terms of prerequisite knowledge,

implementation time, and/or direct financial cost. This accessibility barrier could make it incredibly difficult for experts in different fields (e.g., a biomedical engineer researching the mechanical properties of cells) to access such a device, let alone make modifications to it if necessary. Attractive-type maglev apparatuses have incredible potential for micromanipulation tasks in an extensive variety of research and manufacturing applications; however, in order to facilitate the usage of such powerful devices in practical research settings where time and funding may not come in unlimited supply, a reduction to this accessibility barrier is necessary.

1.3 Contribution of This Thesis

The main motivation for the research presented in this thesis lies in the low accessibility of micromanipulation-ready attractive-type maglev apparatuses. While the construction and maintenance of such a device can be significantly less expensive compared to commercially-available micromanipulation methods, the difficulty of establishing a viable model for 3-DoF position control forms an accessibility barrier that can be insurmountable in various research settings if such an ATMA setup is to be constructed or modified. Much of this accessibility barrier, as well as much of the initial configuration time for developers of such a device (who are assumed to be more familiar with electromagnetic actuation and control), could be eliminated if the initial task of establishing a viable control model could be largely relegated to a computer instead.

An ideal solution to this problem would include a methodology by which 3-DoF control can be established on ATMAs in an automated manner, preferably in a manner that naturally adapts to differences in construction for the specific configuration being used. The contribution of the research presented in this thesis is, simply speaking, just that. My work introduces a novel method by which 3-DoF position control can be established for an ATMA without the need for the intensive magnetic field modelling seen in traditional methods, using a machine learning-assisted process. Furthermore, my work introduces a novel method by which the necessary training data for these machine learning implementations, containing valuable information describing variations in the levitation dynamics as a levitated microrobot moves across its workspace, can be generated in a mostly automated manner with little to no need for human intervention.

By combining the fundamental design principles of an ATMA, a simple vertical position controller for the levitated microrobot, and my newly-introduced method of automatically "training" the ATMA to perform horizontal control, the research presented in this thesis also serves as a proof-of-concept by which an ATMA can be constructed and/or modified

and subsequently trained to levitate a magnetic microrobot with control over 3 degrees of freedom with greatly reduced need for performing mathematical modelling or simulations. This overall methodology has the potential to greatly reduce the accessibility barrier (in regards to expense, time, and prerequisite knowledge/expertise) for creation, modification, and general usage of an ATMA device; especially for experts across various fields who are not necessarily familiar with electromagnetic actuation and control but could make use from a frictionless, maintenance-free actuator.

1.4 Objective & Thesis Outline

The primary objective of this thesis is to present the novel method for automated generation of training data that describe the ATMA's levitation dynamics across the available workspace, as well as the method by which these data are used to directly enable 3-DoF position control; effectively "skipping" the intensive modelling steps that are traditionally necessary for enabling 3-DoF control of an ATMA-levitated microrobot.

Additionally, following the need for improved accessibility of ATMA configurations, an overarching secondary objective of this thesis is to summarize the overall physical concepts and resulting design principles associated with construction of an ATMA capable of 3-DoF control of a levitated microrobot. The intention is to provide a broad summary of the information necessary such that a researcher could, in principle, construct a similar ATMA from scratch with the capability for 3-DoF position control of a levitated microrobot. Expansion of this methodology to 3-DoF force control lies beyond the scope of this thesis; as such, potential methods of expansion to force control (including those that have already been implemented and empirically verified in prior research) are recommended as future work.

With these goals in mind, the remainder of this thesis will be presented as follows:

- Chapter 2 will review the theoretical foundations by which attractive-type magnetic levitation functions, including fundamental principles of electromagnetism and the subsequent design considerations for construction of an ATMA. Section 2.3 will detail the hardware components of the actual levitation device employed in this research, as well as the software components by which the control model is deployed for actual use.

- Chapter 3 aims to summarize the current state-of-the-art of establishing horizontal position control for an ATMA based on its particular configuration. While a variety of

methods can in principle be employed, this section will focus on the method presented in the most recent predecessor of this research. Section 3.2 aims to propose a variety of methods by which machine learning may be able to establish 3-DoF control over a levitated microrobot, with the goal of addressing or bypassing the difficulties and weaknesses associated with traditional methods.

- Chapter 4 briefly explores the potential forms that a successful machine learning implementation may take in the context of establishing 3-DoF control for an ATMA. It then goes on to present the novel methodology for generating the training data used in the final successful implementation of this research; detailing all stages in this process from the data generation algorithm itself to hardware/software considerations that should be taken into account for the sake of time efficiency and safety.

- Chapter 5 presents the methodology by which these training data are utilized to establish the final 3-DoF position control model, including the data format and training of the machine learning models, as well as the methodology for quantifying and accounting for differences in levitation dynamics across the workspace.

- Chapter 6 showcases the actual performance of this final ML-assisted control implementation, and compares its performance qualitatively and quantitatively with 3-DoF control established via the traditional method described in Chapter 3. Section 6.2 provides recommendations for future work, in terms of immediate improvements to position control performance that can be made based on the observed strengths and weaknesses of the presented methodology. Section 6.3 provides recommendations for future work in a broader sense, with a focus on potential expansion to force control and subsequent usage in micromanipulation applications.

- Finally, Chapter 7 summarizes and concludes the work presented in this thesis, as well as its contributions to the field and future steps that can be made in subsequent research.

Chapter 2

Magnetic Levitation System

This section will provide a detailed summary of our knowledge of the principles behind which a typical ATMA works. This will begin with general foundational knowledge on the concepts of magnetic forces and torques (i.e., the physical principles that allow for magnetic actuation to happen in the first place), before shifting focus to the design principles of a typical ATMA levitating a simple microrobot (taking the form of a single axisymmetric permanent magnet in this work) and the advantages this methodology provides.

2.1 Theory of Magnetic Forces and Torques

Our modern understanding of electricity and magnetism can be most succinctly summarized by Maxwell's Equations, a set of coupled partial differential equations that unify the concepts of electricity and magnetism by describing the interplay between electric fields ($\vec{E}$, typically measured in volts/meter) and magnetic fields ($\vec{B}$, Teslas), thus forming the basis of electromagnetic theory. These fundamental equations are:

$$\text{Gauss's Law:} \qquad \nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0} \qquad \oint_{\mathcal{S}} \vec{E} \cdot d\vec{A} = \frac{Q_{\text{enc}}}{\epsilon_0} \tag{2.1}$$

$$\text{Gauss's Law (M):} \qquad \nabla \cdot \vec{B} = 0 \qquad \oint_{\mathcal{S}} \vec{B} \cdot d\vec{A} = 0 \tag{2.2}$$

$$\text{Faraday's Law:} \qquad \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \qquad \oint_{\mathcal{C}} \vec{E} \cdot d\vec{l} = -\frac{d}{dt} \int_{\mathcal{S}} \vec{B} \cdot d\vec{A} \tag{2.3}$$

$$\text{Ampere's Law:} \qquad \nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \qquad \oint_{\mathcal{C}} \vec{B} \cdot d\vec{l} = \mu_0 I_{\text{enc}} + \mu_0 \epsilon_0 \frac{d}{dt} \int_{\mathcal{S}} \vec{E} \cdot d\vec{A} \tag{2.4}$$

where ρ is volumetric electric charge density, $\vec{J}$ is the current density (i.e., the amount of charge per unit time flowing through a cross-sectional area, measured in Amps/m^2), and μ_0 and ϵ_0 are respectively the magnetic permeability and dielectric permittivity of free space. For the "integral forms" of these fundamental equations, $d\vec{A}$ refers to the infinitesimal area element of a surface $\mathcal{S}$, $d\vec{l}$ refers to the infinitesimal length element of the closed contour $\mathcal{C}$ that bounds $\mathcal{S}$, I_{enc} is the current (in Amps) passing through the closed loop $\mathcal{C}$, and t simply refers to time.

In the context of magnetic levitation, the equations of particular importance are Gauss' Law for Magnetism (which states that there are no magnetic monopoles), and Ampere's law (stating that a linear flow of current will generate an axial magnetic field; which consequently means that a "loop" of current will produce a magnetic field aligned with the loop's axis). While less immediately relevant, it is also worth noting Faraday's law and Lenz's law (which state that the magnetic field changing in a conductive medium will induce an electric field, such that the induced electric field generates a magnetic field "resisting" the change in the original magnetic field).

In addition to the Maxwell equations is another fundamental law called the Lorentz Force Law:

$$\vec{F} = q\left(\vec{E} + \vec{v} \times \vec{B}\right) \tag{2.5}$$

which describes the force $\vec{F}$ that electric and magnetic fields will apply to a particle with electrical charge q, travelling at velocity $\vec{v}$ in an external magnetic field $\vec{B}$. In the context of magnetic actuation, the Lorentz Force Law can also be rewritten to show the force applied

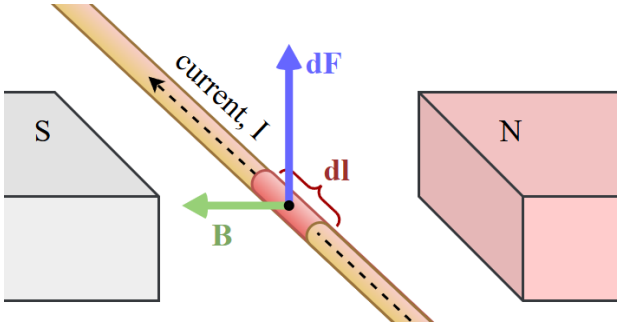


Figure 2.1: Illustration of Lorentz Force Law on a current-carrying wire

to a current-carrying wire in the absence of an external electric field:

$$\vec{F} = I \int \left(d\vec{l} \times \vec{B} \right) \ , \tag{2.6}$$

$$d\vec{F} = I d\vec{l} \times \vec{B} \tag{2.7}$$

where I is the current flowing along the length of elements $d\vec{l}$ comprising the wire, in the presence of an external magnetic field $\vec{B}$.

In the context of magnetic forces and torques applied to a permanent magnet, it is important to note that the simplest permanent magnet with magnetic dipole moment $\vec{P}$ can be represented as an infinitesimal current loop of surface area S with current i facing in the direction $\hat{n}$, according to the Amperian loop model [18]:

$$\vec{P} = iS\hat{n} \tag{2.8}$$

In the presence of an external magnetic field $\vec{B}$, we can calculate the magnetic force on this simple permanent magnet as the Lorentz force throughout the equivalent current loop. That is, if we break the loop into small segments $d\vec{l}$, then the force applied to each of these segments is:

$$d\vec{F} = i d\vec{l} \times \vec{B} \ ; \tag{2.9}$$

and torque applied to each segment is:

$$d\vec{\tau} = \vec{r} \times d\vec{F} \ , \tag{2.10}$$

$$\vec{\tau} = i \oint_C \left[\vec{r} \times \left(d\vec{l} \times \vec{B} \right) \right] \ ; \tag{2.11}$$

where $\vec{r}$ refers to the "lever arm" from the point that the force is applied to the axis of rotation. By applying the vector triple product expansion $\vec{a}\times\left(\vec{b}\times\vec{c}\right)=\vec{b}\left(\vec{c}\cdot\vec{a}\right)-\vec{c}\left(\vec{a}\cdot\vec{b}\right)$, this becomes:

$$\vec{\tau}=i\oint_{\mathcal{C}}\left[d\vec{l}\left(\vec{B}\cdot\vec{r}\right)-\vec{B}\left(\vec{r}\cdot d\vec{l}\right)\right]\ ; \tag{2.12}$$

and by applying Stokes' Theorem $\oint_{\mathcal{C}}\vec{F}\cdot d\vec{r}=\iint_{\mathcal{S}}\left(\nabla\times\vec{F}\right)\cdot d\vec{S}$, this becomes:

$$\vec{\tau}=i\left(\iint_{\mathcal{S}}\left[d\vec{S}\times\nabla\left(\vec{r}\cdot\vec{B}\right)\right]-\vec{B}\iint_{\mathcal{S}}\left(\nabla\times\vec{r}\right)\cdot d\vec{S}\right)\ . \tag{2.13}$$

Because $\vec{r}$ is a straight line, and the current loop is infinitesimally small (i.e., we know that $\vec{B}$ does not change significantly throughout the span of the current loop), it follows that $\nabla\times\vec{r}=0$, and $\nabla\left(\vec{r}\cdot\vec{B}\right)=\vec{B}$. As such, our expression for $\vec{\tau}$ reduces to:

$$\vec{\tau}=i\iint_{\mathcal{S}}d\vec{S}\times\vec{B}\ , \tag{2.14}$$

$$=i\vec{S}\times\vec{B}\ ; \tag{2.15}$$

which, as seen from our definition of the dipole in Equation 2.8, is simply:

$$\vec{\tau}=\vec{P}\times\vec{B}\ . \tag{2.16}$$

Following this, the magnetic force applied to the dipole can be thought of in terms of energy. We can see from Equation 2.16 that the torque is minimized when the magnetic dipole moment and the magnetic field lines are aligned. If the dipole's alignment deviates from the $\vec{B}$-field by some angle θ, then the potential energy released upon a slight angular displacement $d\theta$ is:

$$dU=-dW=\tau d\theta\ , \tag{2.17}$$

$$=PB\sin\theta d\theta\ ; \tag{2.18}$$

where W refers to the amount of work done on the dipole by the external $\vec{B}$-field. Thus, the total potential energy released when aligning this dipole to the $\vec{B}$-field (i.e., returning to $\theta=0$) is:

$$U=\int_{\theta}^{0}PB\sin\theta d\theta\ , \tag{2.19}$$

$$=PB\left(1-\cos\theta\right)\ . \tag{2.20}$$

Since only changes in potential energy are physically meaningful, we can visually simplify this equation by defining $\theta = \frac{\pi}{2}$ to be the point of zero potential. As such, we define the potential energy for a magnetic dipole in an external magnetic field as:

$$U = -PB\cos(\theta) \ , \tag{2.21}$$

$$= -\vec{P} \cdot \vec{B} \ ; \tag{2.22}$$

where the negative sign indicates decreasing potential energy as the dipole aligns itself with the $\vec{B}$-field. From here, the magnetic force is simply the spatial gradient of the work done on the dipole by the magnetic field [19]. As such, the magnetic force applied to a magnetic dipole $\vec{P}$ by an external magnetic field $\vec{B}$ is:

$$\vec{F} = \nabla W \ , \tag{2.23}$$

$$= -\nabla U \ , \tag{2.24}$$

$$= \nabla \left(\vec{P} \cdot \vec{B} \right) \ . \tag{2.25}$$

2.2 Principles of Attractive-Type Magnetic Levitation

A key takeaway of the above is that spatial gradients in an external magnetic field will apply a driving force to a magnetic dipole within it. This is the fundamental principle behind which magnetic actuation can occur, and clever control of this principle can allow for controlled levitation of magnets. The way that magnetic levitation is enabled for a typical ATMA is, simply put, through fine control of the current flowing through an electromagnetic coil (which subsequently affects the magnetic field produced by the turns of wire).

Recall that attractive-type levitation is fundamentally unstable in terms of vertical position control; as such, stable levitation is enabled by continuously tracking the state of the levitated microrobot and accounting for this state when deciding the currents applied to the coils (i.e., creating a closed-loop control system for the vertical position of the microrobot). Creating and controlling a magnetic field for ATMAs is typically done using helical electromagnetic coils (although coils of different shapes, such as square coils, may be feasible in certain embodiments due to increased packing density leading to enhanced $\vec{B}$-fields for the space required [14]). The magnitude of the $\vec{B}$-field produced by each coil

is simply proportional to the electrical current flowing through it, and control of this $\vec{B}$-field is established by changing the coil currents very significantly within short timeframes without the need for any moving parts.

While the magnitude of the magnetic field generated by a single coil simply varies with the electrical current flowing through it, the direction of the field lines in 3D space are largely fixed. As such, control over the $\vec{B}$-field lines' directions in the workspace (i.e., the orientation of a small levitated magnet) requires multiple coils at different orientations to be arranged in a particular way [20]; whereas setups with coils oriented in the same direction are typically restricted to 3D translation of a small magnet and do not intrinsically have the means to modify the magnet's orientation. This thesis utilizes and focuses primarily on such a configuration. The remainder of this subsection will discuss the physical principles and resulting design choices of a typical ATMA that is capable of 3DoF control of a small magnet.

2.2.1 Magnetic Coils

As seen in [maxwell's equations above], Ampere's law states that a flow of electric charges will generate a magnetic field circling around the direction of the current flow. To elaborate, the Biot-Savart law states that the magnetic field produced by a current-carrying wire is as follows:

$$\vec{B}(\vec{r}) = \frac{\mu_0}{4\pi} \int \frac{I\, d\vec{l} \times (\vec{r} - \vec{r'})}{|\vec{r} - \vec{r'}|^3} \tag{2.26}$$

where $\vec{r} = r\hat{r} = \sqrt{x^2 + y^2 + z^2}\,\hat{r}$ is the position of the point of interest (or "observer"), $\vec{r'}$ is the position of the line segment $d\vec{l}$, and I is the current flowing through the wire (see Figure 2.2 for an illustration).

As implied by Ampere's law, a circular current-carrying loop of wire will generate a magnetic field through the loop that is strongest (and perfectly parallel to the loop axis) at the centre. In the special case that the point of interest is along the loop axis $\hat{z}$, the magnetic field produced by a current loop located at z=0 is:

$$\vec{B} = B_z\hat{z} = \frac{\mu_0 I R^2}{2(a^2 + z^2)^{3/2}}\hat{z} \tag{2.27}$$

where a is the radius of the loop.

The *off-axis* magnetic field for this current loop is more difficult to calculate analytically, and while the exact analytical solution is known, it cannot be written as a closed-form

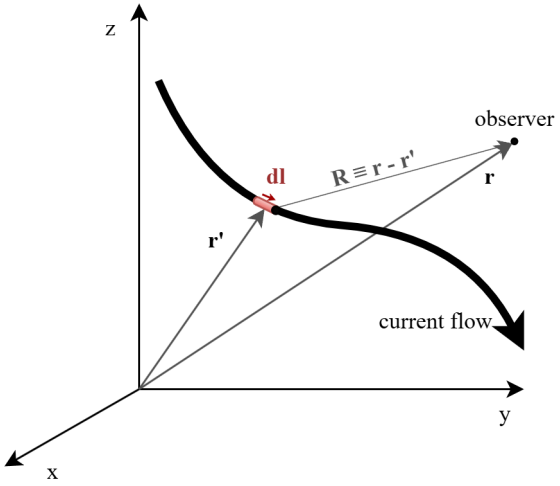


Figure 2.2: Illustration of the Biot-Savart Law

expression [21] [22] [23]. The exact expression for the off-axis magnetic field of a current loop (and the derivation thereof) is beyond the scope of this work; however, for illustrative purposes, we can show the first step in the calculation starting from Equation 2.26 by defining the wire segment $d\vec{l}$ as being a segment of the loop with radius a, such that

$$d\vec{l} = a\,(-\sin\phi\,\hat{x} + \cos\phi\,\hat{y})\;d\phi \;\;, \tag{2.28}$$

$$\vec{B}(\vec{r}) = \frac{\mu_0 I a}{4\pi}\int_0^{2\pi}\frac{(-\sin\phi\,\hat{x} + \cos\phi\,\hat{y})\times\vec{R}}{|\vec{R}|^3}d\phi\;; \tag{2.29}$$

where $\vec{R} \equiv \vec{r} - \vec{r'}$, and ϕ is the azimuthal angle of the line segment around the loop axis $\hat{z}$. If we make the simplifying assumption that the observer distance is much larger than the radius of the current loop (i.e., $r \gg a$), the magnetic field can ultimately be approximated and written as:

$$\vec{B}(\vec{r}) \approx \frac{\mu_0 I\pi a^2}{4\pi}\left[\frac{2\cos\theta}{r^3}\hat{r} + \frac{\sin\theta}{r^3}\hat{\theta} + 0\hat{\phi}\right] \tag{2.30}$$

$$\tag{2.31}$$

where θ is the polar angle of the observer from the loop axis (or z-axis) to the loop plane (or x-y plane). Note that the loop is symmetrical about the $\hat{\phi}$-axis, and thus the resulting $\vec{B}$-field has no $\hat{\phi}$ component.

Taking this a step further, a thin solenoid essentially comprises many aforementioned "current loops" stacked on top of one another, and multi-layer solenoids can be thought of as multiple concentric thin solenoids with the same current. Based on this, the magnetic field of a thick cylindrical coil with height L, inner radius r_{in}, outer radius r_{out}, and cross-sectional winding density (i.e., number of turns per unit area) σ can be written as:

$$\vec{B}(\vec{r}) = \frac{\mu_0 I \sigma}{4\pi} \int_{r_{\text{in}}}^{r_{\text{out}}} \int_{-\frac{L}{2}}^{\frac{L}{2}} \int_0^{2\pi} \frac{(-\sin\phi\,\hat{x} + \cos\phi\,\hat{y}) \times \vec{R}}{|\vec{R}|^3} d\phi\, dz\, dr \;; \tag{2.32}$$

and under the simplifying assumption of large observer distance (i.e., $r \gg a$, and thus $r \approx R$), the magnetic field of a thick solenoid can ultimately be approximated as:

$$\vec{B}(\vec{r}) \approx \frac{\mu_0 N I \pi a^2}{4\pi} \left[\frac{2\cos\theta}{r^3}\hat{r} + \frac{\sin\theta}{r^3}\hat{\theta} + 0\hat{\phi} \right] \;, \tag{2.33}$$

where $N \approx \sigma L\,(r_{\text{in}} - r_{\text{out}})$ is simply the total number of turns in the coil, and $a \approx (r_{\text{in}} - r_{\text{out}})$ is the "average" turn radius (assuming uniform winding density). Note that this is just Equation 2.33, except I has been substituted with NI.

Figure 2.4 provides a qualitative picture of how the magnetic field generated by a thick solenoid behaves. We can see that the magnetic field is strongest inside the coil, and tends to be stronger along the coil's axis; but we can also see a behaviour that may pose a significant problem when performing levitation: Even along this axis, $\vec{B}$-field's magnitude decreases with $\sim r^3$, causing the magnetic field strength to become orders of magnitude weaker with slight movements away from the magnet, resulting in enormous energy costs for this levitation and/or a very restricted vertical range in which this levitation can occur. This, of course, is not desirable– and the solution to this issue is to minimize the magnetic energy being "wasted" through the use of magnetic circuits.

2.2.2 Magnetic Circuit Principles

As seen in Figure 2.4 and Equation 2.33, the strength of a $\vec{B}$-field produced by a solenoid drops with $\sim r^3$, meaning that the effective "working range" is either heavily restricted, or the input current needs to be increased with $\sim r^3$ to compensate. For a typical ATMA, neither of these solutions are particularly realistic due to the benefits of a relatively large workspace and the need for reasonable energy costs during operation, meaning that a more sophisticated approach is required.

One such method is to design an ATMA around the principle of magnetic circuits, which can be thought of as an analog to electric circuits but for magnetic flux $\Phi = \iint_S \vec{B} \cdot d\vec{A}$

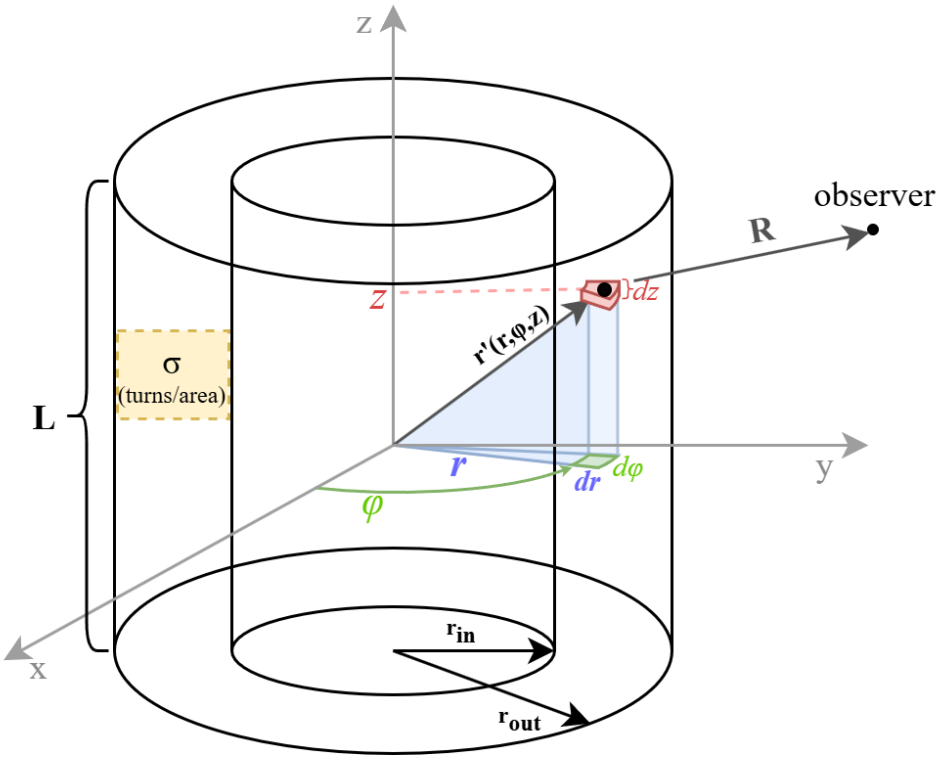


Figure 2.3: Illustration of thick cylindrical solenoid (see Equation 2.32)

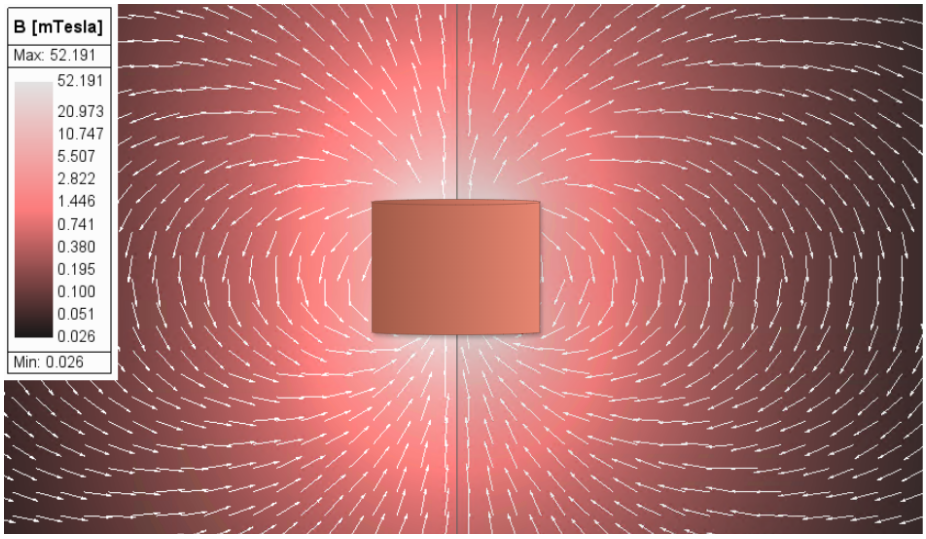


Figure 2.4: Simulation of $\vec{B}$-field generated by a thick cylindrical coil

(where $\vec{A}$ represents the area of some surface S) rather than electrical current. Ferromagnetic materials have the effect of "attracting" magnetic field lines (see Figure 2.5); meaning that the magnetic field generated by each coil can be "channeled" in a controlled manner. The general idea behind magnetic circuits is to use ferromagnetic materials to direct the magnetic field lines such that they are neatly contained in a closed loop path, thus eliminating the issue of magnetic field strength weakening due to the field lines spreading away from the region where they are to be utilized.

To elaborate on the electric circuit analogy, we can describe a magnetic circuit in a manner that is superficially similar to Ohm's law for electric circuits. The equation used for this is called Hopkinson's law:

$$\mathcal{F} = \Phi\mathcal{R} \tag{2.34}$$

where $\mathcal{F}$ [in ampere-turns] is the "magnetomotive force" of the circuit (and serves as the driving force behind a hypothetical "magnetic charge", analogous to electromotive force in Ohm's law), Φ [in Webers, or $\mathrm{T} \cdot \mathrm{m}^2$] is analogous to electrical current, and $\mathcal{R}$ [in H^{-1}] is the "reluctance" of the circuit (closely related to the "leakage flux" that does not follow the intended path, and is analogous to electrical resistance in Ohm's law).

Notably, if we trace the shape of the magnetic circuit as a closed loop $\mathcal{C}$, the magneto-

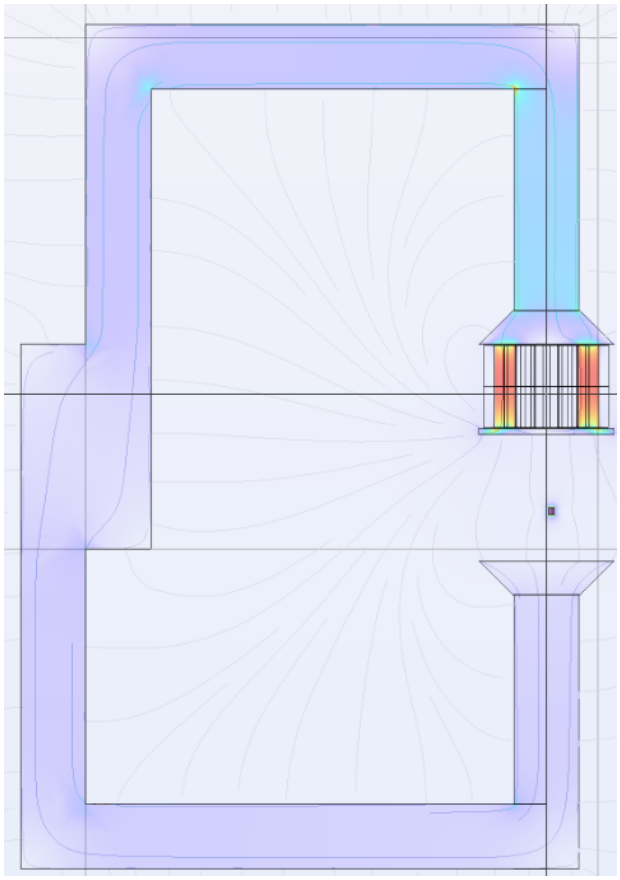


Figure 2.5: Magnetic field line confinement in magnetic circuit

motive force $\mathcal{F}$ can also be defined using Ampere's law as:

$$\mathcal{F} = NI = \oint_{\mathcal{C}} \vec{H} \cdot d\vec{l} \tag{2.35}$$

where N and I are (respectively) the number of turns and the current flowing through the (assumed single) electromagnetic coil that drives the magnetic circuit. Note that multiple coils are commonly used rather than just one, so NI is effectively a simplification of the multiple coils driving the circuit instead; such that a circuit driven by (for example) six identical coils of n turns each would have a magnetomotive force of $\mathcal{F} = NI = nI * 6$.

Continuing the analogy of electric circuits, the reluctance $\mathcal{R}$ in a magnetic circuit is a property of both the material and geometry used throughout the loop. The reluctance of each component i in the circuit can be calculated as:

$$\mathcal{R}_i = \frac{\ell_i}{\mu_i \mathcal{A}_i} \tag{2.36}$$

where ℓ is the length of the component, μ is the magnetic permeability of the component (higher for ferromagnetic materials, and lower in a vacuum), and $\mathcal{A}$ is the (assumed uniform) cross-sectional area of the component. The total reluctance of the magnetic circuit with relevant components connected in "series" is calculated similarly to the electric circuital counterpart:

$$\mathcal{R} = \mathcal{R}_1 + \mathcal{R}_2 + \mathcal{R}_3 + \ \ldots\ . \tag{2.37}$$

Of course, the point of an ATMA is to be able to levitate an object, meaning that there must be a gap in the magnetic circuit that allows this, presumably comprising a fluid with no magnetic properties (such as air). As implied by Equation 2.36, we want to minimize the reluctance of this gap by keeping it as short as possible, and increase the cross-sectional area of this gap (or, rather, the cross-sectional area of the ferromagnetic components of the circuit) so as to minimize the leakage of magnetic field lines.

With this in mind, a simple yet effective design for an ATMA is to create a magnetic circuit out of a ferromagnetic material that does not retain a magnetic polarization after exposure to an external $\vec{B}$-field (e.g., soft iron) in somewhat of a "C" shape (hereafter referred to as the "yoke" of the ATMA), with the air gap of this iron yoke containing the workspace for the levitated magnet. The vertically-oriented electromagnets are affixed to the yoke at the top of the air gap so that their contribution to the magnetic field in the workspace is maximized, and the air gap is ideally as small as possible (while still being large enough to accommodate the desired levitation workspace) so as to minimize leakage

flux. Finally, permanent magnets can also be integrated into this design (in series; e.g., inserted somewhere in the yoke) so as to reduce the electrical current required for producing some magnetic field $\vec{B}$ in the workspace. Figure 2.5 shows a qualitative example of the behaviour of the magnetic field throughout the ATMA used in this research. Note that the magnetic field lines are largely contained within the iron yoke, and that while there is some leakage flux in the air gap containing the levitation workspace, the magnetic field strength in this gap is far more consistent in terms of both direction and magnitude than it would be with just the coils alone.

Using the equations and principles described in this section, it is in principle possible to "design" a magnetic circuit from scratch (including coils and permanent magnets) that is suited for a particular set of specifications and purposes; however, doing so goes beyond the scope of this project.

2.2.3 Simplification of force equations

As described in Section 2.2.2, a typical ATMA such as the one used in this research has vertically-oriented coils placed above the levitation workspace; and the design of the iron yoke channels the magnetic field lines such that they are overwhelmingly vertical in the air gap and relatively little energy is wasted. The $\vec{B}$-field being overwhelmingly vertical throughout the workspace means that a single levitated magnet will always be aligned vertically during operation. As such, Equation 2.23 can be simplified into:

$$\vec{F} = \nabla \left(\begin{bmatrix} 0 \\ 0 \\ P \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ B_z \end{bmatrix} \right) \ ; \tag{2.38}$$

that is,

$$F_x = P\frac{dB_z}{dx} \ , \tag{2.39}$$

$$F_y = P\frac{dB_z}{dy} \ , \tag{2.40}$$

$$F_z = P\frac{dB_z}{dz} \ ; \tag{2.41}$$

where P is the magnitude of the vertically-oriented magnetic dipole moment. For a permanent magnet, this can be written as:

$$\vec{P} = \left(\frac{\vec{B}_r}{\mu_0} \right) V \tag{2.42}$$

where V is the volume of the magnet (in m^3); and $\vec{B}_r$ is the remanence of the magnet (also known as residual magnetization or residual flux density, in T), a property of the magnetized material that is commonly specified by permanent magnet manufacturers [24]. For example, a neodymium (NdFeB) magnet of grade "N52" typically has a remanence value between 14500 and 14800 T [25].

With the ATMA largely set up as described above, assuming a power source that can energize the coils with a controlled current (which should be far more feasible now that the coils' field lines are channeled along the magnetic circuit and do not decrease with $\sim r^3$ from the coils), we are now able to produce a magnetic field capable of counteracting gravity for the magnet. However, since our end goal is to levitate the magnetic microrobot with 3 degrees of freedom (x, y, and z), there are two major obstacles remaining.

The first issue is that even with a coil current that produces a magnetic field such that the microrobot's gravitational force is perfectly counteracted, this is a state of unstable equilibrium, meaning that some form of closed-loop control is necessary to even levitate the microrobot at a constant height (let alone establish horizontal control). This will be elaborated upon in Chapter 3, along with the current state-of-the-art for establishing horizontal position control of the magnet.

The second issue has to do with the physical design of the ATMA as described up until this point: As seen in Equation 2.39, a polarized object such as a levitated microrobot will experience a force in the direction of increasing magnetic field strength. This means that, assuming that the microrobot is levitated at some fixed height using the aforementioned closed-loop control, its (uncontrolled) position on the horizontal plane will accelerate towards some point of maximum magnetic field strength (or minimum magnetic potential energy) on that horizontal plane; hereafter referred to as a B_{max} point. The issue is that, as seen in Figure 2.4, an individual coil has a clear axis along which field strength is maximized– meaning that each energized coil may produce a local minimum of potential energy directly underneath itself that the microrobot may fall into, which will complicate the matter of achieving position control of the microrobot. This issue is still present even with the iron yoke [26]. As such, the following section describes how this issue is solved at the hardware level, by making one final addition to the physical structure of the ATMA.

2.2.4 Pole Pieces

As described above, a component referred to as a "pole piece" is installed onto the ATMA to address the issue of multiple B_{max} points appearing in each horizontal plane over which the microrobot can be levitated. The purpose of installing the pole piece is to ensure that

one B_{max} position exists on the horizontal plane at any given time within the span of the workspace, while allowing enough leakage flux that the horizontal position of this B_{max} position to be shifted as needed.

Design of the pole piece utilizes the phenomenon by which magnetic "charges" tend to concentrate around sharp points and edges as electric charges do [27], whereas flat surfaces distribute these hypothetical charges more evenly. This means that with an array of coils connected to this pole piece, the exact shape and configuration of the pole piece determines how each of the coils' magnetic fields will be affected. For example, a pole piece consisting of a simple flat disk mounted directly to the underside of the coil array can have the effect of "merging" the coils' $\vec{B}$-field contributions such that only one B_{max} position appears on each horizontal plane, even when all of the coils are activated at once. However, must be emphasized that when the $\vec{B}$-field contributions are merged in this way, it is the leakage flux that is responsible for shifting the B_{max} point across each horizontal plane through uneven loading of the coils. Thus, it is important to ensure that the pole piece does not merge the coils' $\vec{B}$-field contributions so strongly that the leakage flux simply becomes negligible.

There is no known theoretical method for precisely determining the effect of the pole piece on the workspace $\vec{B}$-field [28][29]. As such, pole piece selection is typically done either experimentally or through simulations with finite element method (FEM) simulations [12]. For the purposes of 3-DoF control of a single levitated microrobot using a horizontal array of vertically-aligned coils (particularly when these coils are arranged side-by-side in a disk or ring formation), a simple flat disk-shaped pole piece is often sufficient [12] [26]. It is worth noting that depending on the desired application of the ATMA, other pole-piece designs (cross-shape, dome-shape, flat disk with a hole cut into the middle, etc.) may be worth considering as well [26].

The aforementioned empirical pole piece selection is typically done by measuring the magnetic field using a gaussmeter/magnetometer to verify that the desired criteria are met. However, assuming that vertical control (with equal coil current activation) can be established, it should be possible in principle to qualitatively evaluate the effects of the pole piece (without needing to perform simulations or $\vec{B}$-field measurements) by utilizing the random coil current perturbation algorithm described in Section 4.4. Continuously tracking the microrobot's horizontal position as the system inputs are randomly fluctuated can yield useful information about the effect of the pole piece on levitation in a semi-automated manner (e.g., by revealing regions that the microrobot can get "stuck" in, regions that the microrobot has difficulty accessing, etc.); although caveats to this technique may include needing to calibrate the feedforward and feedback control algorithms for vertical control with each pole piece, on top of this randomized motion taking a long amount of time before

movement patterns are revealed. (Refer to Section 3.2 for further details on establishing this vertical control.)

2.2.5 General Assembly of Levitation Apparatus

Section 2.1 described the physical principles governing magnetic levitation and introduced a set of corresponding design choices used in establishing an ATMA. However, this alone is not sufficient for establishing controlled attractive-type magnetic levitation, let alone with 3DoF control. The electromagnetic coils of course need a power source to provide electrical current sufficient for levitation; and as mentioned previously, the vertical position of a single permanent magnet experiencing an attractive magnetic force from above is inherently unstable (in the sense that any movement towards the magnetic field source will subsequently increase the attractive force, and any movement away from the magnetic field source will subsequently decrease the attractive force; meaning that a set of coils producing a constant magnetic field will at best provide an unstable equilibrium against gravity for the magnet). To keep the vertical position of a levitated microrobot controlled and stable with an ATMA, some form of feedback control is needed, and the currents applied to the coils (which are to be adjusted via controller, based on the current state and the desired state of the levitated magnet) must be adjusted quickly enough to reliably "correct" the microrobot's trajectory.

The simplest method of measuring the state of the levitated microrobot is to continuously track its position. In the context of micromanipulation and other small-scale applications, it is important to keep the position tracking measurements as precise and as responsive as possible; meaning that precision instruments such as laser sensors can be highly useful for this purpose. The raw position sensor readings can be sent to a controller for interpretation, which would then process the data as needed before updating the amount of current (or voltage) needed in each coil. Depending on the particular embodiment of the setup, the controller can then send this updated "input" information to the power source of the coils (such as an amplifier), which will generate an updated magnetic field that the microrobot will respond to accordingly.

Figure 2.6 provides a broad graphical summary of this aforementioned flow of information for a typical ATMA. While this description will be common for many ATMAs across various scales and desired applications, it is of course heavily simplified. The details regarding the vertical and horizontal ATMA control algorithms, as well as the current state-of-the-art for designing such control algorithms (especially for horizontal position control, which is far less straightforward than vertical position control) will be discussed

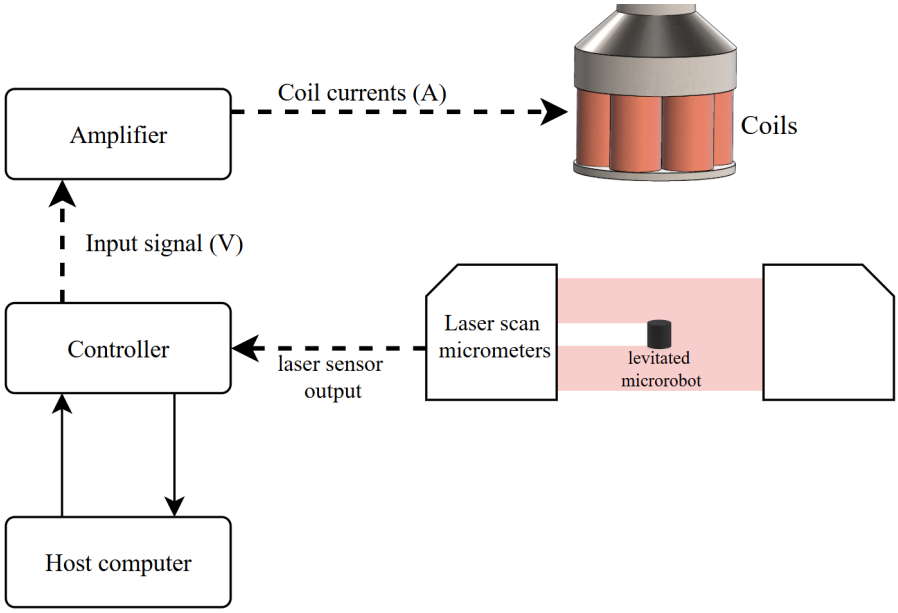


Figure 2.6: Flow of information during closed-loop control of levitation. Analog signals are indicated with dashed arrows

in some more detail in Section 3.1. In the meantime, the remainder of this section will be used to introduce the hardware and software configuration for the ATMA used in this research.

2.3 The Setup

Figure 2.7 shows a photograph of the actual ATMA configuration used in this research. For this setup, the passive components of the magnetic drive unit (i.e., the "magnetic circuit" that produces the magnetic force that drives the motion of the levitated magnet) are overwhelmingly composed of low-carbon stainless steel (with an assumed relative permeability of $\mu_r \approx 4000$ [4][12], hereafter referred to as "soft iron") due to its useful properties of high magnetic permeability and low coercivity. The magnetic circuit itself comprises a soft-iron yoke arranged in an approximate C-shape, and affixed to the upper end of the yoke's air gap are 6 pairs of thick cylindrical soft-iron-core solenoids arranged in a hexagonal shape (with each "pair" comprising one vertically-oriented solenoid stacked on top of another, hereafter referred to as a "coil" for simplicity). A single flat disk-shaped soft-iron pole piece

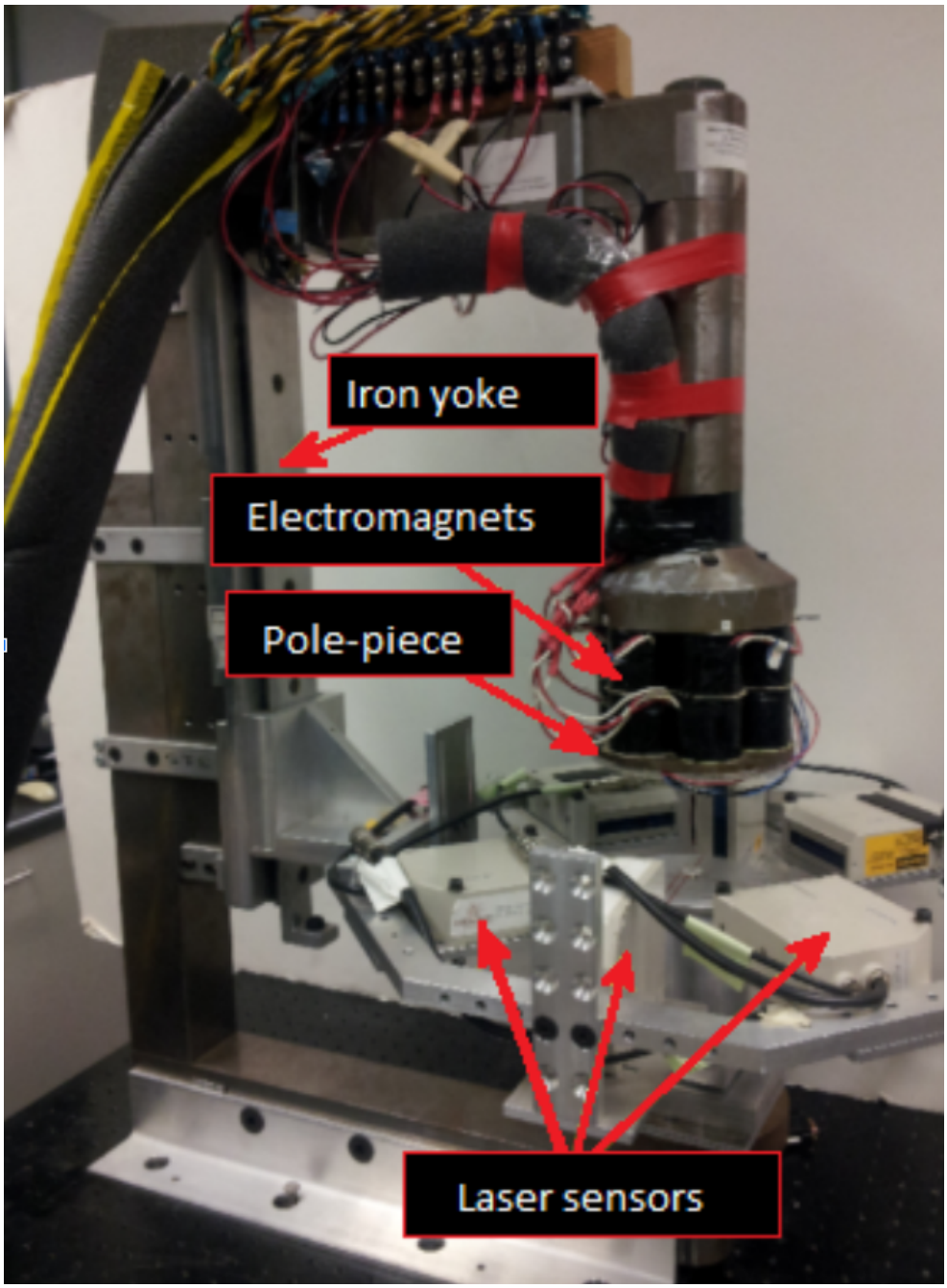


Figure 2.7: Photograph of the final ATMA configuration

is affixed to the bottom of the six coils. Figure 2.8a illustrates this overall configuration with more clarity.

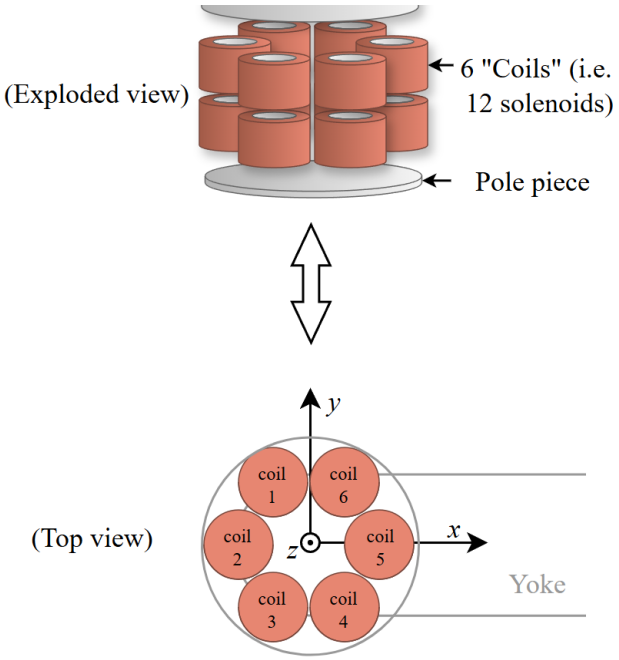
Each aforementioned solenoid is made of approximately 840 turns of #22 AWG copper wire, with an inner radius of 1cm, outer radius of 2cm, and length of 4cm; yielding a resistance of approximately 4Ω and an inductance of approximately 28mH (with slight variance among the solenoids as a result of manufacturing imperfections) [12]. The twelve solenoids are energized independently of each other (albeit controlled in pairs [4]) using a custom-built current amplifier [12], with two Sorensen DCS40-30E DC power supplies (each set to output at a constant 40V) as the power source. The amplifier behaves as a voltage-controlled current source, with the current flowing through each of the coils determined by the input voltage fed into the amplifier. The output current (in A) generated by the amplifier for each coil was previously measured to be:

$$i = 0.355v \tag{2.43}$$

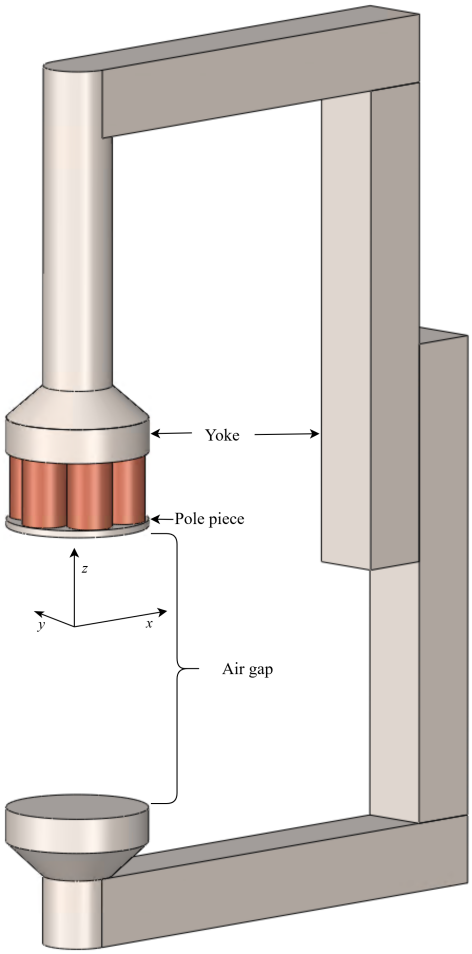
where v is the input voltage (in V) fed into the amplifier by the digital-to-analog board of the controller.

The controller itself comprises a dSPACE PX20 expansion box housing a dSPACE DS1006 processor board (which utilizes a 2.8 GHz quad-core AMD Opteron processor), connected to a DS2004 analog-to-digital (i.e., A/D) board and a DS2103 digital-to-analog (i.e., D/A) board. The DS2103 D/A board is connected to the input terminals of the aforementioned amplifier, and has a configurable output range of either $\pm$5V or $\pm$10V (set to the latter for our purposes– which, as per Equation 2.43, effectively leads to a maximum coil current of $\pm 3\,55$A). The DS2004 A/D board, on the other hand, is connected to the output terminals of a laser sensor configuration; and has a configurable input range of either $\pm$5V or $\pm$10V (set to the latter for our purposes, matching the voltage range of the laser sensors' output signal) [4]. The DS2004 A/D board has a signal conversion time of $0\,8\mu$s [30], and the DS2103 D/A board has a rated settling time of approximately 10μs [31].

The aforementioned laser sensor configuration is used for tracking the position of the microrobot, and outputs an analog voltage to the DS2004 A/D board based on the detected position. It comprises three KEYENCE LS-5041 high-speed laser scan micrometers (each of which is a transmitter/receiver pair, placed across the workspace from each other). Each of these three laser scan micrometers (hereafter referred to as one of three "laser sensors") has a $2\text{mm} - 40\text{mm}$ scanning range, operates at 1200Hz, and outputs an analog voltage between -10V and $+10$V corresponding to the detected position of the object. While the method by which the laser sensors detect position is technically configurable (as described



(a) Configuration of coils above the workspace



(b) Assembly of "magnetic circuit" for attractive-type levitation

Figure 2.8: Diagram of the final ATMA configuration

in [32]), the configuration used in this research and in its immediate predecessor [4] is such that in the presence of one or more objects obstructing the transmitter's laser beam, the obstruction with the *greatest height* relative to a vertically-oriented laser sensor (i.e., the lowest corresponding voltage) is what the sensor reports as its output voltage. Figure 2.9 provides an illustration of the output voltage determination in the case of multiple distinct objects that obstruct the transmitter beam. Note that the highest point of the laser beam obstruction, rather than the center of the highest distinct object, is what the output voltage corresponds to. It is also worth noting that these sensors are rated as having a measurement repeatability of $\pm 0.3\mu$m [33]; although in practice, position readings during operation have highly variable precisions based on where the obstruction is within the scanning range [4]. Additionally, possibly due to their age, the measurement precisions of the three sensors are observed to be somewhat inconsistent compared to each other.

An important consideration is the scenario of the upper edge of an object going beyond the upper boundary of a laser sensor (corresponding to -10V, or $+40$mm). When this occurs, the laser sensor's output voltage will *not* stay at -10V, but will instead switch its position determination mode to reporting a different edge *or* the distance between multiple edges, based on the number of distinct objects obstructing the beam. This new position determination mode will remain until the upper boundary is no longer obstructed. This sudden change in position determination criteria is notably marked by a sharp discontinuous "spike" in the sensor's output voltage before the new voltage is reported (as seen in Figure 2.10), even when an obstruction of moderate size passes this upper boundary at a very slow and steady speed. This combination of the automated shift in position determination criteria, the dependence on the presence and positions of other objects such as an adjustable holding platform, and the sudden (yet, notably, not instantaneous) signal "spike" preceding it makes it difficult to work around these cases in real-time applications. As such, when working with these laser sensors in the aforementioned configuration for maglev applications, the largest possible workspace for the levitated object is effectively restricted such that the object is contained *entirely* within the scanning area of all 3 laser sensors, without reaching any of their upper boundaries. An object passing any of these boundaries will lead to its position being misread, which will typically result in the levitation becoming unstable.

Finally, a dSPACE DS814 Link Board is installed onto the host computer to establish a wired connection to this controller. The host computer uses this connection to deploy a machine-readable control algorithm onto the controller through the dSPACE ControlDesk software (which also facilitates the creation of the user interface(s) to be used during operation). The control algorithms themselves were designed in a block-diagram environment using MATLAB's Simulink software, and converted into machine-readable code for

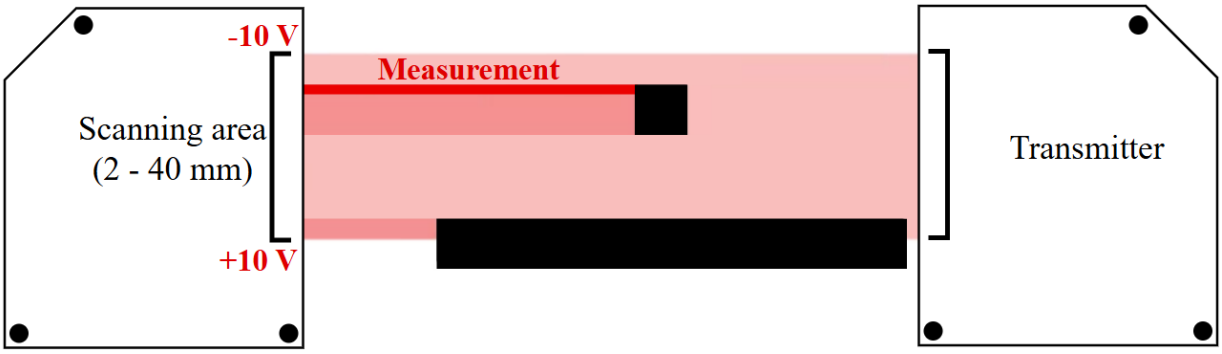


Figure 2.9: Laser sensor detection range for Keyence LS-5041

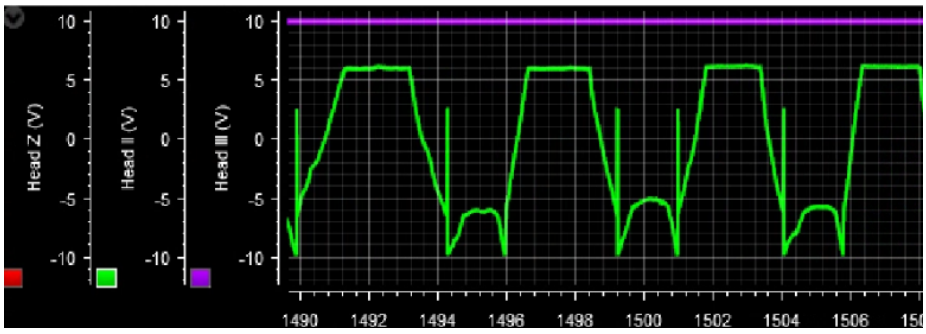


Figure 2.10: Sample recording of "spikes" in laser sensor readings when object exceeds detection range

ControlDesk using MATLAB's Simulink Coder and dSPACE's Real-Time Interface (RTI).

The details regarding the control algorithm itself will be introduced in Chapter 3, and will be expanded upon throughout the rest of this thesis.

2.3.1 Responsiveness of system

The dSPACE controller is capable of operating in real time at a rate of 1 kHz, meaning there is at minimum 1ms of delay in between receiving an input signal and sending a corresponding output to be handled by the plant. The ideal scenario is that when the magnetic field in the workspace is needed to behave in a particular way during operation, the controller would "request" a specific coil current from the amplifier to be sent through the coil(s), which would consistently result in a corresponding magnetic field configuration in the workspace within this 1ms timeframe. However, this "idealized" behaviour of instantaneous responsiveness and perfect correspondence between requested coil currents and final magnetic field is not possible in practice due to various hardware limitations of our setup; and these shortcomings must be accounted for in some way (e.g., in the final control algorithm(s)) for optimal performance. In terms of the relationship between coil currents and the actual magnetic field, it is important to note that this particular setup has yielded noticeable "hysteresis"-like behaviour during open-loop control of horizontal position [34]. Although the cause of this hysteresis effect has not yet been formally investigated, one possible cause may be non-negligible magnetic coercivity in the "soft-iron" material comprising the solenoid cores, pole piece, and yoke; which cause the material to try and "retain" some of its polarization when the magnetic field changes (and may be especially prevalent in the solenoid cores, which are exposed to the strongest magnetic fields as seen in Figure 2.5).

In terms of response time between the controller's "input" and the resulting magnetic field in the workspace (which drives the motion of the levitated magnet), there are additional limitations preventing this from occurring instantaneously. One reason for this is that the custom amplifier used in this configuration is a "semi-active linear amplifier" [12] that can only actively control increases in the coil currents, meaning that there is no active control happening when a current needs to be *decreased* instead. Decreases in coil current are instead subject to the rate at which electrical energy can naturally dissipate through a diode circuit in the amplifier [12], while also being subject to effects such as inductance in the coils. As such, a fully active amplifier that can actively control both increases and decreases to current flow may result in even better responsiveness.

To elaborate on the effects that inductance (e.g., in the soft-iron-core coils) can have

on the responsiveness of the setup: As mentioned in Section 2.1, Faraday's law and Lenz's law state that if a conductor is exposed to a time-varying external magnetic field (or a conductor generating such a field due to a time-varying current flowing through it), the changing $\vec{B}$-field will induce an electromotive force (or voltage), which in turn generates its own magnetic field such that this change in the initial $\vec{B}$-field is opposed. This phenomenon of electrical inductance can be thought of as a conductor's tendency to oppose changes in its own current.

To quantify this effect, we use the inductance L (measured in Henries, H), a property of a conductor's material properties and geometry, to describe the ratio between the induced voltage through the conductor $v(t)$ and the rate of change of the current I:

$$v(t) = L\frac{dI}{dt} \tag{2.44}$$

In the context of electromagnetic coils, the induction L in a simple air-core cylindrical solenoid can be calculated via:

$$L = \frac{N^2}{l}\mu_0 S \ , \tag{2.45}$$

where N is the number of turns, S is the average cross-sectional surface area of the solenoid's turns, l is the length of the solenoid, and μ_0 is the permeability in free space. (This calculation is less straightforward for iron-core solenoids such as the ones used in our ATMA; however, as mentioned in Section 2.3, the manufacturer rates each solenoid as having an inductance of approximately 28mH. Also worth noting is that each solenoid has approximately $R = 4\Omega$ resistance.)

The resulting time delay for changes in current through each solenoid can be characterized by the "inductive time constant" of a circuit:

$$\tau = \frac{L}{R} \ , \tag{2.46}$$

and in the context of a current driven by a potential difference that flows through the solenoid, the current over time t can be changed from some initial current I_1 to some new current I_2 as follows:

$$I(t) = I_2 + (I_1 - I_2)\, e^{-\frac{t}{\tau}} \ . \tag{2.47}$$

The solenoids used in our setup effectively have an inductive time constant of ~ 7ms, which means that (according to Equation 2.47) it takes around ~ 16ms for a voltage-driven current to be within 10% of its new value I_2 from its initial value I_1. This, of course, is far beyond the minimum 1ms time delay associated with our controller.

That being said, this time delay applies to currents flowing through the coil that are driven by a potential difference. The amplifier being used is a voltage-controlled current amplifier [12], which instead behaves closer to an ideal current source (which would force the new current I_2 through the solenoid with zero delay) than to an ideal voltage source (which would result in the $\sim$ 16ms delay described above, neglecting any additional delays from the source itself). However, the semi-active nature of the current amplifier means that only increases in current are forced in this way; whereas *decreases* in current are still subject to time-delaying effects from coil induction.

The end result during closed-loop control is that the magnetic field in the workspace responds much more slowly to requested decreases in current than it responds to requested increases in current. The exact nature of the solenoid inductance-associated time delays in cases of rising/falling current, and how this interacts with the delays present in other components of the system (such as the signal conversion times of the D/A and A/D boards described in Section 2.3), falls beyond the scope of this work (see [12] and [34] for additional details). The most important consequences of these time delays for the purposes of this research include the facts that levitation dynamics can significantly differ between upward and downward motions (and during sudden changes in horizontal position), and that the final control model may require special considerations (such as distinct "P" gains for upwards versus downwards motions for a vertical PID controller) to alleviate these limitations as necessary.

Chapter 3

Control of the Microrobot

3.1 Current State-of-the-Art

Throughout Chapter 1.1, the state of the art for establishing 3-DoF position control for an ATMA was mentioned multiple times, primarily in the context of the difficulty involved in this process. This section aims to summarize the current state of the art by which 3-DoF position control is established, with a particular focus on the most recent "predecessor" to this research seen in [4].

3.1.1 Magnetic Force Approximation

In very early predecessors to this research, horizontal position control of the ATMA was established by modelling the horizontal magnetic force applied to a single magnet in the workspace, using a combination of finite element model (FEM) simulations and approximations of the magnetic force that a single coil would yield [35]. While this method of approximating the force contribution of each coil did show some (tenuous) agreement with simulations, the model notably assumed simple superposition between each coil's magnetic field contributions and was unable to account for the presence of the yoke or pole piece, leading to limitations in final controller performance [34].

A more recent study with emphasis on the horizontal levitation dynamics of this ATMA was performed and presented in [4], marking the state of the art for establishing horizontal position control of an ATMA-levitated microrobot (as of the time of writing).

Due to the studies presented in [35] and [4] being conducted on the same base magnetic circuit as the one used in this research, the general construction and assembly of the ATMA is largely the same as was described in Section 2.3. In [4], position of the levitated microrobot was measured in real time using laser scan micrometers; and an A/D board connecting these laser sensors to the dSPACE box were used to communicate these sensor readings to the controller, from which the next set of input coil currents could be determined in real time.

As seen in Section 2.2.4, unevenly loaded coil currents during stable vertical levitation will result in a shifted B_{max} position that the magnetic microrobot will accelerate towards. However, because three are multiple coils being employed simultaneously, efficient horizontal position control requires a model that not only understands the effect that each coil has on the microrobot's motion, but the effect of the overall construction (i.e., complexities introduced by the yoke, iron coil cores, and pole piece that make the workspace magnetic fields impossible to model analytically) on the magnetic microrobot as well.

This was performed using empirical measurements of the $\vec{B}$-fields associated with each coil, combined with analytical calculations that aimed to summarize the patterns observed in these empirical measurements. The process began with the assumption that, based on the symmetric hexagonal arrangement of the identical coils above the workspace, the coils would each have an equal effect on the vertical magnetic force. Empirical gaussmeter measurements of the $\vec{B}$-field associated with each coil $\mathcal{C}$ revealed that the magnitude of each $\vec{B}_{\mathcal{C}}$ varied parabolically with the x-y axes across each horizontal plane, albeit with some asymmetry between the x and y axes due to the presence of the iron yoke (although symmetry could still be assumed about the yoke axis to reduce the number of variables in the $\vec{B}$-field model). Based on the knowledge that the $\vec{B}$-field associated with each coil (and, thus, the 3D magnetic force) was proportional to the current flowing through that coil, the presented method went on to analytically approximate the total $\vec{B}$-field in the workspace as the sum of each coil's (parabolically-varying) contribution. This approximate analytical form of the $\vec{B}$-field was simplified further with additional symmetry assumptions, and then linearized about the center of the workspace under the assumption that the navigation range of the microrobot would be relatively constrained (at least, compared to the horizontal span of the hexagonal coil array). Finally, the gradient of this $\vec{B}$-field model was calculated, which (as per Equation 2.39) revealed an (albeit unparameterized) analytical equation for the magnetic force in 3 dimensions.

As a brief summary of this process, the most significant output of this intensive calculation was a simplified analytical model which revealed the relative effect of each coil current on the 3D magnetic force experienced by a levitated magnet. Following the coil

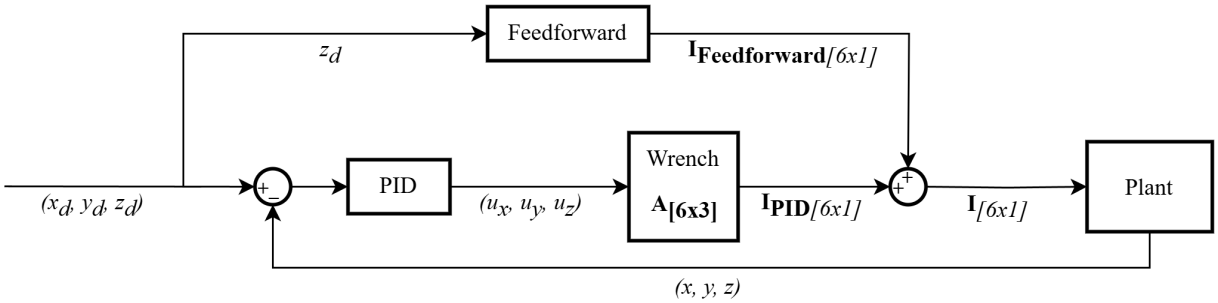


Figure 3.1: Current SOTA of 3DOF position control; see Equation 3.1 for $\mathbf{A}_{\text{SOTA}}$ (Adapted from [4])

numbering scheme and the 3D coordinate system defined in Figure 2.8, these levitation dynamics were summarized using the wrench matrix:

$$\mathbf{A}_{\text{SOTA}} = \begin{bmatrix} -0.0833 & 0.2500 & 0.1667 \\ -0.1667 & 0 & 0.1667 \\ -0.0833 & -0.2500 & 0.1667 \\ 0.0833 & -0.2500 & 0.1667 \\ 0.1667 & 0 & 0.1667 \\ 0.0833 & 0.2500 & 0.1667 \end{bmatrix}, \tag{3.1}$$

a 6×3 matrix whose columns represent the relative magnetic force applied to the microrobot by each coil's current in the x, y, and z directions, respectively. Essentially, if the microrobot were levitated in place with a fixed ratio of currents between the six coils (hereafter referred to as the coil current ratio, or CCR), this "wrench matrix" represented a direct mapping between the CCR and the resulting horizontal magnetic force applied to the magnet; and thus a 3-DoF closed-loop control model could be implemented as seen in Figure 3.1. (Note that this is an exceedingly oversimplified summary of the process by which this wrench matrix was generated; please refer to [4] for the full details.)

Note that the control algorithm seen in Figure 3.1 contains a "feedforward" component in addition to the feedback loop. This is specifically implemented to assist the vertical control of the magnet; and was initially found by analyzing the transfer function between the measured $\vec{B}$-field and the input voltage of the controller during the frequency response identification process of the ATMA [12]. However, as will be discussed in Section 3.2, this frequency response identification step is not strictly necessary; as the feedforward component of the vertical position control algorithm can be established more intuitively via empirical calibration.

3.1.2 Alternate Embodiments of the State-of-the-Art

It is worth noting that the state-of-the-art for establishing 3-DoF position control involves usage of 3 laser sensors simultaneously to determine the 3D position of the magnet. However, for applications where the x-y sensors cannot be used (e.g., due to optical blockage), it has been shown [36] that Hall-effect sensors can be arranged on the pole-piece such that their $\vec{B}$-field measurements during levitation can be directly associated with an x-y $\mathrm{B_{max}}$ position that the microrobot will accelerate towards. This effectively allows for 3-DoF position control even when only using a single (vertical) laser sensor, albeit with less x-y accuracy than if laser sensors were used to measure position on all three axes. An important caveat to this method is that it also only determines the $\mathrm{B_{max}}$ position itself, rather than the actual microrobot position. This means that horizontal position control can only be achieved under the assumption that the microrobot is actually moving to the $\mathrm{B_{max}}$ point without obstruction. This still offers many applications in navigation through fluids and other media that may be optically blocked [37], although the lack of feedback of the exact position of the microrobot makes force control less feasible in these applications.

3.1.3 Signal Processing During Levitation

With the fundamental 3-DoF position control algorithm established as described in Section 3.1.1, additional improvements to performance could be made through a combination of analog and digital filtering of the laser sensor signals.

As described in Section 2.3, the laser sensors have noise in their measurements that must be accounted for, especially for the "derivative" component of a PID controller (as fluctuations due to noise would otherwise be overwhelming compared to actual recorded motions of a levitated object). Figure 3.2 shows a typical unprocessed laser sensor recording of the microrobot resting motionlessly on its holding platform. While the measured fluctuations are only on the scale of μm, they are still significant enough to warrant smoothing the position signal through filtering.

Methods of filtering out the noise in these data include analog and digital filters, which have some practical differences that should be accounted for [38]. In [4], an analog lowpass filter was applied directly to the analog laser sensor signals before being transmitted to the controller; and a virtual 6th-order Butterworth filter with a normalized cutoff frequency of 0.6 was applied to the laser sensor signal prior to being fed into the PID controller.

The signal filtering methods were slightly altered for the research presented in this thesis. The analog low-pass filters were removed due to introduction of inconsistent behaviour

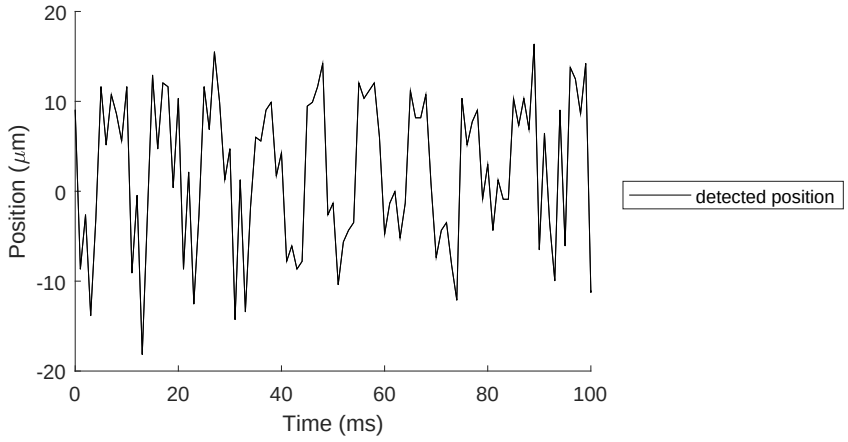


Figure 3.2: Sample laser scan micrometer readings of a perfectly stationary object

from degraded wire/contact quality in the years since installation; and this increase to the general number of potential "points of failure", as well as the inability to customize the parameters of these filters, led to the decision that keeping them installed was suboptimal.

Note that it is essentially impossible to implement a real-time digital filter with precisely zero phase distortion or zero group delay. Experimentation with a variety of digital filters revealed the introduction of significant delay (e.g., $\sim$5 ms for the aforementioned Butterworth filter) to the laser sensor signals, as well as a slight noise reduction in exchange for drastically worsened noise when a measurement was altered discontinuously (e.g., due to the microrobot's position being misread due to its surface reflectivity; see Section 5.1 for details). As a result of this, as well as the general lack of sensitivity to noise, the signals fed into the "position" and "integral" components of the PID controller had all signal filters removed so as to maximize responsiveness and eliminate the issues that were introduced by digital filters. In contrast, the "derivative" signal is significantly more susceptible to noise while being less immediately sensitive to ms-scale delays; as such, the aforementioned Butterworth filter was retained, in addition to a 10ms moving mean filter for additional smoothing (in time with the 10ms periodicity of the noise, as seen in Figure 3.2). This was found experimentally to strike the best balance between smoothing (for precise "derivative" signal calculations) with minimal loss of responsiveness (for the "position" and "integral" signals), leading to the best performance.

3.1.4 Limitations and Proposed Solutions

The state-of-the-art method presented in [4] was successfully able to establish 3-DoF position control with enhanced performance over its predecessor studies. However, this methodology still has several prominent caveats that leave room for improvement.

Firstly, the analytical calculations were performed with many simplifying assumptions, which were based either on the assumption that the levitation would be largely constrained to the center of the workspace (in which case the analytical model would not necessarily describe the levitation dynamics outside that region), or on symmetries specific to the hexagonal coil configuration used in our ATMA (which would not be applicable to other configurations whose coils are not arranged in an identical manner). This limits not only the accuracy of the final control model across the workspace, but also the applicability of this method to other similarly-structured ATMAs with different coil configurations.

Secondly, the wrench matrix was constructed based on this simplified analytical calculation of the magnetic force equation; however, notably, introducing a shift to the CCR during levitation does *not* result in a constant horizontal force being applied in practice. Instead, as mentioned previously, the position of the B_{max} point changes such that the microrobot will accelerate towards it before eventually settling. This indicates that the wrench matrix is most accurate to levitation in the very center of the workspace, but is less applicable outside of that range. (However, it is important to note that due to the approximately parabolic nature of the magnetic field in the workspace, this does lead to a convenient corollary by which the magnetic force– which, as a reminder, is proportional to the gradient of the magnetic field as per Equation 2.39– is approximately linearly proportional to the horizontal distance between the microrobot and its target B_{max} point [4][36][39]. This corollary is key to the success of the PID algorithm, both in [4] and in the novel machine learning-assisted method presented in this thesis.)

Additionally, despite the overall difficulty and effort required in performing these analytical magnetic field approximations (which poses an accessibility barrier in terms of difficulty, time investment, prerequisite knowledge, and/or financial cost), the process described above produces only **one** wrench matrix. If performed successfully, this matrix aptly describes the levitation dynamics associated with the centermost subregion of the workspace; however, it does not necessarily apply beyond that range. This can lead to a consistent decrease in PID controller performance as the levitated microrobot approaches the outer boundaries of the workspace, as was observed in [4].

Furthermore, this process generates a (single) wrench matrix for that particular ATMA configuration. However, if the configuration is to be altered in any way (e.g., through

addition or displacement of coils; through alterations made to the pole piece; etc.), the entire process of analytically obtaining the magnetic force model must effectively be redone.

If the researcher's goal is to avoid having to do analytical math, a conceptually straightforward alternative to this method (in principle) may be to empirically obtain a detailed model of the $\vec{B}$-field produced in 3D space by each coil individually (as opposed to an unparameterized model that does not directly utilize large quantities of $\vec{B}$-field measurements). This can either be done through numerous empirical Gaussmeter measurements, or through FEM simulations. However, performing the empirical measurements will still require highly precise specialized equipment beyond what is directly used in the levitation (such as a Gaussmeter for measuring the $\vec{B}$-field along each axis, and a robot arm for holding this Gaussmeter), and will be very time-consuming to repeat the measurements if the configuration of the ATMA is altered as described above. Additionally, FEM simulations may require surmounting a prohibitively steep learning curve for experts in different fields of study. The setup must be accurately constructed within the FEM software tool, and manufacturing imperfections in the actual components often cannot realistically be accounted for. Furthermore, the necessary software may be prohibitively expensive to access.

An overarching goal of this research project is to minimize the amount of time, effort, and prerequisite knowledge needed to create a 3-DoF control model, by automating as much of this process as possible. This is for the purpose of improving ease of access for experts across different fields to establish (or re-establish) this 3-DoF control after making custom modifications to a maglev setup (such as additional coils, different coil placement, etc.) for their own purposes, or even after building an ATMA from scratch.

In an ideal scenario, a method could employ machine learning to essentially "bypass" the overwhelming majority of this process, by observing and subsequently learning to predict how the microrobot moves in response to system inputs. It is in principle possible to generate this training data in a fully automated manner, by recording randomly-varied system inputs (i.e., coil currents) and the resulting trajectory of the microrobot as they vary over time; and these recordings can be processed to form training data for a large variety of possible ML implementations. (Of course, Chapters 4 and 5 detail the process for doing just that.)

For the task of enabling horizontal control in an ATMA (be it position control or force control), perhaps the most naively straightforward method of achieving this using a machine learning model (given a sufficient quantity of training data that can showcase the relationship between system inputs and system outputs) is to simply train a neural network (NN) to directly associate a system input with the output behaviour that the researcher

seeks to control. Of course, this is much easier said than done, and the feasibility of training a neural network in this way is further complicated by the hysteresis effect between coil currents and the actual $\vec{B}$-field in the workspace.

For the sake of illustrating the variety of methods that may be viable *in principle*, applications could potentially include input/output pairings such as:

- associating the input CCR and initial microrobot velocity with the resulting trajectory of the microrobot (which may require a NN architecture that can properly "understand" sequential, time-series data)
- associating the input CCR with the resulting final "B_{max}" position of the microrobot (which may be doable with simple and fast NN architectures, but treats the entire trajectory associated with a CCR as a single data point and will fail to account for complexities such as the hysteresis effect)
- associating the input CCR and position of the microrobot with its instantaneous acceleration (which, if successful, could allow simple NN architectures to utilize each step in time in the dataset as a unique data point– but requires precise calculation of acceleration curves for the full dataset, and may still fail to account for complexities such as the hysteresis effect)
- associating changes in the input CCR with the corresponding change in position of the microrobot (which has low dimensionality and can take advantage of the assumption of linear proportionality between horizontal magnetic force and the microrobot's distance from the B_{max} point to generate a wrench matrix for later use)

Certain methods of utilizing the training data may require a ML model that can process sequential (e.g., time-series) data, which traditional NNs such as shallow or deep neural networks are fundamentally less suited for. While less straightforward in both concept and implementation, dynamic neural networks are capable of interpreting these sequential data [40]; meaning that they may be capable of utilizing the sequential nature of recorded microrobot trajectory data to predict the microrobot's motions in response to an input CCR (or perhaps predict a CCR that would yield some desired motion) with greater accuracy and consistency. Additionally, dynamic networks may be particularly suitable for observing and quantifying the effects of hysteresis on the relationship between CCR and B_{max} position; and a dynamic network that properly accounts for this could even allow for fully open-loop horizontal position control (in an obstruction-free environment) without the need for active horizontal position tracking during operation.

While this research (particularly, in Chapters 5 and 6) focuses on a particular method of implementing ML that showed a great degree of success, associating inputs and outputs as described above could potentially be used to enable tasks such as:

- "Direct force control": Neural network that can directly output the CCR for a desired (input) horizontal acceleration (by relating CCR and microrobot position to resulting acceleration, with each point in time acting as a data point)
- "Direct position control": Neural network that can directly output the CCR for a desired (input) horizontal B_{max} position (by relating CCR and initial microrobot position to final microrobot position; with each "step" of movement associated with a new CCR producing a single data point)
- "Open-loop direct position control": Neural network that can directly output the CCR required for a desired (input) horizontal B_{max} position (by relating CCR to final microrobot position, with hysteresis effects ignored for the sake of a simpler model due to lower dimensionality)
- "AI-assisted position control": Relating change in CCR to change in microrobot position (as an approximation, but using this to generate a wrench matrix)

Of course, all of these potential methods require training data. The process by which the necessary training dataset is procedurally generated is detailed in Chapter 4. The remainder of this chapter will focus on establishing vertical position control (which does not require any math, $\vec{B}$-field measurements, modelling, or indeed anything beyond the hardware and software components comprising the ATMA itself) as a prerequisite step to this automated data generation.

3.2 Modified Control Algorithm

This section describes how 1D vertical position control of the levitated microrobot is established, as a prerequisite to both the generation and usage of training data described in Chapters 4 and 5.

A relatively "safe" assumption applied in the state-of-the-art methodology for establishing the magnetic force model is the postulation that each coil contributes equally to the vertical magnetic force applied to the microrobot. This assumption is used here as well, as it keeps initialization of stable levitation relatively simple.

Overall, vertical control and subsequent stable levitation of the microrobot can actually be established with relative ease, and nearly all of this process can be achieved through empirical calibration. When the microrobot is placed on the holding platform in the workspace, the coil currents can all be slowly and simultaneously increased until the vertical magnetic force is sufficient to counteract gravity (which will be marked by the magnet simply jumping up and sticking to the pole piece). As such, calibration of vertical levitation begins by first observing the approximate "threshold" coil currents beyond which the magnet leaves the holding platform.

With this in mind, the overall process to initially establish stable levitation can be performed as follows:

1. Place the magnet on the platform and make sure its vertical position is continuously measurable by the position sensors.
2. Increase coil currents until the magnet suddenly jumps up and sticks to the coils.
3. Use a PID controller (as seen in Figure 3.3) to continually alter the coil currents so that the magnet moves to some desired (constant) levitation height.
4. Tune the PID gains such that the magnet consistently converges to the desired height without its levitation becoming unstable. For those unfamiliar, there are many existing tools and resources (e.g., [41]) that exist to assist in this step. However, a common way of performing this step manually is to:
   (a) Increase the "P" gain until magnet starts oscillating
   (b) Increase the "D" gain until the magnet stabilizes and converges onto a point with minimal/no oscillation
   (c) Increase the "I" gain to eliminate any steady-state error
   (d) Using these three values as a starting point, alter these gains as necessary until satisfactory behaviour is achieved

Once vertical feedback control is initially established, an important step to improving the quantity and quantity of automatically-generated training data (as per Chapter 4) is to improve the response time and overall robustness of this vertical control. One part of this process is to establish the feedforward component of the vertical position control algorithm. Simply put, the purpose of the feedforward algorithm is to automatically offset the gravitational force so as to reduce the control effort demanded of the PID loop. With the PID

controller already established, this can be implemented by simply levitating the magnet at a variety of heights in the workspace, and recording the average coil currents (or amplifier voltage inputs) associated with stable levitation at each height. For example, one might measure that stable levitation at some observed heights $z \in \{1.0, 2.0, 3.0, 4.0, 5.0\}$mm draws average coil currents of $I_{\text{Feedforward}} \in \{0.831, 0.813, 0.796, 0.780, 0.764, 0.749\}$A, respectively. Measuring several such values across the z-range of the workspace, and interpolating between them (even if linearly) based on the z-position of the microrobot, is essentially all that initialization of the feedforward controller entails. This feedforward component is incorporated into the vertical position controller as seen in Figure 3.3.

One consideration that should be made, however, is that after establishing this feedforward component of the controller, the feedforward signal should be decreased slightly when the PID loop is turned off. This is so that the microrobot consistently falls back down onto the holding platform, as opposed to having a $\sim 50\%$ chance to jump up to the coils instead. Additionally, note that if other magnets are to be used, it is helpful to ensure that the feedforward coil currents scale proportionally with microrobot mass (as the gravitational force is proportional to the $\vec{B}$-field required to offset it, which in turn is proportional to the total coil current), as well as having the feedforward coil currents scale inverse-proportionally with the microrobot's magnetic polarization (as per Equation 2.42). This will allow the feedforward algorithm to apply to different magnets (as long as their masses, volumes, and polarizations are known) without needing to be re-calibrated for each one.

Finally, just as the feedforward controller was tuned to account for differing levitation dynamics at different heights, it may also be necessary to dynamically adjust the PID gains in a similar fashion for optimum performance at all heights in the workspace. For example, the optimal PID gains for levitating the microrobot 2mm above the platform might be different from the optimal PID gains for levitation 10mm above the platform; as such, implementing a PID "gain scheduling" algorithm (in which PID gains are tuned under various operating conditions, and interpolated between based on the state of the system) may be crucial for keeping levitation stability fully robust at all possible heights in the available workspace.

From here, vertical control has been established and is highly robust. Possible machine learning-related methods for handling horizontal control have been introduced, but there is of course one absolute requirement to experimenting with any of these methods: training data. The next chapter will delve into the full details of how the training dataset is procedurally generated using the actual hardware.

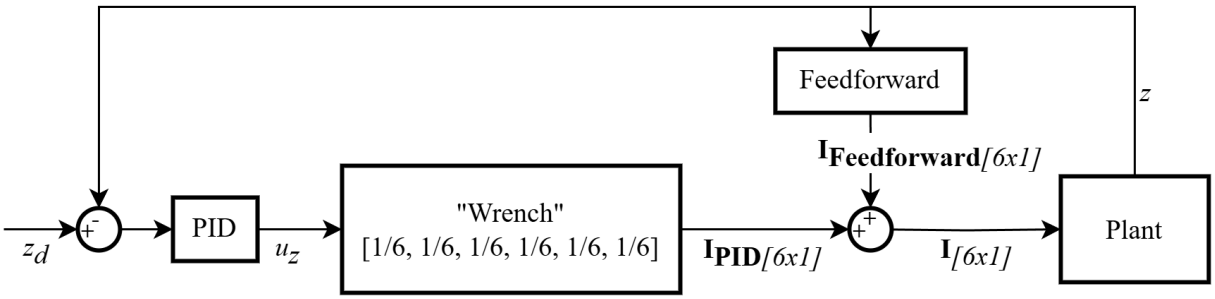


Figure 3.3: Feedforward + feedback vertical position controller, used in the modified algorithm

Chapter 4

Automated Training Data Generation

Up until this point in the thesis, we have discussed the current state of the art for establishing both vertical and horizontal control in a typical ATMA. While vertical position control is relatively straightforward to establish, horizontal control is a different matter entirely (see Section 3.2); and the combined amount of analytical modelling, simulation, empirical measurement, time, and/or cost associated with this particular step in the process can form an accessibility barrier for researchers and professionals outside this domain of expertise. To reiterate the primary motivation of this research, being able to automate this step is the overall goal that the next several chapters will focus on.

Establishing horizontal control is a step that can in principle be automated using machine learning, with one such method being presented in Chapter 5 of this thesis. The overall purpose of a successful ML implementation, in the context of this thesis, would be to "bypass" the steps of needing to manually model the magnetic field contributed by each coil throughout the workspace (whether analytically, empirically, and/or computationally). There are of course numerous possible ML algorithms that may be suitable for such a task, although all such algorithms share the same crucial requirement: training data. As such, with accessibility being a primary goal, this chapter presents an automated algorithm by which the requisite training data for such an ML implementation can be (and have been) generated with little to no need for human intervention.

When it comes to automating parts of a particular task using machine learning, one of course must first ask: Exactly what kind of task will the potential ML model(s) perform? What kind of ML architectures could be suitable for such a task? What kind of data would

need to be recorded to train this ML model, and how difficult and/or time-consuming would it be to obtain this data?

In the context of using ML to "bypass" the most complicated and time-consuming steps of establishing horizontal control for a single-magnet ATMA, we start by taking advantage of the fact that vertical control of the levitated magnetic microrobot is fairly simple to establish and calibrate as seen in Section 3.2. With this as a starting point, horizontal magnetic forces can be applied to the microrobot by unevenly loading the currents running through each of the coils so as to shift the B_{max} point that the microrobot accelerates towards (as described in Section 2.2), and the sum of the coils' currents is ideally adjusted by the vertical position controller in order to maintain a stable levitation height.

To very briefly touch on two particularly relevant classes of ML methods that may be used, shallow neural networks (NNs) are relatively fast and simple architectures that are particularly well suited for learning simple, non-sequential relationships between some inputs and some outputs; and for applications where it is imperative to properly account for sequential and/or temporal relationships between data (e.g., when utilizing the time-series nature of a levitated microrobot's recorded trajectory), dynamic neural networks may be a more suitable option [40]. There are of course numerous other ML methods that may be usable for a similar range of tasks (e.g., reinforcement learning models which may "learn" how the state of the system responds to specific inputs through a process of trial and error), although they will not be discussed in detail in this thesis.

Because of the variety of data types that can potentially be utilized for establishing some form of horizontal control with ML, a researcher aiming to experiment with a variety of different NN architectures in this context may save a lot of time by accounting for this variety of data during its initial (empirical) generation. In the context of this work, varying the system inputs and simply recording the microrobot's resulting trajectory will of course yield positional data of the microrobot; but with proper processing, may potentially be usable for calculating the acceleration of the microrobot varying with time in response to the inputs.

To elaborate on this: The laser sensors used for tracking the microrobot's position will continually output the microrobot's position as it varies with time, at a rate of 1000 Hz. These laser sensor readings are inherently noisy, meaning that the noise must of course be filtered out– but filters that are highly effective at removing noise (e.g., low-pass filters with very low cutoff frequencies) will also have the effect of filtering out actual sudden movements of the microrobot in cases of high acceleration. Data imprecisions due to laser sensor noise are relatively constant in the context of speeds that a levitated microrobot can realistically achieve during operation; meaning that if accelerations of the microrobot

are to be considered in ML implementations, higher microrobot accelerations are preferred during training data generation so that the (constant) noise in position data has less of an impact in comparison to the true motions of the magnet.

Additionally, as described in Section 2.2.4, a properly assembled ATMA will have *one* position of minimum magnetic potential energy / maximum magnetic field on each horizontal plane (referred to throughout this work as a B_{max} point) that depends on the relative current load in each coil, which the microrobot will accelerate towards and eventually settle onto. As such, for training ML models to find patterns associated with changes in position, **it is useful to introduce random CCR perturbations one at a time and wait until the final microrobot position is known** before proceeding. This also allows position data to be processed into a relatively simple, lightweight format (i.e., a single vector associated with position, with a corresponding vector associated with coil currents) that can be handled by simpler (i.e., faster) NN architectures.

Finally, an algorithm that generates training data without the need for human intervention by automatically operating the plant must be able to handle potential system failures during this process. One important consideration is that certain input CCRs may result in levitation of the microrobot becoming unstable, during which the microrobot may leave the workspace entirely. The algorithm must be able to identify when this occurs and automatically "recover" the microrobot accordingly, for the sake of minimizing the need for human oversight during the process. Another important consideration (which is less immediately tied to the system's response to changing inputs) is that depending on construction and usage scenarios, coils may accrue heat over time. This can cause changes in their electromagnetic behaviour that may be difficult to model, or perhaps even to the point that damage to the hardware occurs, which may necessitate automated "cooldown" periods during the data generation process. Similarly, hardware limitations such as maximum current that the amplifier can generate, maximum detection range of the position sensors, etc. should also be accounted for so as to prevent damage.

The following subsection details the actual methodology for the automated generation of training data.

4.1 Initial Calibrations

As prerequisites for automated data generation, the algorithm assumes that vertical control of a levitated magnet has already been established and calibrated (e.g., as in Section 3.2), and that the closed-loop control algorithm is sufficiently robust that sudden changes in

magnet height can be introduced without the levitation becoming unstable. This is so that the algorithm can introduce perturbations to the coil current ratio (CCR) with minimal risk of the levitation destabilizing and/or leaving the workspace.

While this calibration step is not strictly necessary, the automated data generation algorithm presented in this section records the position of the magnetic microrobot (in units of mm) based on the following calibrations. The laser sensors measure position using the furthest edge of the detected object (see Section 2.3 for details), meaning that the height and diameter of the (assumed cylindrical) microrobot must be accounted for so that the x-y laser sensor readings accurately output the position of the center of the microrobot, while the z-sensor readings accurately track the height of the microrobot above its holding platform (on which it rests when the coils are turned off). The 3D coordinate system (i.e., x, y, and z) itself is oriented as seen in Figure 2.8. Vertical position z is defined as 0 when the microrobot is at rest on the platform, while the y-axis is centered on the microrobot's position when it is levitated with equal coil currents (i.e., a CCR of $[1, 1, 1, 1, 1, 1]$) at the desired levitation height (for which the data are generated) above the holding platform. This calibration can be done with x as well; however, due to the narrow beam of the vertical laser sensor (which is aimed parallel to the y-axis), the x-axis in this work was instead defined such that $x = 0$ along the beam itself. These coordinates should be kept consistent between recordings, including if subsequent datasets are to be recorded at different heights (since the horizontal B_{max} position of the microrobot will vary with levitation height even with equal coil current ratios, as a result of the asymmetry of the shape of the ATMA).

In addition to this coordinate system defining the microrobot position, other preemptive "calibrations" include ranges associated with the laser sensors' readings of the microrobot at rest on the holding platform, the holding platform in the absence of the microrobot, and the like; as well as the x, y laser sensors' readings when the microrobot is about to leave the detection range of the vertically oriented z-sensor (at which point the levitation will become unstable because the microrobot's height cannot be observed). These are accounted for in the algorithm, in the sense that it can automatically detect when the workspace is empty, when the microrobot is resting on the platform, or when the microrobot is dangerously close to leaving the detection range of a laser sensor.

4.2 Initial Hardware Considerations

Prior to levitation, there are some final considerations as for the physical configuration of the laser sensors and the workspace that should be taken into account.

As will be seen in Section 4.4, the training data in the current embodiment will introduce perturbations to the coil currents one at a time, and wait for the levitated microrobot to settle on its new B_{max} point before the algorithm changes the CCR again. As seen in Chapter 3, the horizontal magnetic force applied to the microrobot is linearly proportional to its horizontal distance from the B_{max} point; meaning that in the absence of any significant braking forces, it will simply oscillate around this B_{max} point [4][17][39] and generally take a long time for its periodic motions to decay sufficiently. As such, an aluminum block is used as the holding platform, due to aluminum's properties of being nonmagnetic but highly conductive (meaning that the changing magnetic field caused by the microrobot's motion near the aluminum will induce eddy currents, which in turn generate their own magnetic fields that slow down the motion of the magnet). This allows the aluminum platform to effectively serve as a magnetic brake for the levitated microrobot, which allows its horizontal motions to be damped such that it converges onto each new B_{max} point more quickly and smoothly. This braking effect is of course dependent on both the material and geometry of the platform, and the amount of braking introduced during operation can be adjusted by altering the placement of the platform (or, similarly, the selected levitation height of the microrobot relative to the platform).

Note that if a magnetic brake is applied in this way, the final position of the microrobot associated with each new CCR will not significantly change; however, the trajectory that the microrobot follows to reach this final B_{max} point *will* change significantly, and this must be accounted for if (for example) the microrobot's accelerations are to be calculated from these trajectories for the purpose of calculating the applied magnetic force (refer to [42] for an empirically-verified analytical model of the braking effects caused by the aluminum platform for our particular setup). Additionally, note that alternate embodiments of this data-generation algorithm may be able to negate the need for a magnetic braking platform by observing the microrobot's trajectory (associated with each new CCR) in real time and calculating the point on the x, y plane around which the microrobot is oscillating. As will be discussed in Section 5.4, the horizontal motions of the levitated microrobot would behave very similarly to a simple harmonic oscillator with little damping, which follows a relatively simple mathematical formula and therefore would make it significantly easier to calculate accelerations (and, therefore, magnetic forces) directly from measured data.

Additionally, an important consideration to make for this algorithm is that upon vertical levitation becoming unstable, the microrobot will often move upwards uncontrollably and "stick" to the bottom of the pole piece, requiring human intervention to recover. Because the main idea of this algorithm is to minimize the need for human interaction, it may help to place a soft nonmagnetic, electrically insulating barrier (such as a wide sheet of packing foam) just above the workspace. This is to serve the purpose of preventing the microrobot

from sticking to the pole piece if it moves beyond the upper boundaries of the workspace, and allowing it to simply fall back down onto the holding platform when the coils are turned off (e.g., through an "emergency stop" switch coded into the main control loop).

4.3 Initial Software Considerations

As mentioned in Section 2.3, the main control loop for the setup is programmed via Simulink and deployed onto the dSPACE controller using the ControlDesk software. While the data generation loop can technically be programmed in this same way, an alternative approach was instead employed that allowed for greater flexibility, which was to allow Python to interface with the dSPACE controller (also via ControlDesk) during operation of the main control algorithm. Parameters such as the microrobot's 3D position, the final coil currents (or, rather, the final "input" voltages sent to the amplifier in order to request the final coil currents as seen in Equation 2.43), the laser sensors' unprocessed analog outputs (in V), the desired levitation height z_d, and various diagnostics described below are recorded from within ControlDesk and saved for later analysis.

Essentially, the parameters associated with the main vertical control algorithm (e.g., desired levitation height z_d, coil current ratio, vertical PID controller gains, on/off toggles for the feedforward and feedback components of the vertical position controller, an "emergency stop" that sets all coil currents to 0 when enabled, etc.) can be controlled by a user during operation via the ControlDesk software on the host computer; and Python can also interface with ControlDesk to read and write these parameters programmatically. As such, the current implementation of the automated data generation algorithm is to use Python to dynamically alter input parameters from the host computer based on various real-time levitation diagnostics, running in parallel with the levitation algorithm itself on the dSPACE controller.

It is worth noting that while the controller itself operates at 1000 Hz, there is an approximate $\sim$ 6ms delay whenever the Python-ControlDesk interface reads or writes a variable. As such, it is preferred that information that needs to be updated in real time (e.g., position and velocity of the microrobot, time-sensitive diagnostics, etc.) are handled by the controller; whereas information that is less immediately relevant to the vertical position controller (e.g., the CCR, which typically stays constant for hundreds of milliseconds at a time) may be delivered via the Python-dSPACE interface without needing to be programmed directly into the main control loop. These time-sensitive diagnostics which are calculated in real time (i.e., at 1000 Hz) on the controller include things such as:

- Is the microrobot close to leaving a sensor's detection range?
- Has the microrobot actually left the detection range of any/all of the sensors?
- Has the vertical levitation become unstable (marked by the z velocity exceeding a certain speed threshold and continuing to increase)?
- Has the microrobot settled onto its new B_{max} position (marked by the x, y velocities both falling below a certain speed threshold for a certain period of time)?
- Is the control loop requesting a coil current beyond what the amplifier can actually output?
- Is the "emergency stop" enabled?

With these hardware and software considerations in place, the actual data generation algorithm can now be described in full detail.

4.4 Data Generation Loop

With initial hardware and software considerations in place as described above, this subsection will describe the process by which training data are actually generated.

The purpose of the algorithm is to generate large amounts of data that will ideally span the horizontal range of the workspace, by maintaining a fixed levitation height and recording the microrobot's 3D movement as the coil current ratios are randomly fluctuated. The fixed levitation height (i.e., target height z_d) is defined according to user preference.

The data generation process begins by levitating the microrobot at the height z_d (e.g., at 2 0mm above the holding platform), with equal coil currents. The coil current ratio $\vec{i}_{\text{CCR}}$ is an input to the system represented as a 6-vector (with each element representing the relative current flowing through its respective coil), with a sum of 6 such that equal coil currents can be represented by a CCR of:

$$\vec{i}_{\text{CCR}} \equiv \begin{bmatrix} i_{\text{coil 1}} \\ i_{\text{coil 2}} \\ i_{\text{coil 3}} \\ i_{\text{coil 4}} \\ i_{\text{coil 5}} \\ i_{\text{coil 6}} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} . \tag{4.1}$$

From here, data generation can begin. Each new "target" CCR is selected by generating a perturbation $\Delta\vec{i}_{\text{CCR}}$ ("normalized" such that the sum of the coils' currents does not change) and adding it to the CCR. This new target CCR can be represented as:

$$\vec{i}_{\text{target CCR}} = \vec{i}_{\text{initial CCR}} + \Delta\vec{i}_{\text{CCR}} \ , \tag{4.2}$$

$$= \vec{i}_{\text{initial CCR}} + \begin{bmatrix} \Delta i_1 \\ \Delta i_2 \\ \Delta i_3 \\ \Delta i_4 \\ \Delta i_5 \\ \Delta i_6 \end{bmatrix} \ ; \tag{4.3}$$

where Δi_{C} (corresponding to coil C) can be generated entirely randomly but is instead generated in a more pseudo-random manner so as to encourage movement of the levitated magnet. As discussed in Section 2.2.4, the levitated microrobot has a tendency to move towards the B_{max} point; i.e., despite the complexity in modelling the magnetic field, it can still be intuitively understood that the microrobot will tend to move towards a coil with high current rather than a coil with lower current. Because the coils are arranged in a hexagonal shape, the current perturbations are decided by a curve that somewhat resembles a sine wave; such that the coil with the most positive current perturbation will tend to be opposite to the coil with the most negative current perturbation. The relative current for each coil C (numbered 1 through 6) has its perturbation Δi_{C} calculated as follows:

$$\begin{aligned} \Delta i_{\text{C}} = &\left|\sin\left(\text{offset} + (\text{C}-1)\frac{\pi}{3}\right)\right|^{\beta} \\ &* \text{sign}\left(\text{offset} + (\text{C}-1)\frac{\pi}{3}\right) \\ &+ (\text{additional random perturbation}_{\text{C}}) \ , \\ \text{for C} \in &[1,6] \ . \end{aligned} \tag{4.4}$$

Here, the "offset" parameter is a uniformly randomly-generated number from 0 to 2π that is constant for the full set of 6 coil current perturbations. It effectively defines which coils are most strongly emphasized / deemphasized relative to the others, while the β parameter (also constant for each full set of 6 coil current perturbations) defines how strongly coils *adjacent* to these strongly emphasized / deemphasized coils follow suit. By contrast, the "additional random perturbation$_{\text{C}}$" parameter is a value that is indeed entirely randomly generated for each coil, and adjusting the possible range of this parameter will influence how "random" the CCR perturbation is compared to the underlying sinusoid-resembling

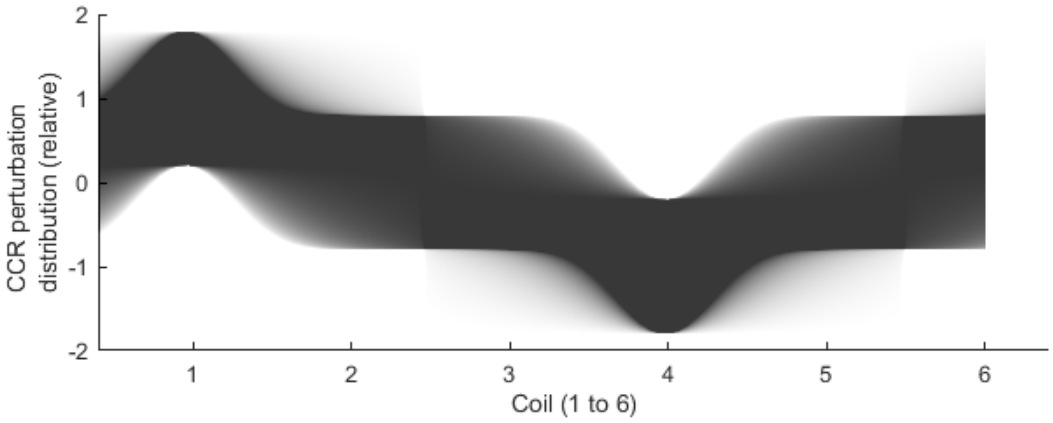


Figure 4.1: Probability density cloud for the per-coil current perturbation selection, visualized with "offset"$= 0$

curve. To illustrate Equation 4.4 more intuitively, Figure 4.1 shows the range of values that each coil's perturbation Δi_C may take for some individual "offset" value. For the sake of encouraging random exploration in the dataset generated and used in this research, each β was a random value between 0 and 10 while the "additional random perturbation$_C$" for each coil was a random number within ± 0.8.

However, while the target CCR is randomly generated as described above, introducing a large CCR perturbation too suddenly will increase the likelihood of the levitation becoming immediately unstable; and decreasing the scale of the perturbation itself may slow down "exploration" across the workspace by causing the microrobot to make small movements in an effectively random direction. In order to encourage larger horizontal movements with each CCR perturbation while reducing the likelihood of levitation becoming unstable, each perturbation is relatively large (with each the relative current of each coil varying by as much as 0.8) but applied in gradual "steps", rather than being immediately applied in its entirety. As such, the CCR is gradually scaled to its target value as per:

$$\vec{i}_{\text{CCR}} = \vec{i}_{\text{initial CCR}} + \alpha * \Delta\vec{i}_{\text{CCR}} \ , \tag{4.5}$$

where α is a unitless "scaling factor" that ranges from 0 to 1, incremented in steps of 0.25.

With this in mind, the data generation loop functions as follows:

1. A CCR perturbation is generated, and "normalized" such that the sum of the target

CCR does not change. For example, if the coils are initially equally loaded as per:

$$\vec{i}_{\text{initial CCR}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \tag{4.6}$$

then the randomly-generated target CCR ("normalized" such that it still adds up to 6) might be something like:

$$\vec{i}_{\text{target CCR}} = \begin{bmatrix} 0.8 \\ 0.3 \\ 1.4 \\ 0.1 \\ 1.9 \\ 1.5 \end{bmatrix}.$$

2. The actual CCR of the system is incremented to this target CCR in steps of 25% at a time. Upon initiation of each "step", the system waits for the microrobot to converge onto its new B_{max} position before incrementing again. (As mentioned in Section 4.3, "convergence" is defined by the microrobot's horizontal speed falling under a certain threshold for a set period of time; and increasing this threshold can increase the pace at which training data are generated, but will correspondingly cut the recording short before the microrobot properly settles onto each new B_{max} point.) Success or failure of this step is handled accordingly:

   (a) If the microrobot converges, the CCR proceeds to the next step.

   (b) If the microrobot reaches the edge of the workspace before converging onto its new B_{max} position, then its CCR is reverted to its previous value and a new CCR is generated (i.e., the incrementing process is aborted and the process returns to Step 1).

   (c) If the microrobot experiences some form of "critical failure" before converging onto its new B_{max} position (e.g., leaving the workspace due to levitation becoming unstable), then data recording is immediately paused, and the microrobot is automatically recovered and re-centered (i.e., returned to the center of the workspace with a CCR of $[1, 1, 1, 1, 1, 1]$ and recording re-enabled) before the process returns to Step 1.

3. If the target CCR was successfully applied, the process returns to Step 1 and is repeated.

4. When the above steps have been running continuously for some specified amount of time, the "cooldown mode" is activated; during which recording is paused, the microrobot is lowered onto the platform at the center of the workspace, and the coils are turned off until they have cooled down. Then, the process repeats.

Note that when the microrobot leaves the workspace or when some other "critical failure" occurs during operation, the microrobot must be recovered and placed back into the workspace so that operation can continue. This process is automated. When the data generation algorithm detects that the microrobot has left the workspace (usually as a result of leaving the limited horizontal range of the z-sensor), recording is paused and a "recovery mode" is enabled. Recovery mode begins by enabling the "emergency stop" coded into the vertical control algorithm, which turns off the power supplied to the coils and allows the microrobot to fall down to the holding platform.

As described in previous sections, equal loading of the coil currents forms a B_{max} point near the center of the workspace, which the microrobot will tend to move towards. Thus, when the microrobot is detected as having landed onto the platform (or when suitable time has passed after enabling the emergency stop, if the microrobot cannot be seen by the laser sensors when resting on the platform), the CCR is incremented towards the "default" CCR of $[1, 1, 1, 1, 1, 1]$, and the "feedforward" component of the vertical position controller (which, as described in Section 3.1, serves the purpose of offsetting gravity) is very briefly enabled. This causes the microrobot to briefly experience a small horizontal force towards the center of the workspace, while its gravitational force is largely offset (reducing its friction against the holding platform), resulting in the microrobot being gently "nudged" towards the center of the workspace even in the absence of position readings. Repeating this process allows the microrobot to be reliably pulled towards the center of the workspace in an open-loop nature, one small nudge at a time; and once the microrobot has been pulled sufficiently close to the centre that the z-sensor can detect it, recovery of the microrobot is considered successful and operation can resume.

Of course, this is assuming that the microrobot successfully landed onto the platform when the emergency stop was first activated. If the microrobot were to miss the platform somehow (e.g., by falling past an insufficiently wide holding platform), then the microrobot typically falls far below the workspace; meaning that the coils can no longer influence the microrobot's motion in any way and manual retrieval becomes necessary. This is the only point in the data generation algorithm when human interaction is needed.

Figure 4.2 summarizes this overall process by which training data are generated. As mentioned previously, the ControlDesk software continuously records the state of the system (including microrobot position, coil currents / amplifier input voltages, time elapsed, etc.) and saves the information for later processing. With this process in mind, a total of 13263 seconds' worth of continuous trajectory data (i.e., 3 hours and 41 minutes' worth of recorded stable levitation, excluding time spent in the aforementioned "cooldown mode" and "recovery mode") were generated and saved for use in subsequent ML applications. While the minimum amount of data necessary for enabling horizontal control with ML (or, in practical terms, the minimum amount of time required for the data generation algorithm to enable horizontal control via ML implementation) has not been investigated, it is worth noting that the entirety of this data generation was performed autonomously and without the need for human attention (let alone intervention).

One potential concern in terms of the time needed to generate a dataset is that the microrobot's motion throughout the workspace is somewhat akin to a random walk, albeit one that frequently resets to a default state upon certain conditions being met (i.e., the CCR of $[1, 1, 1, 1, 1, 1]$ in parameter space, or $[x, y, z] \approx [0, 0, z_0]$ in real 3D space). The result is that the resulting dataset will tend to be overrepresented near the center of the workspace as seen in Figure 4.3, and this imbalance may need to be accounted for (e.g., via data imputation or via removal of overrepresented data [43]) in certain ML implementations.

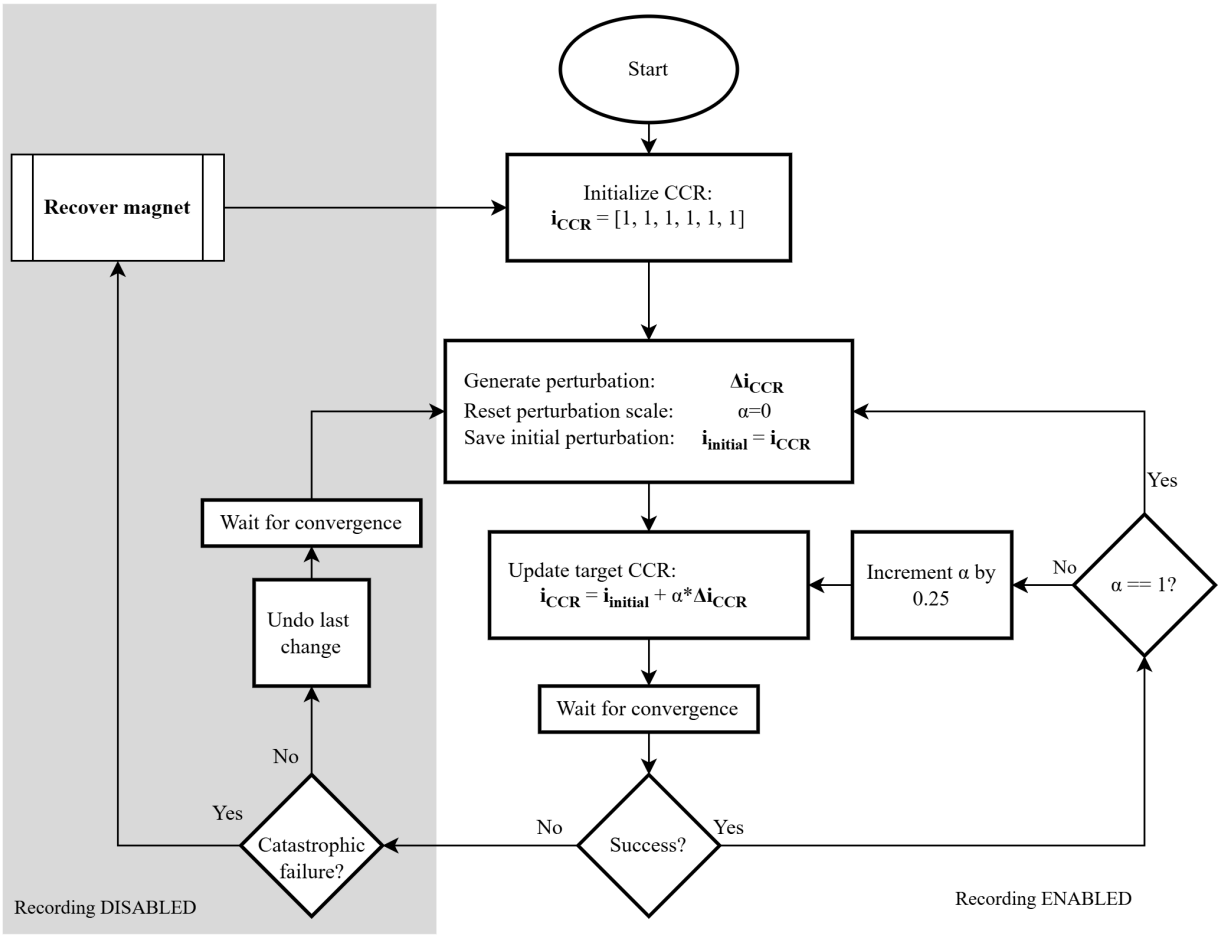


Figure 4.2: Flowchart of data generation loop

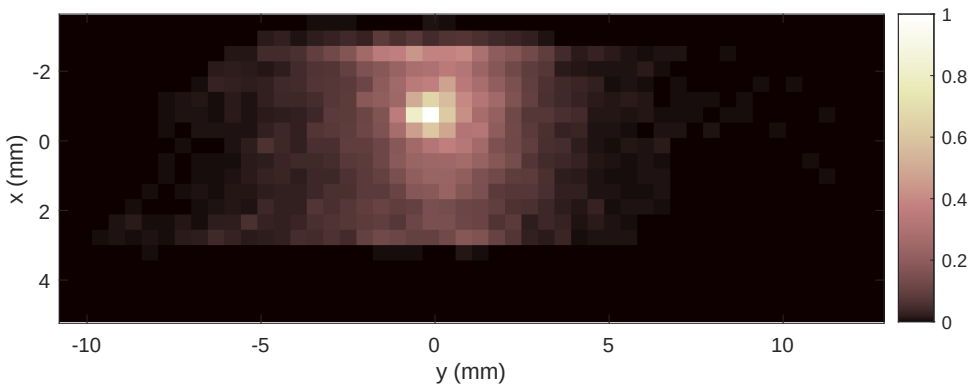


Figure 4.3: Heatmap of all generated training data (specifically, "start" and "end" points associated with a new CCR). Note the overrepresentation of data points near the center of the workspace

Chapter 5

Machine Learning Implementation for Horizontal Position Control

Chapter 4 described a method by which large amounts of data associating ATMA system inputs with the resulting motion of the levitated magnetic microrobot can be produced in an automated manner. These raw data include the time-series measurements of the 3D positions of the microrobot, the coil currents (or, more accurately, the input voltages sent to the amplifier which directly correspond to coil currents), the raw position sensor readings, and various measurement diagnostics; which can then be processed and incorporated into any number of potential ML applications depending on the needs of the researcher.

In this section, we present the final method by which ML was incorporated to enable horizontal control, as well as the preprocessing techniques applied to this raw dataset in order to produce viable training data for the ML models. While the method presented here is *not* the only method for which experiments were conducted, the ongoing nature of these experiments makes them less presentable as primary results; as such, the remainder of this thesis will primarily focus on the successful ML implementation introduced in this section.

5.1 Initial Data Processing

As mentioned above, the recordings of ATMA system inputs and the corresponding "end effector" outputs were generated as described in Chapter 4, but these data must be processed into a more usable format before they can be used for training a machine learning

algorithm. The method by which these data are processed (and the filters used for smoothing the trajectory data, if any) will vary depending on both the intended usage and the final design of the ML implementation. As such, while some consideration will be given to alternative ML experiments, this section will primarily focus on preparation of the recorded data for use in the final ML method for which this section is dedicated.

The unprocessed data are saved as continuous recordings, with each "discontinuity" in the recording (which occurs when recording is paused for any reason, including for the automated coil cooldown periods) marking the end of each file. The recordings are processed and compiled into a final combined table, for use in machine learning applications. The general steps by which the data recordings are processed into a usable format (excluding the ML-specific "feature engineering" steps such as data dropout) are as follows:

1. The data processing algorithm first identifies erroneous "spikes" in the continuous z-versus-time recordings, in which the microrobot's z-position jumps up and down very suddenly (either due to the position being misread as a result of an overly reflective microrobot reflecting a laser beam into the incorrect receiver, or due to some other source of unstable levitation). When such spikes occur, they typically either last for brief $\sim$100ms periods (in the case of the reflection issue) without affecting the data generation loop, or they mark the end of a data generation "step" (after which the microrobot either returns to its previous horizontal position, or the recording ends entirely due to the levitation becoming too unstable for recording to continue; see Section 4.4). The ranges in the recorded data over which these "spikes" are identified are marked so that they are not used in training.

   (a) Despite all erroneous "spikes" being caused by unwanted vertical jitter, the fact that the $\vec{B}$-field is not *perfectly* vertical throughout the workspace means that these vertical motions can have a small but noticeable effect on recorded horizontal motions as well. This is the reason why such occurrences are marked for deletion prior to training a model for horizontal control.

   (b) This step mainly benefits experiments that utilize the entire trajectory of each "step" (e.g., acceleration calculations). The final implementation described in this section only consider the initial/final x-y position data associated with each "step", so it is largely unaffected by this part of the data processing algorithm.

2. Each continuous recording is separated into "steps" associated with a particular coil current ratio.

3. The position-vs-time data are smoothed so as to enable accurate velocity calculations. As mentioned in Sections 2.3 and 3.1.3 , the laser sensor readings are quite precise overall, but still have noisy data that must be smoothed for optimal precision in calculations. Processing prerecorded data (as opposed to processing part of a feedback loop in real time) allows for more flexibility in the methods used, as non-causal filters can be utilized and any group delay in the filtered signal can be undone.

    (a) Figure 5.1 displays the frequency response of the laser sensors for a recording of a completely stationary magnet at rest on the platform (i.e., coils are turned off). Notably, this figure shows a sharp peak at f_n =100Hz, indicating the frequency of the noise that contributes most strongly to the recorded data. This noise is overwhelmingly removed by simply applying a moving mean filter with a width of 10ms (or 10 samples, since our controller operates at f_s =1000Hz); however, in practice, there is still some leftover noise that must be accounted for due to the way MATLAB and Simulink handle moving mean filters whose sliding window length (in samples) is an even number [44]. This leftover noise is removed from the dataset by using what we will hereafter refer to as a "resampling filter", which essentially works by resampling the recorded data such that a moving mean filter with an *odd*-sampled width (in this case, 11 samples) will capture exactly one cycle of the 100Hz noise [45]. Once the moving mean filter is applied to the resampled data, the resampling step is undone and the signal (now properly smoothed) is returned to its original time domain.

4. Once the noise in the recorded data has been removed to a satisfactory degree, corrections can optionally be made to the x-y position on which the microrobot converges at the end of each "step". As mentioned in Section 4.4, shortening the duration of each "step" during recording can result in less precise calculations of the final B_{max} position (which, by default, is simply the final position of the microrobot at the end of each "step"); however, this can optionally be accounted for as described in Section 5.4.

5. The gradients of the now-smoothed position data are optionally calculated to obtain the approximate velocities of the microrobot throughout the recordings, which may be used in certain ML applications. (Calculating accelerations would be ideal, but is less straightforward than simply taking the gradient of the gradient due to its sheer sensitivity to noise from the position-versus-time curve.

6. Finally, all relevant data from the recordings are compiled into a single table. In the context of the ML experiment presented in this section, these "relevant data" are the

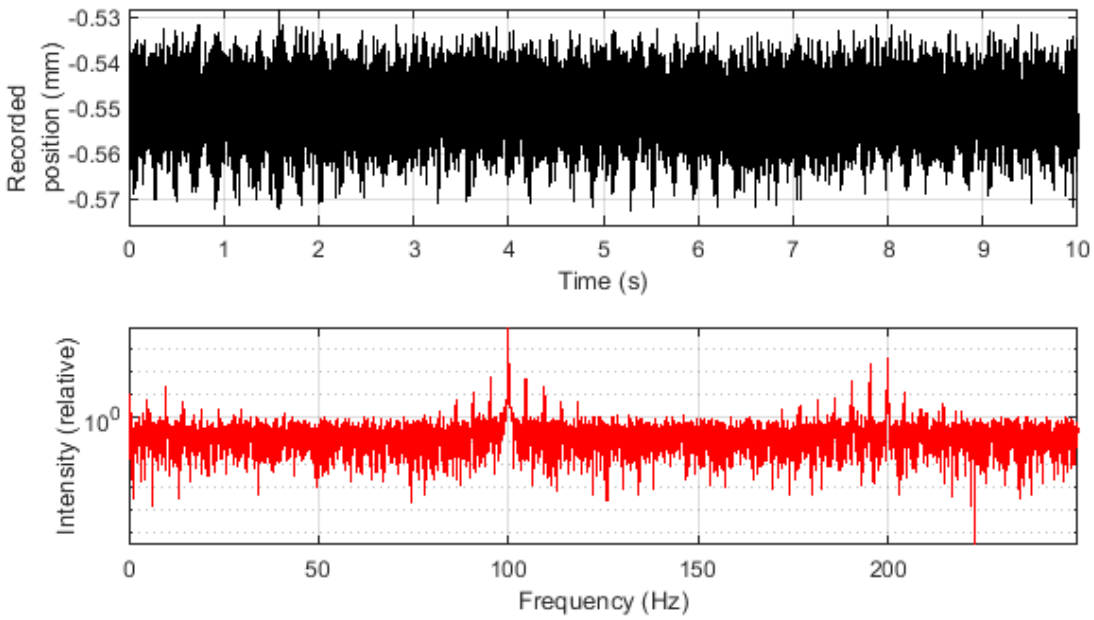


Figure 5.1: Power spectrum of unmodified, unfiltered laser sensor readings of a perfectly stationary object

initial and final microrobot positions at each step, the CCR corresponding to that step, the initial velocities at that step (optionally), the "extrapolated" final positions of the microrobot at each step (optionally), and the duration between the start and end of that step.

Once the recorded data have been processed and compiled into a single location, experimentation with ML methods that utilize these data can begin.

5.2 Machine Learning-Assisted Position Control

The remainder of this chapter will present the methodology and implementation of the method by which ML was subsequently employed to utilize the automatically-generated training data to enable horizontal position control of the levitated magnet.

To very briefly summarize the ML method presented throughout the remainder of this section, the training data generated as described in Chapter 4 and processed as described

in Section 5.1 are utilized to train a series of neural networks to predict the relative magnetic force experienced by the axisymmetric microrobot based on the input coil current ratio (CCR). These neural networks are then used to generate a wrench matrix that is representative of the dynamics exhibited in their training data; and this (largely automated) process can then be repeated with different subsets of training data to generate wrench matrices that represent the differing levitation dynamics across different regions in the workspace.

Section 5.2.1 will elaborate on the principles that allow this method to work in the first place, while Section 5.2.2 will elaborate on the methodology itself. Finally, Section 5.3 will present the actual parameters that were employed for this methodology so as to enable horizontal position control of a levitated single-magnet carrier for our ATMA.

5.2.1 Motivation & Goals

One of the most conceptually straightforward methods of generating wrench matrices using machine learning is to train a neural network to relate the system inputs to the resulting forces applied to the object of interest. The magnetic forces applied to a freely-levitating magnet by an ATMA can in principle be calculated if the microrobot's accelerations throughout the full set of training data can be found, and if any additional forces such as "braking effects" (primarily, the magnetic braking effects that may be deliberately employed to speed up data generation as seen in Chapter 4) can be modelled accurately and precisely. However, the presence of noise in the recorded data can make accelerations difficult to calculate with sufficient precision; let alone in a manner that can be employed consistently across different ATMA embodiments whose position detection methods may not have the same sampling rate, measurement precision, noise levels, and the like. As such, for the sake of accessibility and repeatability of the final methodology, a different approach to measuring magnetic force is needed.

As described in Equation 2.39, the magnetic force experienced by an axisymmetric magnet levitated by an ATMA is dependent on the spatial gradient of the magnetic field strength at the magnet's position. Previous studies have observed that when the magnetic microrobot is levitated at a constant height, the magnetic field generated across the x-y plane rises and falls in an approximately parabolic manner with respect to x and y [4]. Together, these behaviours of ATMAs result in a corollary that the steady-state horizontal magnetic force applied to the microrobot (that is, the magnetic force applied to the microrobot prior to speed-dependent effects such as magnetic braking or air resistance) is linearly proportional to the distance between the microrobot and its B_{max} point on the

x-y plane, akin to a simple harmonic oscillator:

$$F_x = K_1 \left(x_{\mathrm{B_{max}}} - x_{\mathrm{start}} \right) \ , \tag{5.1}$$

$$F_y = K_2 \left(y_{\mathrm{B_{max}}} - y_{\mathrm{start}} \right) \ ; \tag{5.2}$$

where K_1 and K_2 can be thought of as the "stiffnesses" of the magnetic field in each direction, and can be treated as approximately constant in each horizontal plane [39]. This behaviour can be easily observed when the microrobot is trying to move to its $\mathrm{B_{max}}$ point but is obstructed by an object that can measure its applied force, such as a cantilever beam [4].

An important consequence of this behaviour is that if the microrobot is levitated in place and a new CCR is applied (such that the microrobot moves to a new $\mathrm{B_{max}}$ point on the horizontal plane), the initial horizontal magnetic force applied to the microrobot is proportional to its final horizontal displacement. That is, simply measuring the microrobot's change in x-y position in response to a new CCR (which is significantly easier to capture with position sensors than acceleration data) will yield the relative horizontal magnetic force that was applied to the magnet– meaning that a ML model trained to predict the change in position associated with a change in CCR will in principle be able to output a wrench matrix for horizontal control of the magnet!

A notable caveat to this method that must be accounted for is that the levitation dynamics vary throughout the workspace, meaning that a single wrench matrix may not be sufficient to capture these position-dependent dynamics. However, the fact that these wrench matrices are generated via machine learning (rather than via analytical calculations or empirical measurements) means that by separating the training data into position-dependent groups, this method can just as easily generate *multiple* wrench matrices that each capture the dynamics within a different region of the workspace. A further advantage of this method is that, because each trained network will try to predict the change in microrobot position that will result from a change in CCR, all of the networks' outputs will have the same units (i.e., the same proportionality between network output and actual magnetic force). This means that the final horizontal position (or force) controller that utilizes these wrench matrices will only need to be tuned once, and these tuned parameters will in principle work just as well for new wrench matrices (if any) as they are added to the control model.

There are further advantages to this method of ML implementation that cemented its usage over others in this research, mainly in regards to its relative ease of implementation. An important consideration is the time taken to run the final horizontal control model; and the fact that this method uses ML to generate wrench matrices that are used in the

final real-time control algorithm (as opposed to the NNs themselves being employed during real-time control) means that there is less need for the trained NNs to make predictions incredibly quickly. Another consideration is that this model does not need to be able to output a CCR that will yield desired input motion, which would be far more complex than the "forward" relationship of predicting motion from an input CCR (due to the possibility of multiple CCRs yielding the same microrobot motion, which the model would need to "know" how to choose between). Additionally, as alluded to above, the presented method allows the relative magnetic force to be predicted without needing to account for (let alone quantify) the magnetic braking effects from nearby conductive (but otherwise nonmagnetic) materials, from objects that are deliberately placed to speed up data generation (such as an aluminum holding platform) to unwanted conductive materials that happen to be near the workspace (such as copper wires connected to the position detection devices).

Furthermore, because this method uses ML to simply generate wrench matrices (i.e., summaries of the horizontal force dynamics during levitation), it follows that the precision of each individual NN's outputs actually has *very little* effect on the smallest scale over which the final control model can be applied. That is, while more precision is of course ideal, the NNs presented in this method do not actually require the ability to make μm-scale position predictions or μN force predictions in order for the final control model to operate on those scales.

Expanding on the above idea, it should be noted that an immensely important aspect of many NNs is the *consistency* of their predictions– but the fact that the NNs in this method are not actually used in real-time control, combined with their correspondingly relaxed requirements in terms of processing time, means that even wild inconsistencies in the final trained NNs' outputs (i.e., high variance in each network due to problems like overfitting, sensitivity to weight+bias initialization, and/or unaccounted-for complexities such as the hysteresis effect) can actually be alleviated to a degree by simply training *multiple* such networks and averaging their outputs [46]. In other words, the quality of the final wrench matrices is entirely dependent on the collective *accuracy* of the trained NNs' outputs; whereas issues regarding poor *precision* (i.e., high variance) in each network can actually be alleviated by generating each network multiple times and combining their results for more consistent behaviour and better generalization.

5.2.2 Generation & Usage of Wrench Matrices

The ML implementation presented in this chapter operates with the goal of obtaining a "wrench matrix", which represents the relative horizontal force that each coil will apply to

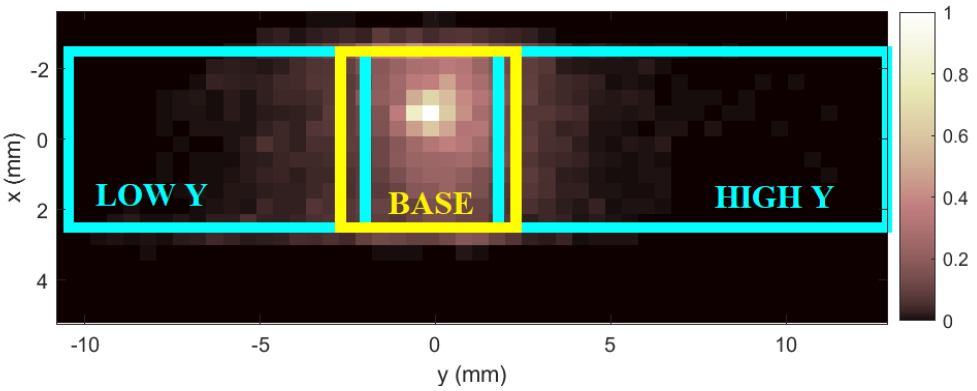


Figure 5.2: Visualization of the three distinct "subregions" defined in this research, each exhibiting distinct levitation dynamics

the microrobot throughout the horizontal plane on which data was recorded. Because the dynamics vary throughout the workspace over which training data are generated, accounting for these varying dynamics in the final control model is done by generating multiple wrench matrices, each of which is trained on a particular subset of the training data associated with a particular range on the x-y plane (hereafter referred to as a "subregion" of the recorded dataset).

These "subregions" do not necessarily need to be mutually exclusive. For example, the subregion representing the "center" of the workspace could have its respective wrench matrix trained on the innermost 90% of the dataset; but using this wrench matrix alone may result in worsened performance at high y values, due to the "central" wrench not adequately reflecting the dynamics in this region. As such, a second subregion representing the "high Y" region of the workspace could be defined, and its corresponding wrench matrix generated accordingly. Figure 5.2 displays the "subregions" that were chosen for generating the wrench matrices used for the purposes of this research. Note that while this research only employs three ML-generated wrench matrices in the final horizontal control model, it is perfectly possible to separate the workspace into a greater number of subregions that are each governed by their own wrench matrix– albeit with the caveat that each subregion still needs an appropriate amount of training data for its wrench matrix to be accurately generated, meaning that increasing "granularity" of the selected subregions will correspondingly require more training data and/or more overlap between the boundaries of each subregion (with this "overlap" primarily functioning as a workaround in the case of a subregion having insufficient training data).

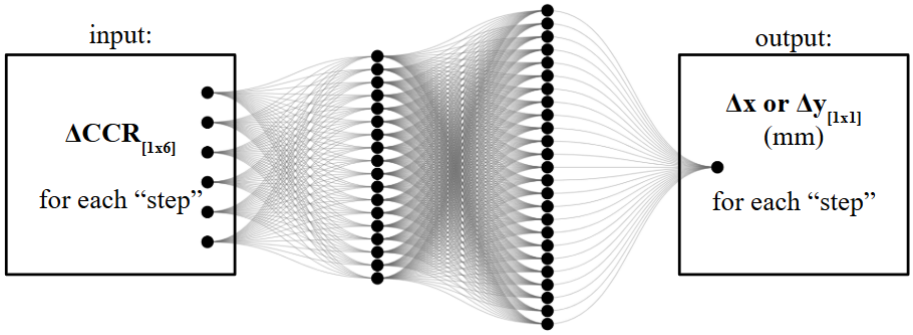


Figure 5.3: Visual summary of shallow neural network structure (fully connected)

When a subregion and its respective range of training data are selected, its wrench matrix is generated by training a series of shallow NNs to predict the change in x-y position of the levitated microrobot (which, as described in Section 5.2.1, is a quantity that is linearly proportional to the horizontal magnetic force) as a result of some inputted change in CCR. Subsequently, these trained networks are asked to predict the resulting motion due to changes in *one coil current at a time*, and these predictions are what populate the elements of the wrench matrix corresponding to each coil.

The levitation dynamics of the microrobot across the x and y directions are handled separately. For each direction, the predictors of each shallow NN are $[1\times6]$ vectors representing the change in CCR, with corresponding responses being the (scalar) associated change in (x or y) position, as summarized by Figure 5.3. Because each training sample does not include information about the "sequential" nature of these quantities (in regards to the hysteresis effect, in which changing the CCR and then undoing that change may not return the microrobot back to its original position), the hysteresis effect is unlikely to be directly accounted for by each network. However, the pseudo-random nature in which training data were generated as described in Chapter 4 means that the unaccounted-for hysteresis effect will likely affect the networks in the same way that random noise would. That is, the yet-unaccounted-for hysteresis effect is expected to manifest as *variance* in the trained NNs, rather than as bias (as described in Section 5.2.1).

The optimal shallow NN configurations are obtained through Bayesian optimization (specifically, through MATLAB's "*bayesopt()*" functionality), performed for each subregion for each of the x and y directions using mean squared error (MSE) as the minimization objective. In particular, the optimization variables included:

- Size of each hidden layer

- Training function (with options including Levenberg-Marquardt backpropagation, Bayesian regularization backpropagation, and scaled conjugate gradient backpropagation)
- Data preprocessing methods (with options including normalization and standardization, each with and without dimensionality reduction through principle component analysis)
- Learning rate
- L2 regularization

The optimization was performed using one, two, or three hidden layers (so as to determine the best balance between network performance and computation time); each using a 70/15/15 training/testing/validation data split. It was found that the performance of each network had extreme variance dependent on initialization of weights and biases, even with identical training data and hyperparameters; and these inconsistencies made it difficult for the Bayesian algorithm to perform hyperparameter optimization as intended. This was worked around by training at least 20 networks for each set of hyperparameters, and using the performance of the best out of those networks to calculate the objective function for the purposes of Bayesian optimization.

After finding the optimal hyperparameters for the x and y directions within each subregion, training and utilizing the final neural networks could begin. For each of the x and y directions in each of the previously-defined subregions, a set of M distinct neural networks were trained using the same set of training data and the same hyperparameters (with each of these M networks selected as a "best out of 10" as was done in the Bayesian optimization step, in order to reduce the likelihood of unintentionally utilizing a poorly-performing network in the final application). The final ML model that was used to predict the changes in x or y position within each subregion (hereafter referred to as a "combined network") is simply the mean of the predictions of the M networks. This parameter M would ideally be set as high as possible to compensate for the low precision of each network's prediction; however, due to time constraints at this phase in the research, M was simply set to 50 networks per direction per subregion.

Once the "combined networks" for x and y have been generated for each subregion, they can then be used to generate the respective wrench matrices. Each subregion's wrench

matrix is generated by populating the elements of the array $\mathbf{A}_{\text{sub}}$, such that:

$$\mathbf{A}_{\text{sub}} = \begin{bmatrix} a_{X,C=1} & a_{Y,C=1} & 0.1667 \\ a_{X,C=2} & a_{Y,C=2} & 0.1667 \\ a_{X,C=3} & a_{Y,C=3} & 0.1667 \\ a_{X,C=4} & a_{Y,C=4} & 0.1667 \\ a_{X,C=5} & a_{Y,C=5} & 0.1667 \\ a_{X,C=6} & a_{Y,C=6} & 0.1667 \end{bmatrix}, \tag{5.3}$$

where the constant 0.1667 figure in the third "a_Z" column comes from the assumption that the six coils contribute equally to the magnetic force in the z-direction. Here, each of the matrix elements associated with position x / y and coil C is calculated from the respective combined network as the average change in microrobot position (Δx or Δy, depending on the combined network used) predicted by a varying input CCR perturbation $\Delta\vec{i}_C$ (such that, as an example, the input CCR perturbation associated with Coil 1 would take the form of $\Delta\vec{i}_1 = [\frac{5\Delta i}{6}, -\frac{\Delta i}{6}, -\frac{\Delta i}{6}, -\frac{\Delta i}{6}, -\frac{\Delta i}{6}, -\frac{\Delta i}{6}]$, where the value of perturbation $\Delta\vec{i}_1$ is varied by incrementing Δi from 0.1 to 1 in small steps). Note that these final values that serve as the wrench matrices' elements do not need to be "normalized" or otherwise postprocessed in any way; and the proportionality between these matrix elements (i.e., changes in microrobot position associated with each coil) and the actual horizontal magnetic force is instead handled entirely by the PID algorithm. Figure 5.4 summarizes the overall process by which wrench matrices are generated for each subregion.

Upon generation of the "central" wrench matrix (or the "base" wrench matrix, which is trained with the majority of the generated data due to over-representation near the center of the workspace), this wrench matrix can be implemented into a horizontal PID control algorithm such that it converts the PID controller's signal into the final plant input, as per the steps described in Section 3.1. However, with multiple wrenches to choose from, the controller can be modified such that the 3DOF PID signals are instead fed into a subsystem referred to in this thesis as a "wrench scheduler", as seen in Figure 5.5. The purpose of this "wrench scheduling" subsystem is to dynamically choose (or interpolate) between the available wrench matrices based on the microrobot's position, such that the levitation is always controlled with a suitable wrench matrix even as levitation dynamics vary throughout the workspace.

When the wrench matrix is altered during real-time control as the microrobot passes from one subregion to another, this alteration must be introduced gradually so as to not introduce any "discontinuities" to the control loop when the microrobot passes by some invisible boundary line. Because subregions may share borders (or perhaps even overlap, in the event that padding a subregion's dataset is necessary), the current iteration of

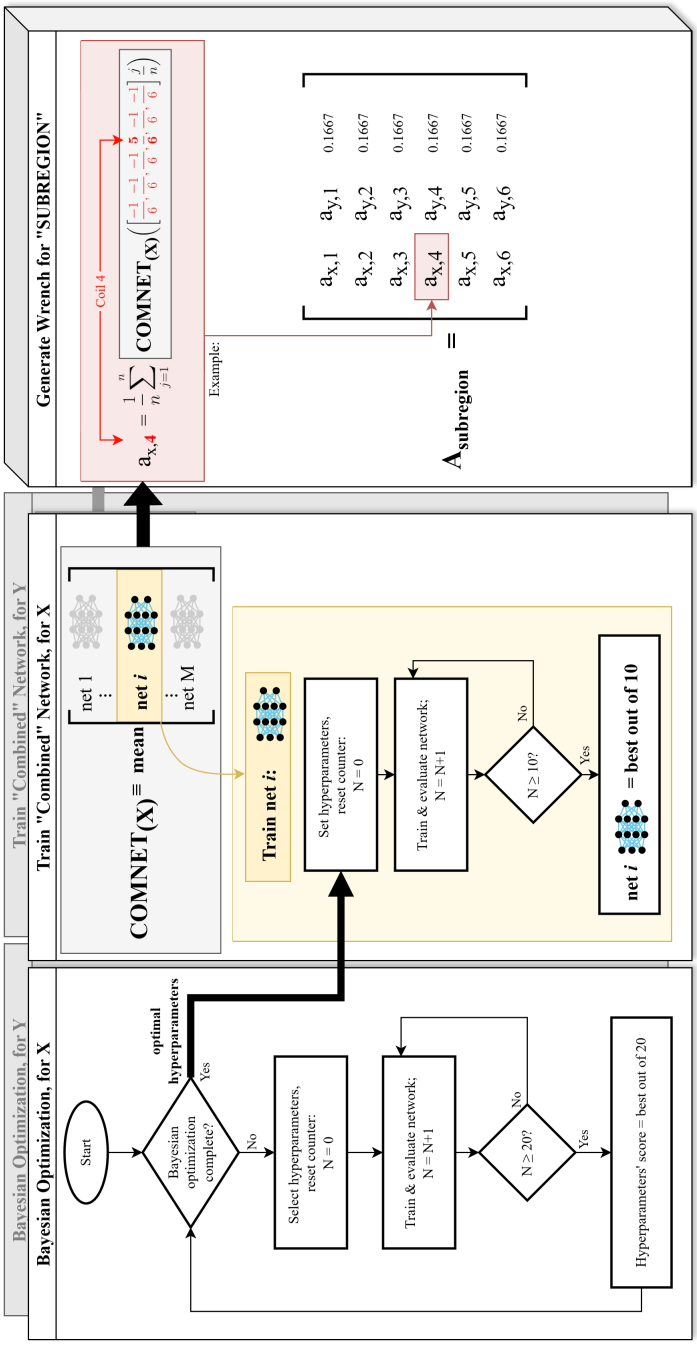


Figure 5.4: Flowchart for generating the wrench matrix for a single "subregion". Note the x and y components using separate networks!

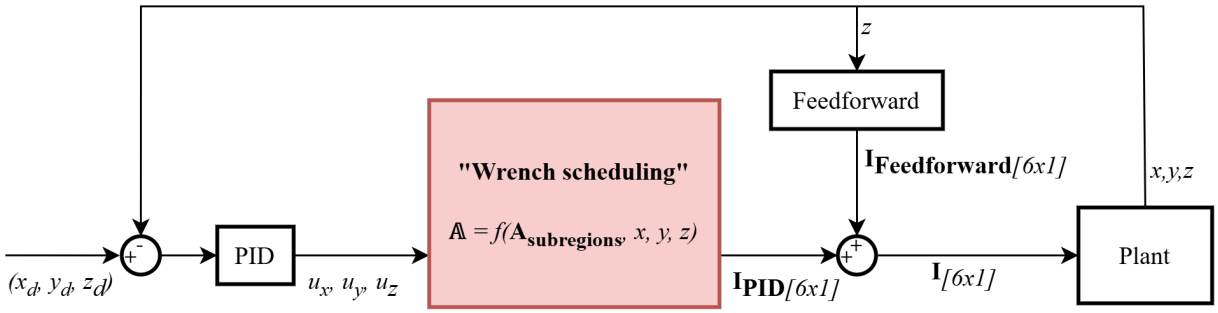


Figure 5.5: Summary of final 3-DoF position control model

the algorithm interpolates between the wrench matrices based on the *average position* (as opposed to the outer boundaries) associated with each non-central subregion. For example, if the wrench matrix associated with the "HIGH Y" subregion was trained using data in the range $y \in [y_{\text{HIGH Y, min}}, y_{\text{HIGH Y, max}}]$, its wrench matrix could be fully applied when $y > \frac{y_{\text{HIGH Y, min}} + y_{\text{HIGH Y, max}}}{2}$, instead of over the entire span of the subregion. Similarly, if the "LOW Y" subregion considers data within the range $y \in [y_{\text{LOW Y, min}}, y_{\text{LOW Y, max}}]$; its wrench matrix could be applied in full when $y < \frac{y_{\text{LOW Y, min}} + y_{\text{LOW Y, max}}}{2}$. The "BASE" subregion (which contains the majority of the full dataset) could be applied in the same manner; or, if it is considered to be sufficiently accurate over its range of values, its wrench matrix could be fully applied over its range of values (i.e., when $y \in [y_{\text{BASE, min}}, y_{\text{BASE, max}}]$). Finally, when the levitated microrobot falls between these aforementioned ranges that are considered to be fully governed by the subregions' respective wrench matrices, the 3D position control algorithm simply produces an "intermediary" wrench matrix through linear interpolation.

Figure 5.6 provides a visual summary of this "wrench scheduling" algorithm in the 1D case. This research only presents the 1D case for this algorithm (i.e., where subregions and their wrench matrices are defined only along the y-axis of the workspace) due to the x-range of this configuration being very narrow, which leads to the levitation dynamics exhibiting less significant variation across the available x-range. However, given a wider x-range of the workspace, this algorithm could be expanded to the 2D case following the same principles.

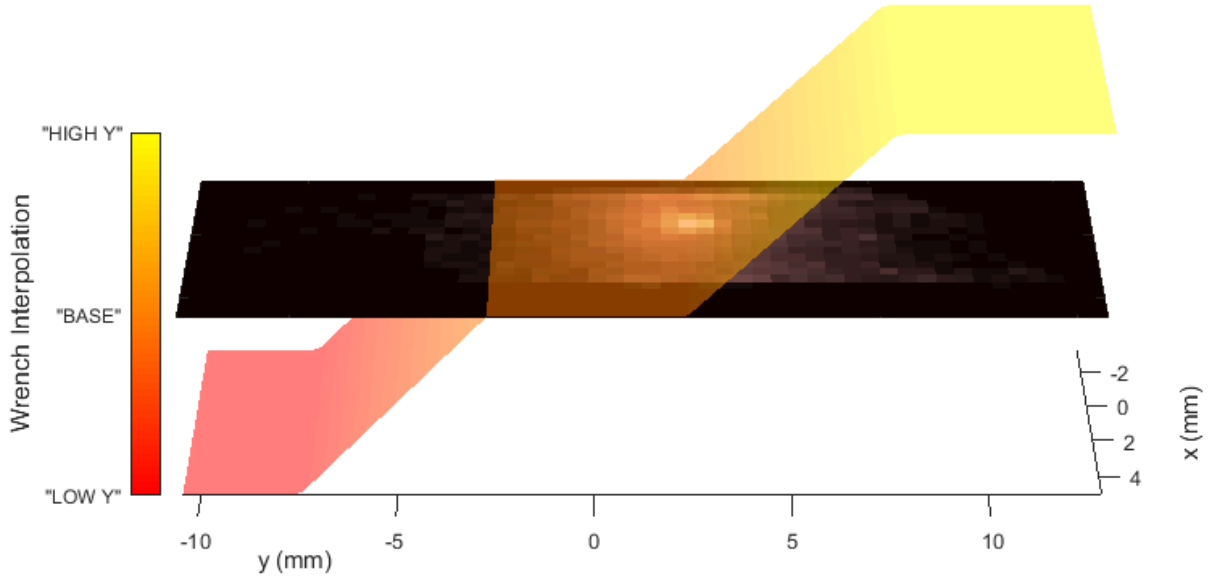


Figure 5.6: Visualization of how the final horizontal control model selects (and interpolates) between the available wrench matrices based on microrobot position (in the 1D case)

5.3 Final Parameters

As described previously, the levitation workspace was separated into three distinct "subregions" as visualized in Figure 5.2. The process described in Section 5.2.2 included training many neural networks with the data associated with each subregion, and these neural networks were used to produce a wrench matrix describing the horizontal levitation dynamics throughout its respective subregion.

Table 5.1 displays the optimal hyperparameters that were found through the Bayesian hyperparameter optimization process for each of the three subregions. By training and combining a number of neural networks for each subregion using these optimal hyperparameters, a wrench matrix corresponding to each subregion was generated. The final values of these wrench matrices (and the regions over which they fully apply during the final control algorithm) are:

$$\mathbf{A}_{\text{BASE}} = \begin{bmatrix} -0.3916 & 0.6670 & 0.1667 \\ -0.6938 & 0.0748 & 0.1667 \\ -0.4567 & -0.6052 & 0.1667 \\ 0.3547 & -0.6327 & 0.1667 \\ 0.8542 & 0.0652 & 0.1667 \\ 0.5238 & 0.7651 & 0.1667 \end{bmatrix}, \text{ for } y \in [-2.7\text{mm}, 2.3\text{mm}] \tag{5.4}$$

$$\mathbf{A}_{\text{HIGH Y}} = \begin{bmatrix} -0.4382 & 0.8795 & 0.1667 \\ -1.0842 & 0.1465 & 0.1667 \\ -0.8262 & -1.4016 & 0.1667 \\ 0.7130 & -1.4546 & 0.1667 \\ 1.1960 & 0.3060 & 0.1667 \\ 0.5674 & 0.8293 & 0.1667 \end{bmatrix}, \text{ for } y > \text{mean}\,(5\text{mm}, 10\text{mm}) \tag{5.5}$$

$$\mathbf{A}_{\text{LOW Y}} = \begin{bmatrix} -0.3231 & 0.9558 & 0.1667 \\ -1.2939 & 0.2757 & 0.1667 \\ -0.2898 & -1.1806 & 0.1667 \\ 0.4042 & -0.8296 & 0.1667 \\ 0.9029 & -0.0918 & 0.1667 \\ 0.5050 & 1.6513 & 0.1667 \end{bmatrix}, \text{ for } y < \text{mean}\,(-10\text{mm}, -5\text{mm}) \tag{5.6}$$

For regions in between these areas where a specific wrench matrix is fully applied, interpolation between the wrench matrices is used as described above.

Table 5.1: Optimal hyperparameters for each subregion

Subregion		**Hidden Layers**	**Training Function**	**Preprocessing**	**Learning Rate**	**L2 Regularization**
"BASE"	X	[37,50]	trainbr	standardize	0.014247	0.097802
	Y	[50,13]	trainbr	standardize	0.0020217	0.081187
"HIGH Y"	X	[24]	trainbr	default	0.00040437	0.080204
	Y	[26]	trainlm	normalize	0.024054	0.096126
"LOW Y"	X	[40]	trainbr	standardize+PCA	0.0001259	0.062506
	Y	[50]	trainbr	normalize+PCA	0.014546	0.099002

5.4 Alternative Methods

It is worth noting that stopping each "step" prematurely during data generation (see Section 4.4) may have unintentional consequences on the reliability of certain measurements. Each "step" of data generation ideally ends when the microrobot settles onto its final position, but due to the time it takes for the microrobot to truly settle onto this point (especially if the simple harmonic nature of its oscillation is underdamped or overdamped by the aluminum platform), the algorithm can stop each "step" early by relaxing the speed threshold at which the microrobot is deemed as having "stopped". This comes at the cost of precision, although it is in principle possible to alleviate this issue by calculating the "true" stopping point of the microrobot in each "step" via extrapolation.

If the microrobot behaves as an underdamped oscillator on the horizontal plane, its equation of motion in the x- and y- directions (under the assumption of constant vertical distance from the aluminum platform) will take the approximate form of:

$$x(t) = A_x e^{-B_x t} \cos\left(C_x t + D_x\right) + E_x \ , \tag{5.7}$$

$$y(t) = A_y e^{-B_y t} \cos\left(C_y t + D_y\right) + E_y \ ; \tag{5.8}$$

for some unknown constants A, B, C, D, and E. Similarly, if the microrobot behaves as an overdamped oscillator, its equation of motion will follow the form:

$$x(t) = A'_x e^{-B'_x t} + C'_x \ , \tag{5.9}$$

$$y(t) = A'_y e^{-B'_y t} + C'_y \ . \tag{5.10}$$

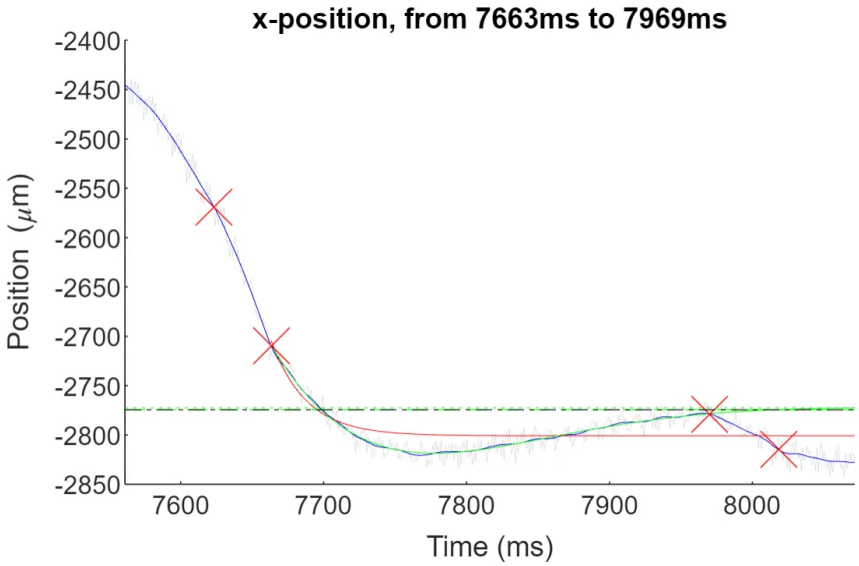


Figure 5.7: Example of the "true" B_{max} point (black dashed line) of a CCR "step" being extrapolated from raw data (blue), assuming either an overdamped (red) or underdamped (green) oscillation

In the cases of recorded "steps" (i.e., trajectories associated with a newly introduced CCR, ending immediately before the CCR changes) ending before the microrobot's horizontal motion actually stops, the microrobot's trajectory in the x-y directions can be fitted to one of these damped oscillator models, and successful fits can be extrapolated to predict the microrobot's true "final position" as seen in Figure 5.7. (In the context of the data recordings generated and used in this thesis, performing this step did not have a significant impact on the overall performance of the final control model.)

Chapter 6

Performance & Discussion of ML-Assisted Methodology

With the ML-assisted 3-DoF position control model implemented for our ATMA as described throughout Chapters 4 and 5, this section presents and analyzes the preliminary performance of this model based on qualitative and quantitative comparison between the novel ML-assisted model with the previous state-of-the-art. Followup experiments that can improve the performance of the overall methodology presented throughout this thesis are described in Section 6.2; and recommendations for future developments and potential applications for the presented methodology are described in Section 6.3.

6.1 Performance of Final Model

As described in Chapter 5, the final ML-assisted control model presented in this thesis generates "wrench matrices" that describe the levitation dynamics across the workspace using machine learning. The initial wrench matrices generated using this method, each corresponding to a different "subregion" of the workspace, where it is assumed that levitation behaves differently than in the center due to altered proximity between the microrobot, the coils, and any objects comprising the ATMA or near the workspace that may influence the levitation behaviour. These ML-generated wrench matrices were shown in Equations 5.4 to 5.6; and the primary point of comparison is the analytically-obtained wrench matrix from the current state-of-the-art, shown in Equation 3.1.

A notable consequence of our ML-assisted wrench matrix generation algorithm is that the final "combined" neural networks associated with each subregion are trained to predict

the change in horizontal position that will result from a perturbation to the input CCR. This means that the wrench matrices generated for each subregion share the same "scaling"; and thus account for variations in levitation dynamics across the defined subregions before performing PID gain scheduling across the horizontal plane (which may otherwise improve performance if a single wrench matrix is to be used instead). As such, to evaluate the performance of the ML-assisted method compared to the previous SOTA, the control models presented in this section all share a single set of PID gains across the horizontal plane. The wrench matrix generated through the previous SOTA method is multiplied by a scalar value to match the overall "scale" shared by the ML-generated wrenches, so as to minimize any performance impact associated with inconsistent scaling between the wrench matrices. Additionally, for the sake of contextualizing any observed differences in performance between the two methodologies, a third methodology is also included in these performance analyses that comprises simply the wrench matrix associated with the "BASE" subregion (i.e., Equation 5.4, with no wrench scheduling). As illustrated in Figure 5.6, the performance of this "BASE wrench only" method should be functionally indistinguishable from the wrench scheduling method within the "BASE" subregion.

In order to evaluate the performance of each methodology, each of the aforementioned control models was deployed onto the hardware and used to operate a series of levitation experiments in which the microrobot's physical response to a predefined "reference" trajectory was measured. Section 6.1.1 will describe the immediate qualitative differences between the three control methods. Afterwards, Section 6.1.2 will present the similarities and differences in performance in more detail, particularly in regard to the transient response of the control models.

6.1.1 Qualitative Results

For these performance evaluations, the control models' trajectory-tracking responses were recorded for various preset trajectories; including a "figure eight", "rectangle", "vertical helix", "horizontal helix", and "oval", for which the target position is continuously varied over up to 3 dimensions at a time. Figures 6.1 through 6.5 illustrate these trajectory tracking experiments as they would appear in 3D space.

Perhaps the most immediate observation that can be made from these recordings is the considerable overlap between all three methods, in the sense that all of them are seemingly able to keep up with the desired path with little to no issue. Figures 6.6 through 6.10 display top-down views of these trajectories as well as the time-series variations in x, y, and z for each of the models, allowing for the disparities to be observed with slightly more detail.

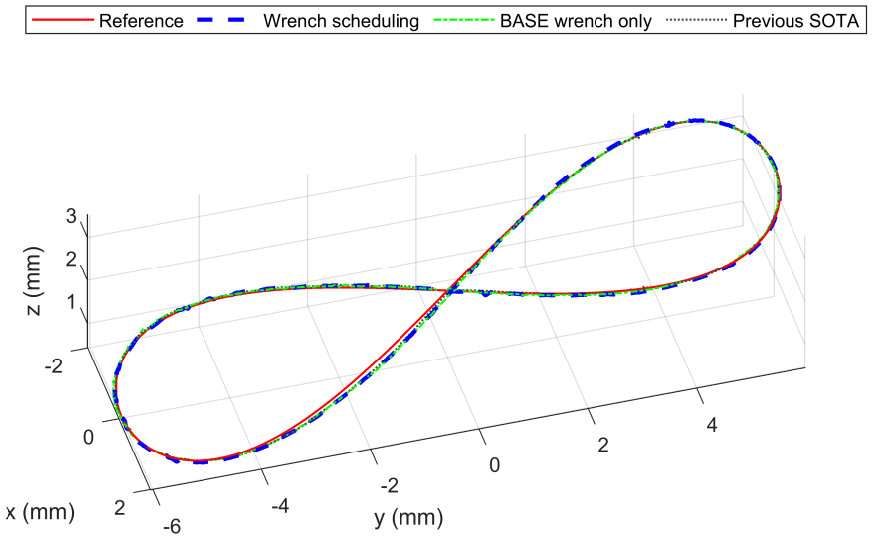


Figure 6.1: 3D trajectory tracking comparison ("figure eight")

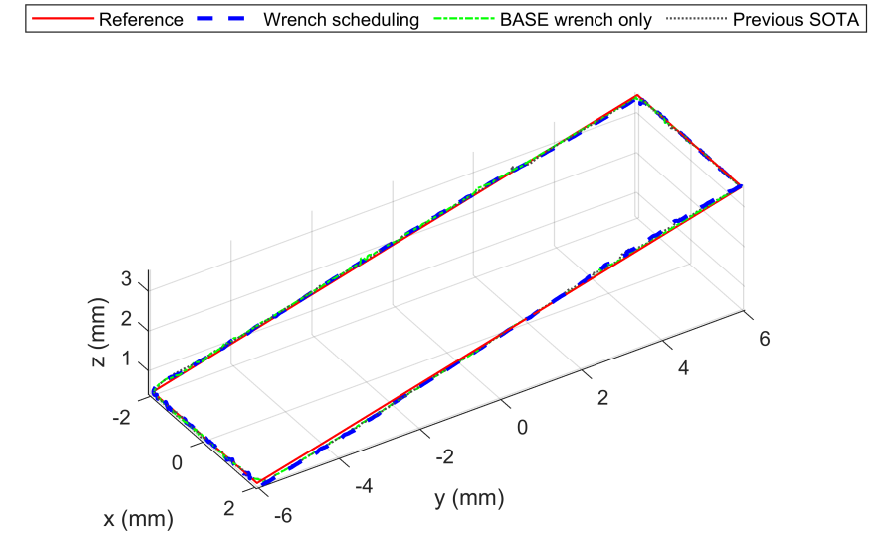


Figure 6.2: 3D trajectory tracking comparison ("rectangle")

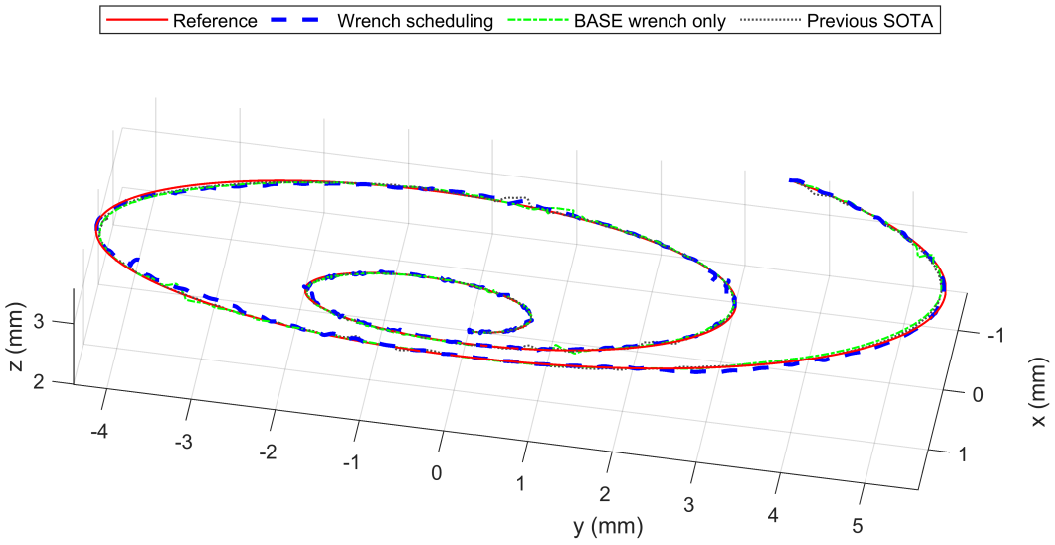


Figure 6.3: 3D trajectory tracking comparison ("vertical helix")

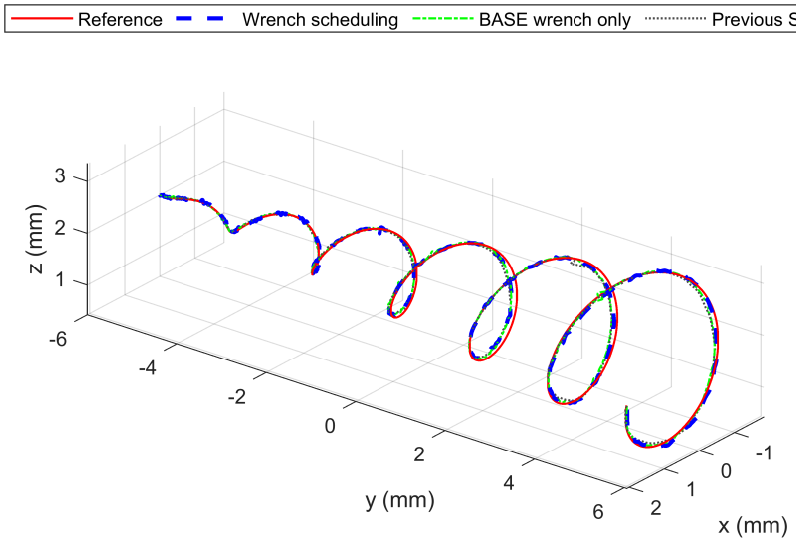


Figure 6.4: 3D trajectory tracking comparison ("horizontal helix")

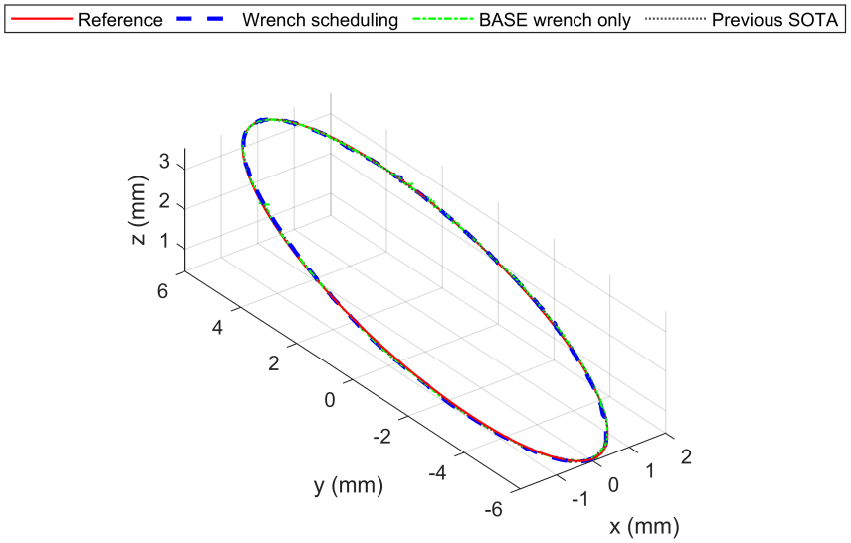
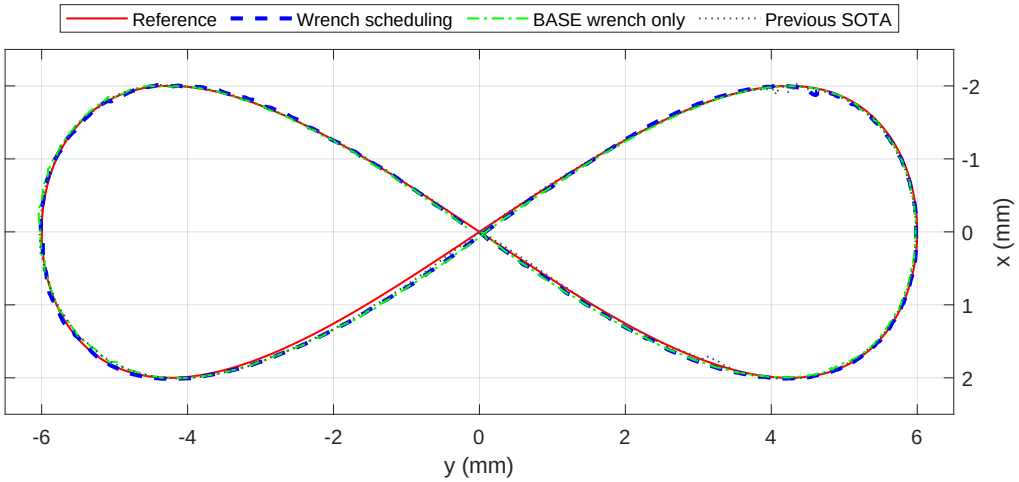


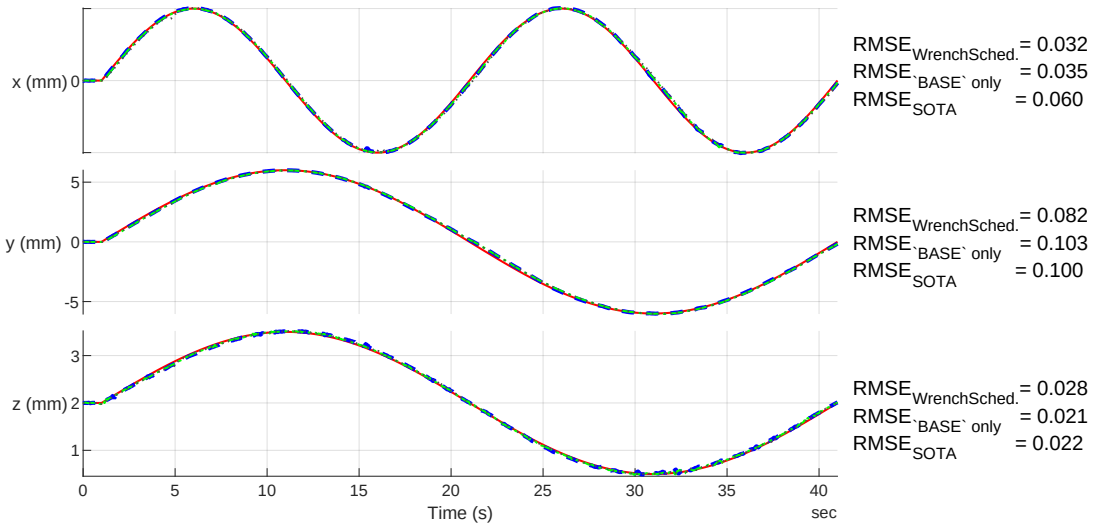
Figure 6.5: 3D trajectory tracking comparison ("oval")

As seen in these figures, the position tracking response is overall quite consistently reliable between all three control models. Notable instances of unwanted behaviour, however, can be observed in Figures 6.6 and 6.7. In the "figure eight" trajectory, **all three control models** notably fail to fully keep up with the changes in target position near the center of the workspace, where the translational velocity is highest. However, a particularly noteworthy difference in behaviour can be observed in Figure 6.7, in the corners of the rectangular path that the microrobot is moved towards at high y-speeds. The "BASE only" and "SOTA" control models are unable to reach these corners before turning to move in the x-direction; however, the "wrench scheduling" control model is able to reach that corner and perform the sudden turn with significantly greater success than with the other two methods (which do not intrinsically take varying levitation dynamics into account). With these three control models being identical *except* for their wrench matrices, and with the ML-generated wrench matrices associated with the "HIGH Y" and "LOW Y" regions (Equations 5.5 and 5.6) reporting significantly increased horizontal magnetic forces relative to input CCR perturbations: one possible interpretation of this behaviour is that "wrench scheduling" accounting for these subregions' altered levitation dynamics results in relatively decreased "control effort" within these regions.

There are a few more deviations from desired behaviour that can be observed from these figures, such as the inconsistent "jittering" away from the target path exhibited by the

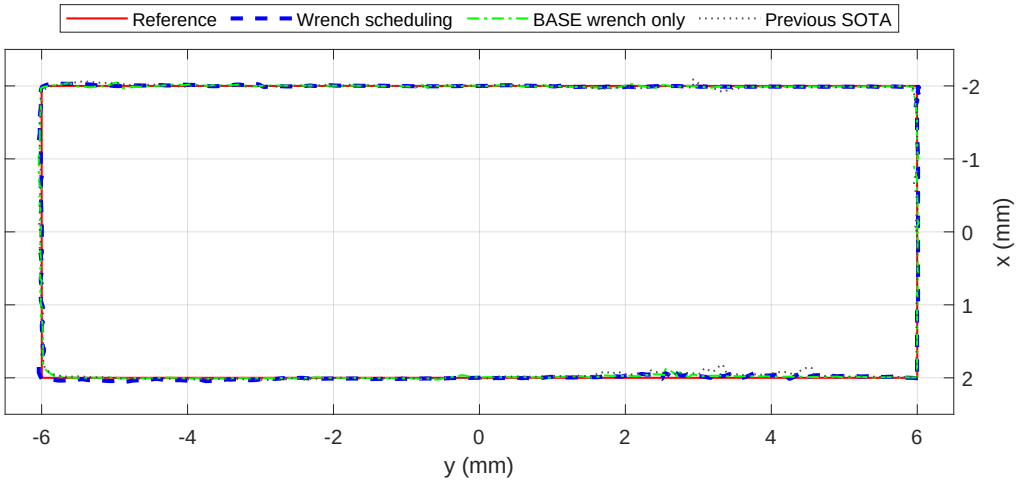


(a) Top-down view

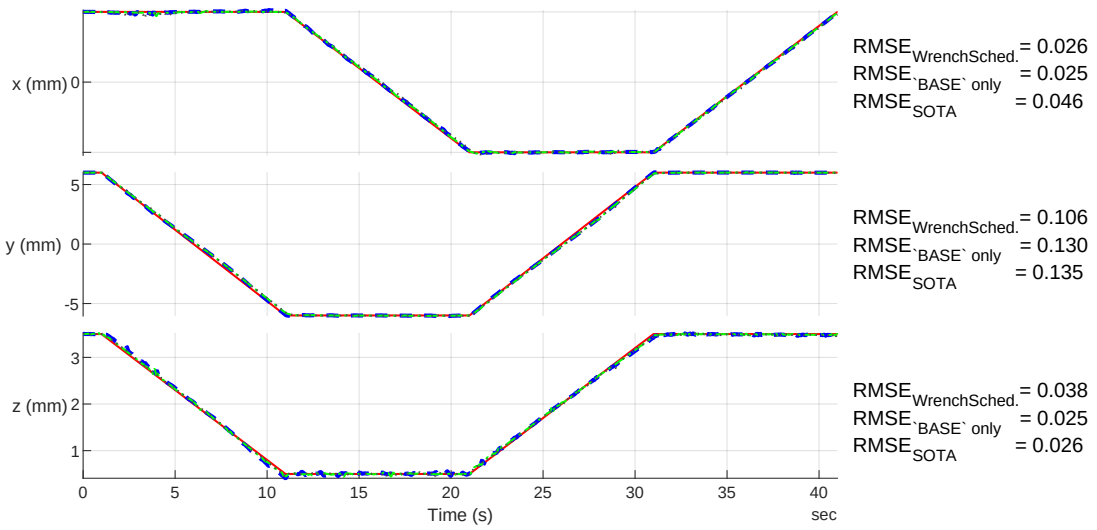


(b) Trajectory (time-series)

Figure 6.6: Trajectory tracking comparison, for "figure eight" path (RMSE in mm)

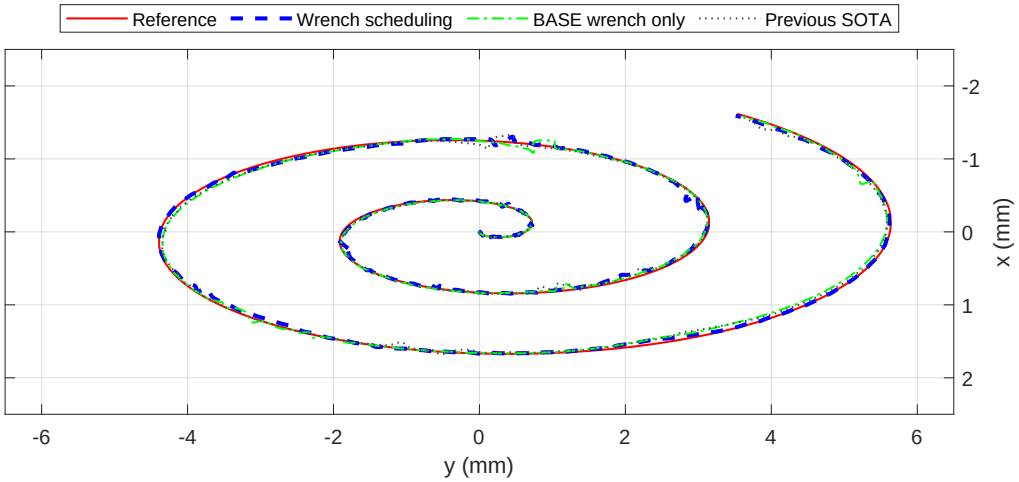


(a) Top-down view

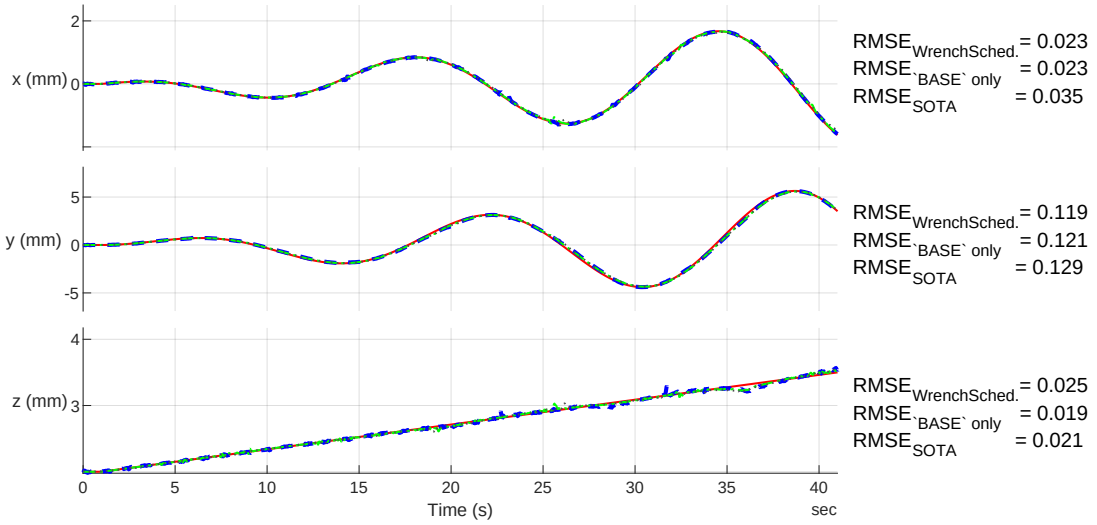


(b) Trajectory (time-series)

Figure 6.7: Trajectory tracking comparison, for “rectangle” path (RMSE in mm)

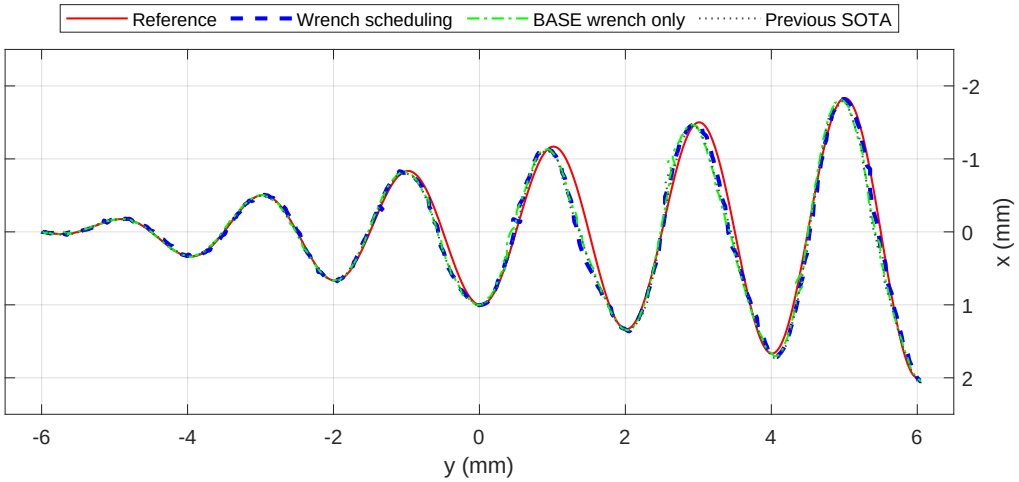


(a) Top-down view

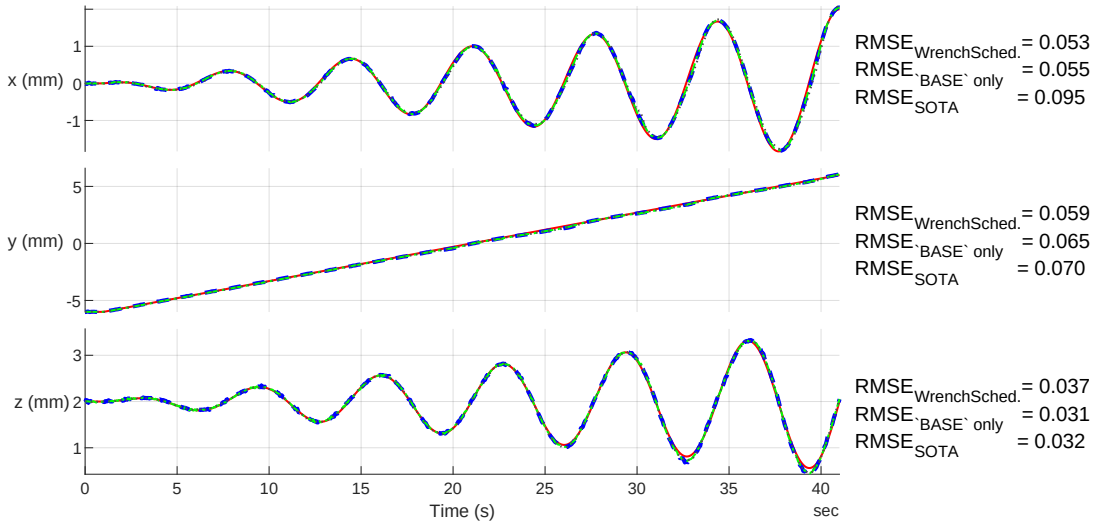


(b) Trajectory (time-series)

Figure 6.8: Trajectory tracking comparison, for "vertical helix" path (RMSE in mm)

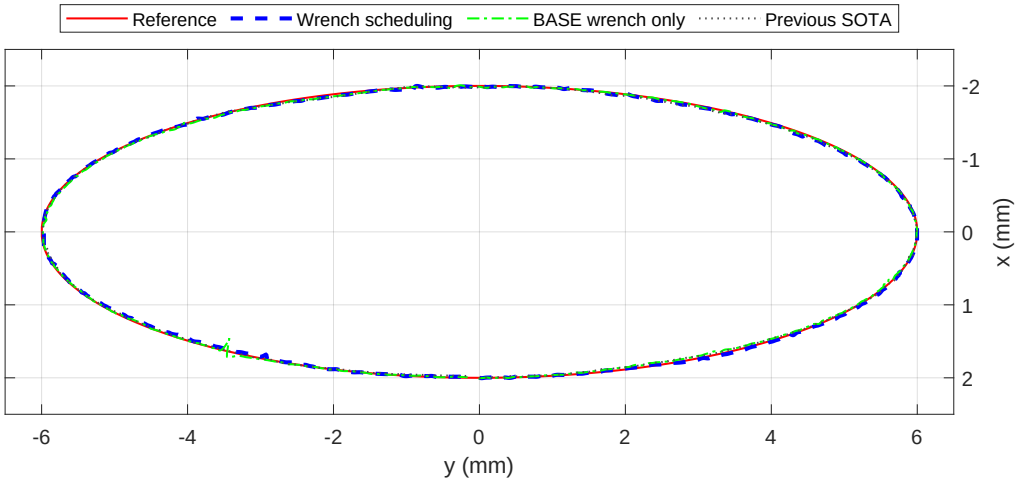


(a) Top-down view

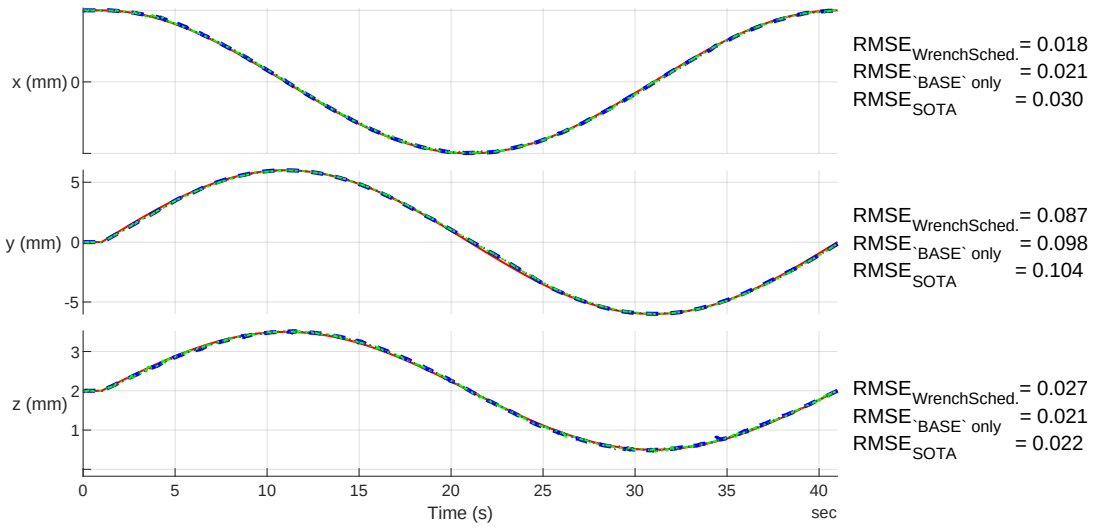


(b) Trajectory (time-series)

Figure 6.9: Trajectory tracking comparison, for "horizontal helix" path (RMSE in mm)



(a) Top-down view



(b) Trajectory (time-series)

Figure 6.10: Trajectory tracking comparison, for "oval" path (RMSE in mm)

"BASE only" model in Figure 6.8. However, the sporadic and inconsistent nature of these instances, combined with the fact that this behaviour is not repeated with the "wrench scheduled" model that should behave completely identically at that position, suggests that these are more likely to be artifacts associated with imperfect laser sensor readings and/or deterioration of the aging levitation platform than with behaviours intrinsic to the control models themselves.

In regards to the qualitative behaviours associated with transient response of the microrobot, it should be noted that when suddenly altering the desired position in one axis, the levitated microrobot tends to experience a small deviation along at least one of the other two remaining axes. This effect, where the microrobot experiences unintended movement along an axis in response to the control signal associated with another, is referred to as "cross coupling"; and is an issue intrinsic to the ATMA configuration [4] that is not inherently addressed by a simple PID controller. As such, all three control models were observed to exhibit the cross coupling issue to similar degrees.

6.1.2 Quantitative Performance Analysis

Trajectory tracking performance can be quantified using the root mean square error (RMSE) for each of the three control models, as a measurement of the average deviation between "reference" and "response" positions throughout each of the recorded trajectories. These RMSE measurements along each axis are displayed in Figures 6.6 through 6.10. Notably, the "wrench scheduling" method demonstrates consistently similar or improved RMSE measurements in comparison to the "BASE only" method along the x-y axes (particularly along the y-axis, where the RMSE error is reduced by as much as 20% depending on the trajectory), albeit at a cost to performance along the z-axis. Similarly, the "BASE only" method demonstrates consistently similar or marginally-improved performance compared to the "previous SOTA" method along the y-z axes; along with a more significant improvement in performance along the x-axis (although the latter is likely due to the difference in relative "scale" between the x- and y-columns of Equation 3.1– which, as mentioned in Section 6.1, highlights a reduced need for separately tuned x- and y-motions in the ML-assisted methods compared to the "previous SOTA" method).

The comparisons between the three control methodologies' response to sudden perturbations to target positions are aptly summarized by Figures 6.11 and 6.12. These figures illustrate the microrobot's response to discrete changes in target position along the x and y axes, respectively. For the plots in Figure 6.11, the target position was varied from $x_d = -1$ to $x_d = 1$ at $y = 5$, $y = 0$, and $y = -5$, respectively; to illustrate any differences

between the models that occur from the "wrench scheduling" method's wrenches being altered at these extreme y-positions (such that the "wrench scheduling" and "BASE only" methods are expected to behave identically at $y = 0$). Similarly, Figure 6.12 shows the transient response associated with sudden perturbations in y_d at $x = 0$; with a difference being that these measurements are *also* taken at different y-positions, such that the microrobot's response to sudden transitions in and out of the "BASE", "HIGH Y", and "LOW Y" subregions (that is, from $y_d = 2 \rightarrow 5$, $y_d = -2 \rightarrow 2$, and $y_d = -5 \rightarrow -2$ can be compared between the different models. The reverse directions are not explicitly presented here, although their behaviour is largely identical.

The final parameters associated with step response in these experiments are shown in Tables 6.1 and 6.2, respectively; where the "settling time" for each measurement is defined based on the microrobot position settling to within 1% of the target value. As seen in these tables (and as can be observed in the aforementioned figures), the performance between the three models is quite similar overall within the "BASE" region of the workspace, as expected. However, the key result can be seen within the "HIGH Y" and "LOW Y" regions; where the "wrench scheduling" method shows consistent improvement over both the "BASE only" and "previous SOTA" wrench models in terms of both rising time and settling time (demonstrating up to a 28% reduction in rise time and 36% reduction in settling time along the y-axis compared to the "previous SOTA" model). As mentioned in Section 6.1.1, these three models have the same PID gains and are entirely identical in every way save for the wrench matrices themselves; which suggests that the novel ML-assisted wrench matrix generation and wrench scheduling methods is successful in improving performance by reducing the necessary "control effort" in the affected subregions.

6.2 Future Steps for Performance Enhancement

As described in Sections 6.1.1 and 6.1.2, the novel ML-assisted wrench generation and wrench scheduling method presented in this thesis is able to perform the same navigation tasks as the previous state-of-the-art with a similar to slightly-improved level of robustness. 3-DoF navigation of the microrobot handles very similarly between these control models during levitation near the center of the workspace. However, on the edges of the workspace where the levitation dynamics are altered due to (primarily, but not necessarily exclusively) significant changes to the microrobot's proximity with each coil, the presented ML-assisted method has demonstrated improved performance over the previous state-of-the-art methodology in terms of both transient response and certain cases of position tracking (e.g., when the microrobot needs to suddenly make a sharp turn while traveling

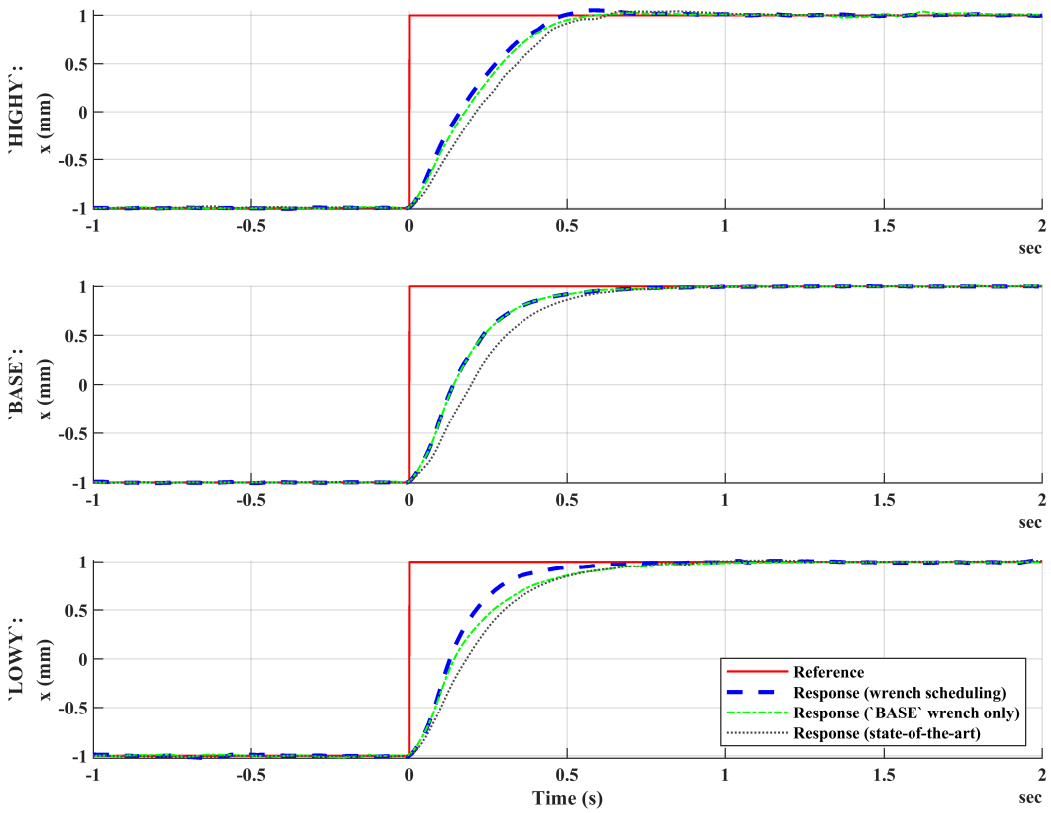


Figure 6.11: Transient response along the x-direction, within each of the "BASE", "HIGH Y", and "LOW Y" subregions

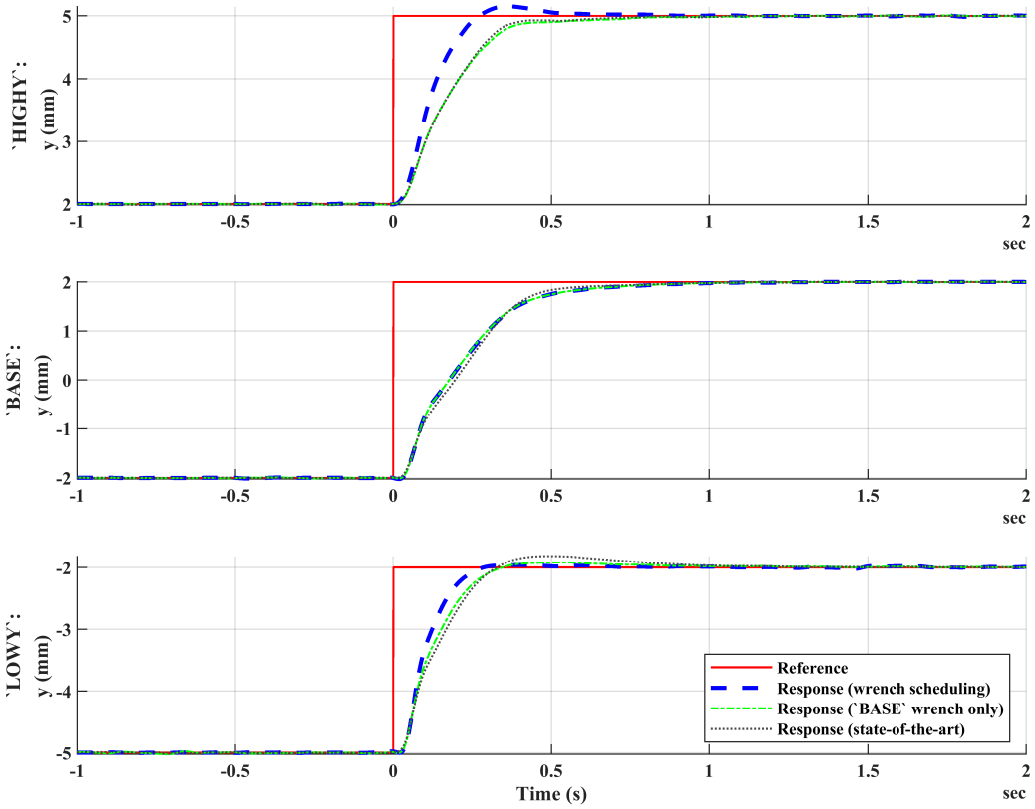


Figure 6.12: Transient response along the y-direction, both within the "BASE" subregion and in transitions between the "BASE" and "HIGH Y" / "LOW Y" subregions

Table 6.1: Comparison of "wrench matrix selection" methods for final PID controllers, for x movement in the defined subregions

		Wrench Matrix Selection		
Tested region		**"Wrench scheduling"**	**"BASE" wrench only**	**Previous SOTA**
"HIGHY"	Rise Time (s)	0.34	0.36	0.38
	Settling Time (s)	0.70	1.65	1.01
	Overshoot (%)	2.62	1.64	2.22
	Undershoot (%)	0.00	0.00	0.00
"BASE"	Rise Time (s)	0.31	0.31	0.38
	Settling Time (s)	0.73	0.80	0.80
	Overshoot (%)	0.18	0.36	0.25
	Undershoot (%)	0.00	0.00	0.00
"LOWY"	Rise Time (s)	0.28	0.39	0.39
	Settling Time (s)	0.86	0.93	0.89
	Overshoot (%)	0.43	0.35	0.98
	Undershoot (%)	0.00	0.00	0.00

Table 6.2: Comparison of "wrench matrix selection" methods for final PID controllers, for y movement *across* the defined subregions

		Wrench Matrix Selection		
Tested region		**"Wrench scheduling"**	**"BASE" wrench only**	**Previous SOTA**
"HIGHY" ↔ "BASE"	Rise Time (s)	0.17	0.27	0.26
	Settling Time (s)	0.62	0.77	0.72
	Overshoot (%)	5.18	0.37	0.10
	Undershoot (%)	0.00	0.00	0.00
"BASE"	Rise Time (s)	0.37	0.36	0.35
	Settling Time (s)	0.90	0.90	0.84
	Overshoot (%)	0.12	0.06	0.28
	Undershoot (%)	0.00	0.00	0.00
"BASE" ↔ "LOWY"	Rise Time (s)	0.15	0.20	0.21
	Settling Time (s)	0.65	1.19	1.02
	Overshoot (%)	1.52	3.02	6.15
	Undershoot (%)	0.00	0.00	0.00

at high speeds). Broadly speaking, the final control model successfully achieves 3-DoF position control and is able to consistently navigate the microrobot anywhere within the available workspace, thus serving as a useful proof of concept for the overall methodology presented by this research. With this being the case, there are still ways in which the final model presented in this thesis can be improved even further.

6.2.1 Hardware Configuration

As mentioned in Section 2.3, position determination of the microrobot is performed using a set of laser scan micrometers, one of which must be oriented vertically so as to capture the height of the magnet. Because vertical position control must be performed in a closed loop, the microrobot must always obstruct this vertical laser scanning beam for levitation to be possible; and this effectively constrains the x-range of the workspace to roughly the diameter D of the microrobot (as the microrobot's centre exceeding $x > D/2$ effectively means leaving the detection range of the z-sensor). A straightforward adjustment to the system that could greatly alleviate this issue could be to widen the microrobot; e.g., by using a wider magnet or by placing wide circular disks at the top and bottom of the magnet (such that the vertical sensor can detect the microrobot over a wider horizontal range without its centre of mass being displaced from its magnetic centroid).

Furthermore, previous research has shown that the minimum operating scale in ATMA position control is largely dependent on the precision and responsiveness of the position measurement instruments [4]. As such, "downscaling" the ATMA used in this research to scales relevant for cell injection applications would ideally involve position sensors that enable position control with $< \mu$m-scale precision; however, the scale of the noise recorded by the now-dated laser scan micrometers in this setup makes this difficult even with rigorous signal processing (see Figure 3.2). Additionally, reduction of noise through improved precision in position measurement can additionally play a crucial role in enabling precise calculation of accelerations from trajectory data, and may even enable such calculations in real-time (in the context of direct usage in micromanipulation and microforce-sensing applications, including through the use of machine learning methods). Finally, an important note for enabling position and force control on microscopic scales is that smaller objects (such as a truly microscopic levitated microrobot) have an increased surface-to-volume ratio, and thus their environmental interactions are more strongly influenced by friction and viscous forces [47]. As such, a reduction in sensor noise sufficient to enhance the precision of real-time velocity calculations may be pivotal to accounting for enhanced friction and viscous forces on small scales.

6.2.2 Position Control

Another aspect that can be improved upon, as mentioned in Section 6.1, is that suddenly changing the target position of the levitated microrobot reveals unwanted cross-coupling effects between the three axes. This effect was not observed to be significantly more pronounced in the outermost regions of the workspace compared to previous work [4]; although a more detailed investigation in future work may be recommended. The cross coupling effect of the PID controller worsening as the microrobot moves further away from the center of the workspace that was observed in [4] (which used a single wrench matrix) may have been partially alleviated in this work by accounting for differing levitation dynamics throughout the workspace; however, the presence of the cross coupling effect itself is a consequence of the physics of the levitation system itself [4] and cannot realistically be accounted for through wrench matrices alone under a standard PID control scheme.

For example, one such manifestation of cross coupling (albeit relatively minor) can be easily observed during vertical levitation in the absence of horizontal position control (e.g., when varying the target levitation height with a fixed coil current ratio). Despite the magnetic field lines in the workspace being assumed to be mostly vertical, the iron yoke affects the $\vec{B}$-field asymmetrically such that the microrobot is typically not perfectly centered between the six coils; and this horizontal B_{max} point will itself vary with height even with equally-loaded coils (let alone a constant but unequally-loaded CCR). This could in principle be alleviated by observing how various B_{max} points vary with height and accounting for this in the control model, although this would likely only account for a small subset of the cross coupling behaviours.

Another issue that can contribute to the cross coupling effect is the initial assumption from Section 3.2 that each coil contributes equally to the vertical force applied to the microrobot, such that the z-column of each wrench matrix is simply kept uniform and constant regardless of "subregion" (as described in Equation 5.3). While this is a safe assumption for levitation of a magnet near the center of an axisymmetric arrangement of coils, this assumption is less valid as the magnet moves further away from this position on the horizontal plane, and resulting discrepancies are handled by the vertical component of the PID algorithm– which, in turn, can lead to additional unwanted horizontal motions due to cross coupling. As a visualization, Figure 6.13 displays a point cloud of relative energy costs associated with levitation at a fixed height across the workspace during random data generation. While the maximum energy associated with each $[x,y]$ position can greatly vary depending on the "efficiency" with which the coils are utilized for vertical control, it can be seen that the *minimum* energy associated with these positions forms a "bowl" shape of sorts underneath this point cloud; indicating that the energy cost for stable vertical

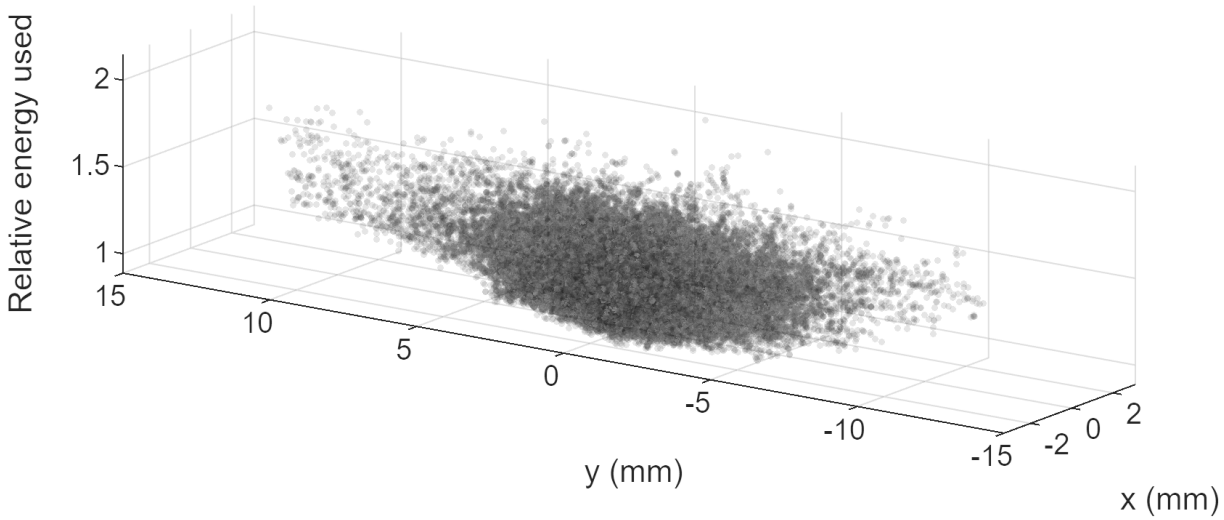


Figure 6.13: Relative energy cost across the workspace for levitation at $z_d = 2.0$mm above the platform, during the randomized data generation process

levitation tends to increase with distance from $[x, y] = [0, 0]$ even when coils are utilized with greater efficiency. Future work could account for this issue by approximating each coil's contribution to the *vertical* magnetic force across the workspace (thereby replacing the third columns in Equations 3.1, 5.4, 5.5, and 5.6); and accounting for the energy costs associated with certain CCRs across the workspace may provide additional benefits for future iterations of the machine learning experiments.

With this being said, alleviating cross coupling in a more direct manner may be beneficial for future work. The final control model presented in this paper controlled the microrobot's 3D position using a PID controller, with this method chosen for its relative ease of implementation and tuning. This is of course just one of many controllers that can be used. Previous studies have demonstrated the viability of an LQG controller for reducing cross coupling effects in this application [47], and it may be possible to expand the control model presented in this thesis in a similar manner.

6.2.3 Training Data Enhancements

There are many improvements that can be made regarding the quantity and spatial distribution of the automatically-generated training data. Perhaps the simplest possible improvement is to allow more time for generating training data, as well as to increase the

number of trained neural networks associated with each subregion (as described in Chapter 5) so as to maximize the accuracy and consistency of the trained networks' outputs in comparison to the recorded data. Expanding on this, it could be useful to fully automate the step in which subregions are defined and their respective wrench matrices are generated; possibly by separating the workspace into "subregions" in a grid-like manner (with the "granularity" of this grid dependent on the quantity and distribution of the training data) and identifying when the levitation dynamics in each subregion is sufficiently distinct from the "central" subregion that a custom wrench matrix is necessary. Spatial imbalances in the dataset can also be alleviated somewhat via increased granularity, as shrinking the boundaries defining each subregion will improve the consistency in spatial distribution observed by each subregion (contrasted with Figure 4.3, in which the datasets for the "HIGH Y" and "LOW Y" subregions are themselves unbalanced due to a relative absence of viable data in the outermost regions). Of course, a caveat to this method is that more data must be generated, to ensure that each subregion has sufficient training data in the first place.

These improvements also highlight the room for improvement in the efficiency of the training data generation algorithm. That is, improvements to the quality and/or quantity of wrench matrices across the workspace would benefit from a more time-efficient method of generating the training data. The methodology described in Chapter 4 introduces horizontal motions to the microrobot in a direction that is almost entirely random, such that the microrobot follows a "random walk"-like path during data generation and its position is reset to the center of the workspace in the event that levitation fails. The resulting dataset heavily overrepresents data near the center of the workspace, which is problematic when there is a need for observing levitation dynamics at the "outskirts" of the workspace. A simple method of generating a more "balanced" dataset could be to automatically select the "offset" parameter in Equation 4.4 such that the microrobot is encouraged to move towards regions that have seen little coverage. Alternatively, a less straightforward (but perhaps more rewarding) method could be to incorporate reinforcement learning algorithms into the training data generation loop; such that the training data generation algorithm can "learn" the effect of inputs on movement direction, and/or which combinations of inputs are more likely to result in desired behaviour such as high horizontal acceleration or undesired behaviour such as increased power cost, coil heat generation, or levitation instability.

With this being said, a potential method for improving the quantity and quality of wrench matrices under an *imbalanced* dataset may be through employing transfer learning. The improvements suggested above are all based on the current methodology of training a neural network from scratch for each defined "subregion" of the workspace. However, future work could include utilizing the trained network of the central (or "BASE") subregion as a

foundation for other subregions' neural networks; such that each subregion starts with an already relatively well-tuned network, and only modifies the weights and biases of this pre-trained network to suit its own data more accurately. In principle, this method may enable a greater quantity and/or quantity of distinct subregions (and, thus, wrench matrices) to be generated, even if the dataset associated with each subregion is relatively lacking [43].

6.3 Recommendations for Future Directions

In contrast to the previous section that described possible methods of directly improving the performance of the overall methodology and final control model presented in this thesis, this section instead aims to discuss possible applications and subsequent research steps that could follow this study.

6.3.1 Force Control

The final control model that was implemented and presented in Chapter 5 and discussed in Section 6.1 enables 3-DoF position control of the levitated microrobot. The natural next step following 3-DoF positional navigation, which would allow expansion of the presented methods to actual micromanipulation applications, is expansion of the existing position control model to 3-DoF force control.

Achieving force control would allow the levitated microrobot to be navigated with high precision in terms of both position control and the actuation force applied to the environment or objects of interest in the workspace, such as living tissue, hazardous chemicals, or small components in microassembly. As mentioned in Chapter 1.1, one possible implementation that could prove incredibly useful is a control model that dynamically switches between position control and force control modes based on the microrobot's proximity to objects, which allows for intuitive controls while preventing excessive impact forces with fragile objects such as microparts or living tissue [48]. Another implementation that has already been demonstrated in prior research is teleoperated position control of the microrobot, where "force control" takes the form of the microrobot's applied forces being communicated to the operator via haptic feedback [17]. Other previously-conducted research, such as the cooperative manipulator presented in [4] and the levitated pick-and-place robot presented in [12] have also already demonstrated the viability of position- and force-controlled ATMAs in various manipulation and micromanipulation tasks.

While there are certain applications in micromanipulation that may not strictly require direct force control of the levitated microrobot [49], enabling force control would certainly allow for a far more vast variety of potential applications (such as those mentioned above) where the microrobot directly applies a controlled contact force on its environment. While enabling force control falls beyond the scope of this study, there are a variety of methods (both preexisting and proposed) that can enable force control using our presented ML-assisted position control model as a foundation. The selection and implementation of such methods, described in the following, are left to future work.

Existing Methods of Expansion to Force Control

As mentioned above and in Chapter 1.1, expansion of horizontal position control to horizontal force control of the levitated microrobot has already been investigated and implemented in predecessors to this research, which used the more "traditional" methods of establishing initial 3-DoF position control as a foundation. These methods of enabling force control, which include dynamic prediction of the B_{max} point on the horizontal plane based on real-time measurements of the $\vec{B}$-field with Hall-effect sensors [4] [17] [36], have already demonstrated the ability to precisely measure contact forces between the levitated microrobot and an object in the workspace without the need for direct installation of microforce sensors (e.g., piezoresistive, piezocapacitive, etc.) onto the microrobot. As such, the enhanced ML-assisted position control methodology presented in this thesis could also be expanded to force control following similar steps.

This final force control model can then be used in combination with the existing position controller for manipulating objects; e.g., through the aforementioned "hybrid" approach that dynamically switches between position control and force control models based on whether the microrobot is observed to be in close proximity with an object (so as to prevent excessive impact forces) [48].

Finally, it is worth noting that both the "traditional" methods of establishing horizontal position control and the novel ML-assisted wrench generator presented in this thesis are used for generating a wrench matrix, whose purpose is to predict the relative magnetic force applied to the microrobot by each coil during stable levitation. If the wrench matrix were to be a perfect descriptor of the magnetic forces applied to the microrobot across the full workspace, then it could in principle be possible to enable force control by simply calibrating the PID signals to the actual steady-state forces that the microrobot applies to its environment. In practice, however, the wrench matrices generated via both the "traditional" and the currently-presented methods are only approximate predictions of magnetic force (as opposed to describing forces precisely enough that open-loop horizontal

position control could be performed precisely). This means that while enabling force control via calibrations in this way is technically possible, these force predictions would likely have very limited precision in practice compared to a more thorough and methodical approach; so its usefulness for micromanipulation may be limited accordingly. Improving the accuracy and precision with which the wrench matrices describe the levitation dynamics in the workspace (e.g., by applying improvements to the presented methodology as described in Section 6.2) may alleviate these limitations.

Force Control via Acceleration Calculations

The methodology for automatically generating training data could also be used to enable force control in a more direct manner by calculating the 3D *accelerations* throughout the automatically-generated recordings, and then using this information to calculate the magnetic force experienced by the microrobot in response to system inputs. As described in Chapter 4, the training data that were recorded and used in this thesis were generated using an aluminum holding block just below the microrobot, which would act as a magnetic brake for the purpose of speeding up generation of training data (since each "step" in this process ends when the B_{max} location is found, and this is done by observing the microrobot come to a stop at that location). These braking effects would need to be accounted for (e.g., as modelled in [42]) so that the magnetic force can be calculated.

Acceleration calculations from trajectory data are highly impacted by noise, meaning that proper filtering is necessary to eliminate noise while keeping actual signal intact. However, the amount (and type) of filtering used can vary based on circumstance. For example, a microrobot making large and gradual motions may have the most accurate acceleration calculations when paired with a low-pass filter with a very low cutoff frequency (such that noise is eliminated while the actual motions of the microrobot are not removed by the filter); but this same filter would be unable to keep up with the microrobot if it were to experience a sudden and strong acceleration in some direction during its motion, since this motion itself would be filtered out.

There are several possible directions that could enable accelerations to be calculated with more precision, beyond simply upgrading the position sensors themselves. It is in principle possible to classify the microrobot's motion using machine learning and apply different filters accordingly, and future work could include implementing this following the precedent set in a recent study for improving vehicle acceleration calculations from positional data based on a combination of clustering and deep learning [50].

However, there could be an alternative with less strict requirements regarding signal

processing methods. Removing the aforementioned magnetic brake from the ATMA would remove the main source of damping, causing the magnet to behave like a minimally-damped simple harmonic oscillator about each newly introduced B_{max} point during training data generation. The data generation algorithm could then be modified to continuously fit the microrobot's ongoing trajectory to a more "physically informed" model, such that the B_{max} point could be accurately predicted during data generation through these fits alone. This would have the benefit of enabling fast data generation for the presented methodology without the need for a magnetic brake. However, the most important benefit of this alteration is that the final recorded trajectories on the x-y plane would (in principle) behave as a minimally damped simple harmonic oscillator model; and fitting the trajectories to this model as described in Section 5.4 could thus enable highly precise calculation of the acceleration-versus-time curves with less reliance on reducing signal-to-noise ratio via signal processing methods.

If acceleration-versus-time curves were to be calculated following these methods (or something similar), could then be used to calculate magnetic forces applied to the microrobot, and this information could be used for subsequent experiments that employ ML to directly enable force control.

6.3.2 Open-Loop Position Control

While largely uninvestigated in this research, it may be possible to use ML to enable horizontal position control in an open loop. As first mentioned in Chapter 2, there is a "hysteresis" in between the input control signal (i.e., the voltage sent to the amplifier) and the actual magnetic field throughout the workspace, which prevents the horizontal B_{max} position from being directly associated with an input coil current ratio. Furthermore, the behaviour of this hysteresis effect appears to be affected by changes in temperature, which may need to be accounted for if the hysteresis effect is to be quantified with reasonable repeatability [4]. If a future study is done to investigate and quantify the behaviour of the hysteresis effect (whether from a theoretical basis or from an empirical, observation-based, ML-assisted methodology) such that it can be automatically "accounted for" during real-time operation, then this could open up a variety of new possibilities for controlling the horizontal microrobot position in the absence of precise position feedback, e.g., by simply training a machine learning model to directly map between CCRs and the resulting B_{max} positions that the microrobot gravitates towards.

While this can already be accomplished using Hall-effect sensors as an intermediate step, doing so still requires continuous "feedback" in the form of the coils' $\vec{B}$-fields measured

by optimally-placed Hall-effect sensors [4] [17] [39]. As such, a hypothetical model that could account for (and eliminate) the hysteresis effect would be able to achieve the same task without the need for optimizing the Hall-effect sensor placement and fitting their readings to the B_{max} point of the magnet. To reiterate, successful implementation of such a model could enable x-y position control without the need for position sensors (which can be applicable in applications involving optical blockage), all without needing to do any sort of modelling or optimization beyond the initial creation of the ATMA and the PID tuning.

Chapter 7

Conclusion

Untethered microrobots that are levitated by an attractive-type maglev apparatus (ATMA) have the potential to solve many problems faced by traditional micromanipulation technologies. The overall motivation for this research stems from the accessibility barrier formed by the complexity and esoteric nature of establishing position control in at least 3 degrees of freedom for a newly constructed or modified ATMA, which can reduce the practical feasibility of utilizing such a device (in terms of prerequisite expertise, time investment, and/or financial expenses) in areas of research that could otherwise benefit strongly from such a device. This accessibility barrier is caused by the complexity in the magnetic fields (and resulting levitation dynamics) associated with a typical ATMA. These magnetic fields dictate the motion of the levitated magnetized microrobot, but are impossible to precisely model analytically and are typically obtained via difficult analytical estimations, magnetic field measurements using precision instruments, and/or simulations with (often expensive) proprietary software.

This research aimed to automate this process of modelling system dynamics using machine learning, without the need for intensive math, manual magnetic field measurements, or simulations. In this work, we introduced a novel methodology for establishing horizontal position control for an ATMA-levitated microrobot, using a machine learning-assisted approach that succeeded in approximating the relative horizontal magnetic force applied to the levitated microrobot by each coil (referred to as the "wrench matrix" throughout this study) as they vary throughout the workspace. The wrench matrices were generated based on recorded data associated with various "subregions" in the workspace based on their distinct levitation dynamics, and the varied levitation dynamics throughout the workspace were accounted for through real-time interpolation between these wrench matrices based on

the microrobot's position (referred to in this work as "wrench scheduling", as an analogue to PID gain scheduling methods).

Additionally, we introduced a methodology for generating the required training data by observing the hardware's response to system inputs during levitation, performed in a fully automated manner and requiring little to no human intervention. These experiments were conducted using a typical ATMA structure, composed of a magnetic circuit arranged in a C-shape (similar to a horseshoe magnet) with levitation driven by 6 pairs of vertically-oriented solenoids arranged hexagonally at the top of the circuit's air gap. The 3D position of the levitated microrobot was captured using laser scan micrometers, and the effective workspace over which levitation was possible (and over which training data were generated and applied) spanned the full detection range of this position detection system.

Results showed that this ML-assisted method of establishing 3-DoF position control performed broadly on par with the traditional position control method used in the predecessor of this work near the center of the workspace, and with consistent improvement over the previous methodology in regions of the workspace that have significantly differing levitation dynamics (e.g., due to altered proximity between the microrobot and the magnetic coils). The enhanced performance in these regions was observed in the context of both position tracking and transient response capabilities of the navigated microrobot, where the novel "wrench scheduling" method in tandem with the ML-assisted wrench matrix generation method consistently demonstrated improved tracking (by up to a 20% reduction in RMSE), as well as shorter rising times and settling times (up to a 28% and 36% reduction, respectively) along the primarily-affected axis compared to traditional methods. The enhanced performance of the final ML-assisted control model compared to the state-of-the-art method displays the validity of the presented methodology, and serves as a valuable proof-of-concept by which precise 3-DoF position control can be established for ATMAs without the intensive analytical, experimental, and/or numerically-simulated magnetic field modelling that traditional methods employed.

The presented method is not flawless, of course. This work did not directly account for the "hysteresis effect" that occurs between input coil currents and final magnetic fields in the workspace (which makes precise open-loop horizontal position control impossible without additional equipment); and this hysteresis effect accordingly impacted the consistency of the machine learning models trained on the procedurally-generated dataset. Machine learning (and subsequent wrench matrix generation) proceeded under the assumption that any hysteresis-attributed inconsistencies would cancel each other out as noise would; however, this has the weaknesses of requiring more training data for producing each meaningful wrench matrix, as well as necessitating the use of multiple "redundant" ML models to eliminate inconsistent behaviours within each final model (which increases computation time

for generating each wrench matrix). Furthermore, the final 3-DoF control model presented in this work was a simple PID controller cannot intrinsically account for cross-coupling effects between the system inputs; leading to noticeable deviations from desired 3D positions when a control input is suddenly introduced. These issues could be alleviated in future work by simply generating more data over time; however, it could also be prudent to modify the data generation algorithm to be more time-efficient in terms of workspace coverage. Furthermore, automating the selection of "subregions" with distinct wrenches (based on identification of differences in levitation dynamics) could improve the reliability and robustness of the wrench matrices produced by its training data. Finally, accounting for the cross-coupling behaviours through implementation of more advanced controllers (as opposed to simple PID) could greatly improve the reliability of control models generated through our presented ML-assisted method.

In terms of future applications of this work, an important next step could be expansion to force control; either through existing methods presented in prior studies (e.g., calculating horizontal magnetic forces based on predictions of B_{max} positions using Hall-effect sensors, as was done in [36]), or through continued experiments to predict magnetic forces without the need for installation of additional sensors (e.g., predicting either acceleration [and therefore magnetic force] from changes in CCR inputs, or predicting B_{max} position from system inputs in a manner that accounts for the hysteresis effect). The currently-presented method for wrench matrix generation technically predicts the magnetic force introduced by changes in coil currents; although this is a very approximate prediction of horizontal magnetic force, so even calibrating the horizontal PID signal to actual magnetic force may have correspondingly limited precision compared to a more careful and methodical approach.

In sum, this work was successfully able to automate most of the process of establishing 3-DoF position control of a levitated microrobot, with measurable improvements to performance over conventional methods while being drastically less demanding to the researcher in terms of time, effort, requisite knowledge, and/or financial barriers. The success of our automated ML-assisted methodology paints a highly optimistic picture of the future of micromanipulation, as it eliminates most of the difficulty in enabling 3-DoF position control in an ATMA-levitated microrobot. This reduces the accessibility barriers associated with maglev micromanipulation technology, and subsequently opens up a wealth of interdisciplinary applications of maglev micromanipulation, paving the way for advancements in areas such as microassembly, materials science, and biomedical engineering.

References

[1] "Failure Analysis - Electrical Micro-probing Measurements." https://www.mpi-corporation.com/ast/applications/failure-analysis/.

[2] Y. Sun and B. J. Nelson, "Biological Cell Injection Using an Autonomous Micro-Robotic System," *The International Journal of Robotics Research*, vol. 21, pp. 861–868, Oct. 2002.

[3] Q. Xu, *Micromachines for Biological Micromanipulation*. Cham: Springer International Publishing, 2018.

[4] X. Zhang, *Cooperative Manipulation Using a Magnetically Navigated Microrobot and a Micromanipulator*. PhD thesis, University of Waterloo, Dec. 2017.

[5] B. J. Nelson, I. K. Kaliakatsos, and J. J. Abbott, "Microrobots for minimally invasive medicine," *Annual Review of Biomedical Engineering*, vol. 12, pp. 55–85, Aug. 2010.

[6] Z. Zhang, X. Wang, J. Liu, C. Dai, and Y. Sun, "Robotic Micromanipulation: Fundamentals and Applications," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 181–203, May 2019.

[7] F.-S. Quan and K. S. Kim, "Medical applications of the intrinsic mechanical properties of single cells," *Acta Biochimica et Biophysica Sinica*, vol. 48, pp. 865–871, Oct. 2016.

[8] Y. Wei and Q. Xu, "An overview of micro-force sensing techniques," *Sensors and Actuators A: Physical*, vol. 234, pp. 359–374, Oct. 2015.

[9] A. Ghanbari, B. Horan, S. Nahavandi, X. Chen, and W. Wang, "Haptic Microrobotic Cell Injection System," *IEEE Systems Journal*, vol. 8, pp. 371–383, June 2014.

[10] Z. Lu, P. Chen, and W. Lin, "Force Sensing and Control in Micromanipulation," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 36, pp. 713–724, Nov. 2006.

[11] M. Savia and H. Koivo, "Contact Micromanipulation—Survey of Strategies," *IEEE/ASME Transactions on Mechatronics*, vol. 14, pp. 504–514, Aug. 2009.

[12] E. Shameli, D. G. Craig, and M. B. Khamesee, "Design and implementation of a magnetically suspended microrobotic pick-and-place system," *Journal of Applied Physics*, vol. 99, p. 08P509, Apr. 2006.

[13] Z. Xu, C. Trakarnchaiyo, C. Stewart, and M. B. Khamesee, "Modular Maglev: Design and implementation of a modular magnetic levitation system to levitate a 2D Halbach array," *Mechatronics*, vol. 99, p. 103148, May 2024.

[14] X. Zhang, C. Trakarnchaiyo, H. Zhang, and M. B. Khamesee, "MagTable: A tabletop system for 6-DOF large range and completely contactless operation using magnetic levitation," *Mechatronics*, vol. 77, p. 102600, Aug. 2021.

[15] N. Arora, M. U. Khan, L. Petit, F. Lamarque, and C. Prelle, "Design and Development of a Planar Electromagnetic Conveyor for the Microfactory," *IEEE/ASME Transactions on Mechatronics*, vol. 24, pp. 1723–1731, Aug. 2019.

[16] "Strong Rare Earth Neodymium Magnets - Low prices, Fast shipping, Huge inventory | K&J Magnetics." https://www.kjmagnetics.com.

[17] M. Mehrtash, *Bilateral Macro-Micro Teleoperation Using A Magnetic Actuation Mechanism*. PhD thesis, University of Waterloo, Sept. 2013.

[18] S. Sabihuddin, *Investigation of Diamagnetic Bearings and Electrical Machine Materials for Flywheel Energy Storage Applications*. PhD thesis, The University of Edinburgh, July 2018.

[19] R. E. Cohen, D. Lide, and G. Trigg, *Physicist's Desk Reference*. Springer Science & Business Media, Jan. 2003.

[20] P. Kazemzadeh Heris and M. Khamesee, "Design and Fabrication of a Magnetic Actuator for Torque and Force Control Estimated by the ANN/SA Algorithm," *Micromachines*, vol. 13, p. 327, Feb. 2022.

[21] A. Caciagli, R. J. Baars, A. P. Philipse, and B. W. Kuipers, "Exact expression for the magnetic field of a finite cylinder with arbitrary uniform magnetization," *Journal of Magnetism and Magnetic Materials*, vol. 456, pp. 423–432, June 2018.

[22] E. E. Callaghan and S. H. Maslen, "The Magnetic Field of a Finite Solenoid," Tech. Rep. NASA-TN-D-465, Oct. 1960.

[23] G. Trebbin, "Off Axis Magnetic Field of a Circular Current Loop."

[24] "Magnet Glossary | K&J Magnetics." https://www.kjmagnetics.com/glossary.asp.

[25] "Neodymium Magnet Specifications | K&J Magnetics." https://www.kjmagnetics.com/neodymium-magnet-specifications.asp.

[26] T. Nakamura and M. Khamesee, "A prototype mechanism for three-dimensional levitated movement of a small magnet," *IEEE/ASME Transactions on Mechatronics*, vol. 2, pp. 41–50, Mar. 1997.

[27] N. Bamps, P. Dular, A. Genon, H. Hedia, W. Legros, and A. Nicolet, "Corner Modelling for BEM in Magnetostatic," in *Boundary Elements XIII* (C. A. Brebbia and G. S. Gipson, eds.), pp. 397–405, Dordrecht: Springer Netherlands, 1991.

[28] E. Deeley, "Avoiding surface impedance modification in BE methods by singularity-free representations," *IEEE Transactions on Magnetics*, vol. 28, pp. 2814–2816, Sept. 1992.

[29] E. Deeley and J. Xiang, "The use of modified surface impedance in a minimum-order boundary element method," *IEEE Transactions on Magnetics*, vol. 26, no. 5, pp. 2762–2764, Sept./1990.

[30] "dSPACE DS2004 Data Sheet."

[31] "dSPACE DS2103 Data Sheet."

[32] "Artisan Technology Group Keyence LS-5501 Built-In Display Type Controller."

[33] "Sensor Head - LS-5041 | KEYENCE Canada." https://www.keyence.ca/products/measure/micrometer/ls-5000/models/ls-5041/.

[34] D. Craig, *Modeling and Control of a Magnetically Levitated Microrobotic System*. PhD thesis, University of Waterloo, 2006.

[35] D. Craig and M. Khamesee, "Derivation of an analytical model for the force produced during the motion of a magnetically suspended object," in *IEEE International Conference Mechatronics and Automation, 2005*, vol. 2, (Niagara Falls, Ont., Canada), pp. 970–974, IEEE, 2005.

[36] M. Mehrtash, M. B. Khamesee, N. Tsuda, and J.-Y. Chang, "Motion control of a magnetically levitated microrobot using magnetic flux measurement," *Microsystem Technologies*, vol. 18, pp. 1417–1424, Sept. 2012.

[37] S. Hosseini and M. B. Khamesee, "Design and control of a magnetically driven capsule-robot for endoscopy and drug delivery," in *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, (Toronto, ON, Canada), pp. 697–702, IEEE, Sept. 2009.

[38] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Pub.., 1999.

[39] M. B. KHAMESEE and M. MEHRTASH, "System and method for providing force information in a magnetic field environment," Sept. 2014.

[40] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic Neural Networks: A Survey," Dec. 2021.

[41] "PID Tuner." https://pidtuner.com/assets/snapshot/snapshot.png.

[42] C. Elbuken, E. Shameli, and M. B. Khamesee, "Modeling and Analysis of Eddy-Current Damping for High-Precision Magnetic Levitation of a Small Magnet," *IEEE Transactions on Magnetics*, vol. 43, pp. 26–32, Jan. 2007.

[43] A. Burkov, *The Hundred-Page Machine Learning Book by Andriy Burkov*.

[44] "Movmean - Moving mean - MATLAB Documentation."

[45] "Signal Smoothing - MATLAB & Simulink Example." https://www.mathworks.com/help/signal/ug/signal-smoothing.html.

[46] "Improve Shallow Neural Network Generalization and Avoid Overfitting - MATLAB & Simulink." https://www.mathworks.com/help/deeplearning/ug/improve-neural-network-generalization-and-avoid-overfitting.html.

[47] M. Mehrtash and M. B. Khamesee, "Design and implementation of LQG\LTR controller for a magnetic telemanipulation system-performance evaluation and energy saving," *Microsystem Technologies*, vol. 17, pp. 1135–1143, June 2011.

[48] Y. Zhou, B. Nelson, and B. Vikramaditya, "Fusing force and vision feedback for micromanipulation," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 2, (Leuven, Belgium), pp. 1220–1225, IEEE, 1998.

[49] C. Elbuken, M. Khamesee, and M. Yavuz, "Design and Implementation of a Micromanipulation System Using a Magnetically Levitated MEMS Robot," *IEEE/ASME Transactions on Mechatronics*, vol. 14, pp. 434–445, Aug. 2009.

[50] H. Luo, X. Hu, and L. Huang, "A Hybrid Model for Vehicle Acceleration Prediction," *Sensors*, vol. 23, p. 7253, Aug. 2023.