

© Copyright by Benedict Isichei 2018
All Rights Reserved

ROBOTIC MANIPULATORS:
CRYOGENIC MAGNETIC MANIPULATION SYSTEM,
TWO-BODY MAGNETIC MANIPULATION, AND LOW-COST
ROBOTIC MANIPULATORS

A Thesis

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

in Partial Fulfillment

of the Requirements for the Degree

Master of Science

in Electrical Engineering

by

Benedict Isichei

Aug 2018

ROBOTIC MANIPULATORS:
CRYOGENIC MAGNETIC MANIPULATION SYSTEM,
TWO-BODY MAGNETIC MANIPULATION, AND LOW-COST
ROBOTIC MANIPULATORS

Benedict Isichei

Approved:

Chair of the Committee
Aaron T. Becker, Assistant Professor
Department of Electrical and Computer Engineering

Committee Members:

Joseph Reich, Assistant Professor
Department of Electrical and Computer Engineering

Daniel Corder, Professor
Department of Computer Science

Suresh K. Khator, Associate Dean,
Cullen College of Engineering

Badrinath Roysam, Professor and Chair,
Electrical and Computer Engineering

Acknowledgements

I would like to thank Dr. Aaron Becker for his patience, advice and support while I was writing this. Thank Dr. Julien Leclerc and Dr. Shiva Shahrokhi for their patience and advice while I was working on their respective projects. I would also like to thank my Parents for their love, encouragement and support during the entire project. Lastly, I'm extremely grateful to the various members of the swarm robotics lab for being there when I had issues and having no problems stepping in with explanations.

ROBOTIC MANIPULATORS:
CRYOGENIC MAGNETIC MANIPULATION SYSTEM,
TWO-BODY MAGNETIC MANIPULATION, AND LOW-COST
ROBOTIC MANIPULATORS

An Abstract

of a

Thesis

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

in Partial Fulfillment

of the Requirements for the Degree

Master of Science

in Electrical Engineering

by

Benedict Isichei

Aug 2018

Abstract

In the field of robotics, there are various control systems utilized depending on the kind of robot being considered. This project presents three different robotic manipulators, and then compares their control systems. The first system is a cryogenic manipulation system. In this system, the robotic workspace is a 150mm³ cube, and the system implements a 3-dimensional control on the robot's position and orientation. To obtain accurate results, and obtain higher torque without a solid Iron core, this system has all its coils cooled by liquid nitrogen to achieve a higher magnetic field without damaging the coils. In the second system, a 2-directional position control method is implemented. This system is created to simulate wall contacts, and implement an algorithm for navigating magnetic robots in a cavity (2- dimensions in this case) given the constraints of the cavity's geometry, and global inputs to the robots. The last system is a robotic manipulator created out of a toy robotic arm kit. The aim here was to increase the robot's accuracy and augment its functionality, while making it userfriendly to students new to robotics.

Table of Contents

Acknowledgements	v
Abstract	vii
Table of Contents	viii
List of Figures	xi
List of Tables	xvi
1 Introduction	1
2 A Magnetic Manipulator Cooled With Liquid Nitrogen	2
2.1 INTRODUCTION	2
2.2 System Presentation	5
2.3 Cooling with Liquid Nitrogen	8
2.3.1 Motivations	8
2.3.2 Scaling Law	10
2.4 Inverse Magnetics	15
2.4.1 Forward problem	15
2.4.2 Inverse problem	17
2.5 Trajectory control	18
2.6 Singularities analysis for a 3-DOF robot	20

2.7	Experimental Results	21
2.8	Conclusion and Future Works	24
3	Exploiting Non-Slip Wall Contacts to Position Two Particles Using The Same Control Input	26
3.1	Introduction	27
3.2	Related Work	28
3.3	Theory	30
3.3.1	Boundary Interaction Model	30
3.3.2	Example: Shortest Path in a Square Workspace	31
3.3.3	Shortest Path in Unit Disk that Intersects Circumference	32
3.4	Position Control of Two Particles Using Boundary Interaction	34
3.4.1	Δ Configuration Space	35
3.4.2	Two Particle Path Planning	36
3.4.3	Convex Polygonal Workspaces: 2-Move Reachable Set	36
3.4.4	Circular Workspaces: 2-Move Reachable Set	40
3.4.5	3D workspaces: Cylinders and Prisms	43
3.5	Simulation	44
3.6	Experimental Results	49
3.6.1	Magnetic Manipulation Setup	49
3.6.2	Intestine Phantom Model	50
3.6.3	Bovine Stomach Cross-section	50
3.7	Conclusion and Future Work	51

4 Low-Cost Robot Manipulator Arms	52
4.1 Lab 1: Assembly and familiarization	53
4.2 Lab 2: Open loop control	53
4.3 Lab 3: Closed-loop control with Potentiometer sensors	54
4.4 Lab 4: Closed-loop control with image processing	54
4.5 Pre-semester prep	54
4.6 Results	55
5 Conclusion	56
References	57

List of Figures

2.1	Picture of the magnetic manipulator while functioning. Inset shows view from the camera used to obtain position feedback of the robot.	3
2.2	CAD model of the magnetic manipulator: (a) exploded view of three cryostats, (b) cross-section view.	7
2.3	Block diagram of (a) the physical hardware and (b) the controller used for position control.	8
2.4	3D representation of a dual-coil collinear EM assembly.	10
2.5	Maximum sustainable magnetic flux density by different EM designs. Magnet specifications are given in Table 2.1.	11
2.6	Plot of the winding volume (a) and power consumption (b) of air-cooled (300K) and LN2-cooled (77K) EMs. Red curves correspond to magnetic assemblies able to produce a gradient strength of 20 mT/m at the system center. Blue curves correspond to magnetic assemblies able to produce a gradient strength of 45 mT/m at the system center.	14
2.7	Map of the conditioning of the actuation matrix \mathbf{A}_0 for (a) $\theta = 0$ and (f) $\theta = \pi/4$. Map of the norm of the total current needed to produce a force of 100 mN along the d axis (plots b, d, g and i) and along the q axis (plots c, e, h and j). Plots b, c, d and e are plotted for $\theta = 0$ and plots g, h, i and j are plotted for $\theta = \pi/4$. Plots b, c, g and h were calculated using \mathbf{A}_0 and plots d, e, i and j were calculated using \mathbf{A}_1	19
2.8	Plot of the trajectory of the robot obtained experimentally. For this dataset, the robot completed the path ten times. The robot and the workspace are shown at right.	22

2.9	Voltage, current, and power used during robot navigation. The curves show 60 seconds of data, which correspond to the robot completing the trajectory in Fig. 2.8 three times.	23
3.1	Workspace and magnetic setup for an experiment to position particles that receive the same control inputs, but cannot move while a control input pushes them into a boundary.	28
3.2	The shortest three-move path that reconfigures two particles has the property that the incident angle equals the reflected angle.	31
3.3	Frames from an implementation of Alg. 1: two particle positioning using walls with non-slip contacts. Particle start positions are shown by squares, and goal positions by circles. Dashed lines show the shortest route if particles could be controlled independently. Solid arrows show path given by Alg. 1.	33
3.4	The shortest path between two points S to E in the unit disk that intersects the circumference. The path length as a function of intersection point, $P = (\cos \theta, \sin \theta)$ is shown at right. See [?].	34
3.5	Workspace and Δ configuration space is shown for an arbitrary convex polygon with $n = 4$ sides. The 2-move reachable sets for this initial configuration (shown by square markers) are drawn in transparent blue.	37

3.6	The Δ configuration space is all possible configurations of $p_2 - p_1$. The sets reachable in two moves, called <i>2-move reachable sets</i> , are drawn with transparent blue polygons. A polygon with n sides has n 2-move reachable sets, but if n is even and the polygon is regular, half the reachable sets overlap. If Δg is in the 2-move reachable sets, we can achieve the required relative position in two moves. If Δg is not in the 2-move reachable set, we define a temporary goal Δg_c (the closest point on the 2-move reachable set to Δg) and apply two moves to achieve Δg_c . We repeat this process until the relative goal position is achieved.	38
3.7	Boundary interaction is used to change the relative positions of the particles. Each particle gets the same control input. (left) If particle 2 contacts the bottom wall before particle 1 reaches a wall, particle 2 can reach anywhere along the green line, and particle 1 can move to anywhere in the shaded area. (middle) Similarly, if particle 2 contacts the right wall before particle 1 reaches a wall, particle 2 can reach anywhere along the green line, and then particle 1 can move to anywhere in the shaded area. (right) The 2-move reachable sets in the Δ configuration space is shown. . . .	39
3.8	Steps to generate the 2-move reachable set when one particle collides with edge $\overline{p_i p_{i+1}}$ of a convex polygonal workspace.	39

3.9 Left top: The possible first contact points for the blue and red particles are shown with blue and red arcs. Left bottom: if the blue particle touches the wall at ψ_{\min} (blue square) the other particle (red square) can move anywhere in the red region. Right bottom: if the blue particle touches the wall at $\psi = \frac{17\pi}{10}$ (blue square) the other particle (red square) can move anywhere in the green region. Right: The Δ configuration space for the corresponding starting positions of the particles is shown. The possible 2-move reachable sets before contact are shown in the Δ configuration as a blue region. If the blue particle contacts the boundary at ψ_{\min} , the reachable Δ configuration is the red set, or the green set if $\psi = \frac{17\pi}{10}$	41
3.10 Illustration on how boundary contacts enable 3D positioning. Once one particle contacts a boundary, the other particle's 2-move reachable set is a prism formed by extending the 2D 2-move reachable set in the $\pm z$ direction.	44
3.11 Frames from reconfiguring two particles. Top six images show a polygonal workspace and the corresponding Δ configuration space with its 2-move reachable sets. Bottom six images show a disk-shaped workspace and the corresponding Δ configuration space with its 2-move reachable sets. For each the move 1 and 3 are simple translations of both particles and so the reachable sets do not change. The reachable set morphs during move 2 because one particle is held stationary by the boundary. See multimedia attachment for animations of each.	45
3.12 Plots show performance with one goal on the boundary.	46
3.13 The worst-case path length occurs when particles must swap antipodes. This can never be achieved but can be asymptotically approached. Plot shows decreasing error as the number of moves grows.	47

3.14 Starting positions of particles 1 and 2 and goal position of particle 2 are fixed, and $\epsilon = 0.001$. The top row of contour plots show the distance if particle 1's goal position is varied in x and y . The middle row shows the number of moves required for the same configurations. The bottom row shows the same configuration but when the particles are interchangeable.	48
3.15 Frames showing particle positions before and after control inputs. Top row: small intestine phantom. Bottom row: cow stomach tissue.	49
4.1 The unmodified Robotic arm	52
4.2 The parts of the robot arm before assembly	53
4.3 The Assembled Robotic arm	53
4.4 The Augmented Robotic arm	54

List of Tables

Chapter 1

Introduction

Here begins the body of the thesis.

Chapter 2

A Magnetic Manipulator Cooled With Liquid Nitrogen

This chapter is based on a Journal paper with Julien Leclerc, Benedict Isichei, and Aaron T. Becker. My contribution was Nathan Hui aided with editing the final version of the paper.

2.1 INTRODUCTION

Magnetic actuation enables non-contact manipulation from a distance. This paper describes a magnetic manipulator, shown in Fig. 2.1, that uses coils cooled by liquid nitrogen to reduce the size and power required to generate dynamic magnetic fields. Magnetic technologies have promise for several areas, including as actuation for minimally invasive surgery [?, ?]. Minimally invasive surgeries reduce patient recovery time, pain, and risks of infection [?]. These procedures are most often performed using a catheter, a tubular device that can access remote areas of the body through small incisions. If all catheter actuation is applied at the proximal end, it is increasingly difficult to accurately control force and orientation of the distal tip as the length increases.

Magnetic actuation can be used to improve the effectiveness of minimally invasive surgeries. The company Stereotaxis [?] manufactures a magnetic system able to control the tip of a magnetic catheter and therefore increase the precision of the medical procedure. The device uses two permanent magnets rotating around the workspace.

Unfortunately, calcified fat deposits can build up inside arteries during the life of a person, a condition called atherosclerosis [?]. These deposits, called *plaques*, may be

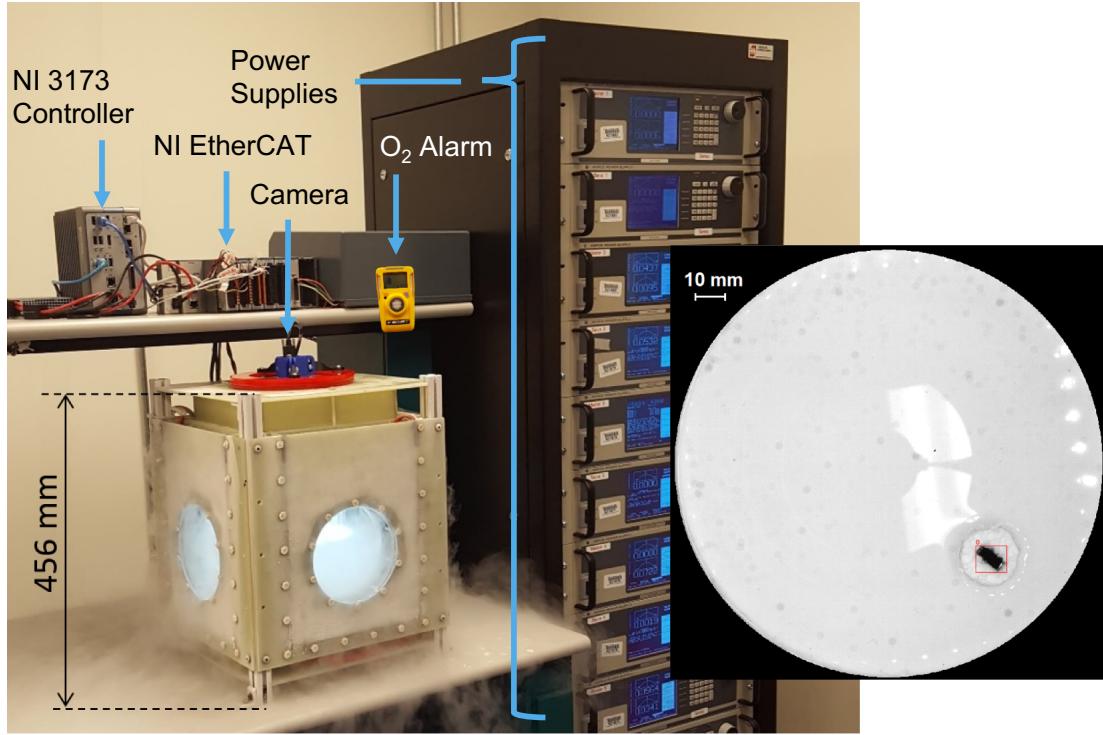


Figure 2.1: Picture of the magnetic manipulator while functioning. Inset shows view from the camera used to obtain position feedback of the robot.

detached by the friction between the catheter and the walls of the blood vessel and cause a stroke [?, ?].

Tetherless magnetic robots could further decrease the invasiveness of procedures [?]. The idea is to use external magnets to manipulate a small magnetic tool or capsule placed inside a patient. The absence of tethers and the ability to navigate without touching the blood vessel walls [?] reduces the risk of plaque detachment. A fast and accurate tracking method can enable precise control of the robot position during navigation. However, permanent magnets are limited by their maximum rotational speed and acceleration and are therefore often unsuitable for fast dynamic control. In contrast, electromagnets (EM) can generate magnetic fields with fast dynamics without moving parts. Their magnetic field change rate is limited by either the power of the generator or the maximum voltage the magnet can sustain.

MRI scanners have EM and can be used as magnetic manipulators [?, ?, ?]. When a ferromagnetic piece (the robot) is placed inside this device, it becomes magnetized by the main magnetic field. Because the main field is approximately uniform (less than 1 ppm of inhomogeneity [?]), it does not produce significant forces on the robot. Instead, the MRI gradient coils can be used to produce a controlled force.

The major advantage of using of MRI scanners for magnetic manipulation is that they are already widely available in hospitals and the robot's position can be tracked in real-time using the MR signal [?, ?]. However, the large static magnetic field in an MRI permanently orients the magnetic field in the same direction, making torque control of robots impossible. The gradient coils can be used to generate forces on the robot.

Power consumption and heat dissipation are the primary limitations on EMs. The current circulating inside a conductor produces Joule losses. Electromagnetic field strength is limited by its maximum power dissipation density. More compact magnet designs can be achieved by reducing the amount of losses produced per unit of flux density. If power consumption is not a concern, the maximum power dissipation density can also be increased.

Adding a ferromagnetic core to an EM is an effective way to increase the magnetic field and the forces produced without increasing the losses. Cores enable increasing the inductance value which in turn increases the total flux produced per ampere. However, the use of a ferromagnetic core requires a closed bore geometry for the magnet. This is often an issue for medical applications. MRIs are designed with open bores on two opposite sides to accommodate the patient. Additional openings could allow medical staff to access the patient during the procedure and decrease feelings of claustrophobia.

Using liquid nitrogen (LN2) is another way to decrease the amount of losses produced per unit of flux density. This method decreases the value of the electrical resistivity of copper and therefore allows more current to circulate inside the magnet for a given

amount of losses. Cooling the EM to cryogenic temperatures offers another advantage: because the temperature of the coolant is low, the maximum safe temperature increase of the coil is larger. LN2 cooling therefore increases the maximum power dissipation density. LN2 is cheap (approximately 0.13 USD/Liter) and available in industrial quantities. It is non-toxic as gaseous nitrogen composes 78% of the volume of our atmosphere. However, if large quantities of liquid nitrogen are evaporated, the level of oxygen in the room might decrease. An adapted ventilation system and a low oxygen alarm must be used to prevent anoxia.

This paper presents the design and test of a magnetic manipulator cooled with LN2. A demonstration system is described in §2.2. The motivations and technical difficulties associated with this type of cooling are discussed in §2.3. A method to perform inverse magnetic calculations is then explained in §2.4. A robot trajectory controller is described in §2.5. Next the system singularities are analyzed for 3-DOF control of a permanent magnet in §2.6. Experimental results are presented and analyzed in §2.7. The paper concludes with lessons learned in this study.

2.2 System Presentation

The magnetic manipulator in Fig. 2.1 is designed to fit a human heart phantom. Detailed build instructions, a bill of materials, and CAD models are available¹. The working volume is a sphere with radius 0.075 m. The system is composed of six copper coils arranged in a cubical shape (see Fig. 2.2). Each coil is placed and held in an independent cryostat. The cryostats contain the LN2 and were built using G10. G10 is a fiberglass-epoxy composite able to withstand cryogenic temperatures. Ordinary plastics become brittle at cryogenics temperatures, but G10 remains resilient. G10 is also electrically nonconductive, an important feature to avoid induced currents. Induced

¹github.com/RoboticSwarmControl/magnetic-manipulator-l2n-source

currents generate a magnetic field that opposes variations in the applied field, making the system less responsive. Induced currents also generate heat.

Uninsulated G10 walls become cold and water present in air condenses and freezes on them. This ice could interfere with objects in the workspace. To prevent icing, the cryostat walls facing the workspace are insulated by a 10 mm thick layer of Styrofoam insulation. Six acrylic plates (one for each internal face) containing a resistive heater and a thermocouple temperature sensor cover the inner face of the insulation. The temperature of the internal walls surrounding the workspace are regulated using a real-time controller. The evaporation rate of LN2 is the product $\mathcal{L} \cdot P_o$, where \mathcal{L} is the latent heat of vaporization, equal to 200 kJ/kg for LN2 and P_o is the power dissipated. The maximum power consumption of each coil of the system is 2 kW. At maximum power, 0.74 liters of LN2 evaporate per minute.

12 Kepco BOP 20-50 generators power the system. Each power supply can generate 20 A under 50 V. Each coil is powered by two of these supplies connected in series. Each coil can, therefore, receive a maximum of 20 A under 100 V. Each set of power supplies is controlled via an analog input. While the current inside an EM is directly proportional to the magnetic flux density produced, the voltage applied on an EM is proportional to the time derivative of the flux density. It is therefore easier to control the produced magnetic field by controlling the current rather than the voltage on the EM. The BOP power supplies can do this when controlled in current mode. In this mode, the power supplies output a current proportional to an analog input.

An industrial controller IC-3173 from National Instrument is used for real-time computation, as shown in Fig. 2.3(a). A set of two Basler acA2040 cameras are attached at orthogonal faces of the magnetic manipulator. These cameras are used to obtain robot position and orientation in real-time (100 Hz). Two high precision NI 9263 analog modules are used to generate the analog signal controlling the power supplies.

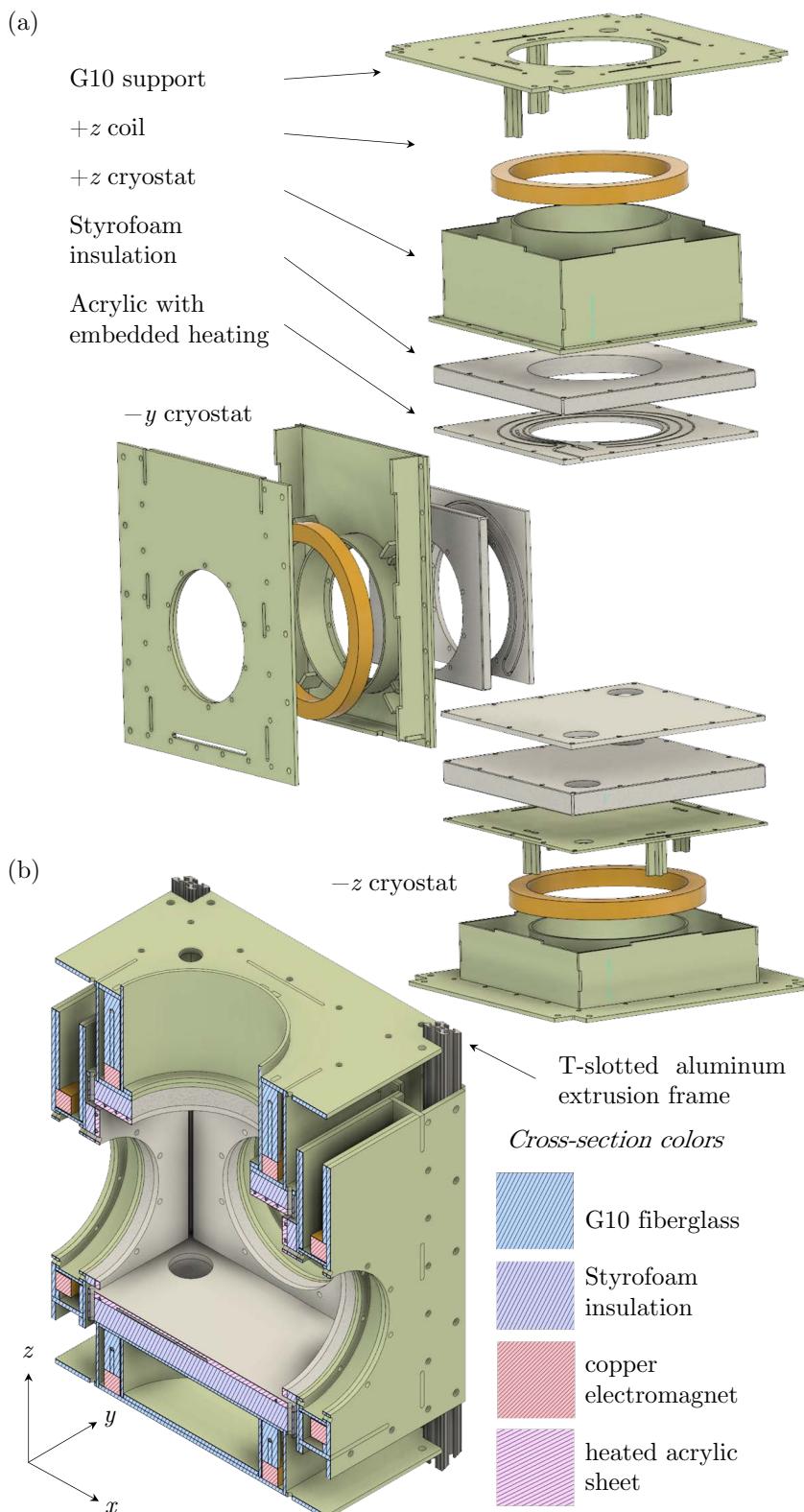


Figure 2.2: CAD model of the magnetic manipulator: (a) exploded view of three cryostats, (b) cross-section view.

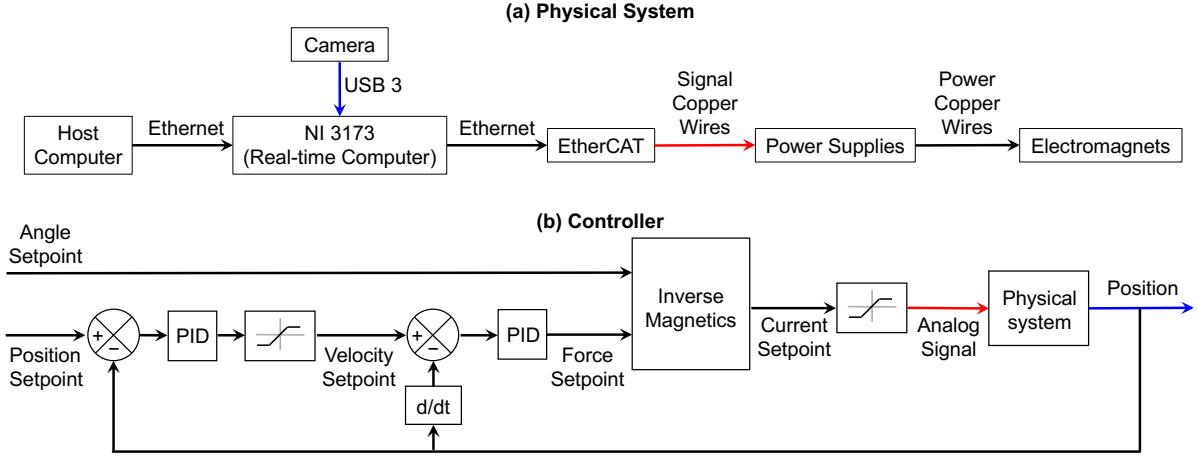


Figure 2.3: Block diagram of (a) the physical hardware and (b) the controller used for position control.

The laboratory is equipped with a low oxygen alarm (Honeywell BW Clip, approximately 140 USD).

2.3 Cooling with Liquid Nitrogen

2.3.1 Motivations

The voltage U_l at the terminals of a resistive coil can be calculated using the following equation:

$$U_l = L \cdot \frac{dI(t)}{dt} + R(t) \cdot I(t), \quad (2.1)$$

where I is the current in the coil, L is the coil inductance, and R is the coil resistance. Two components are present in this equation:

- The term $L \cdot dI(t)/dt$ is related to the magnetic energy change rate. This term does not cause losses if a four quadrant power supply is used. This type of power supply has the ability to transfer stored magnetic energy back to the electrical network. It is desirable to maximize $L \cdot dI(t)/dt$ because a high magnetic energy change rate generates a large force change rate, a desirable feature for robot control.

- The term $R(t) \cdot I(t)$ is associated with Joule losses. It is an undesirable term for two reasons. First, it causes the coil to heat and decreases the energy efficiency of the system. Secondly, power supplies are limited by their maximum voltage. The voltage across the coil is shared between the two terms of equation 2.1. If the term $R(t) \cdot I(t)$ is increased, less voltage is available for the term $L \cdot dI(t)/dt$. LN2 cooling allows reducing the value of $R(t)$ by 87%;

The magnetic flux Φ produced by a solenoid is proportional to its inductance as shown in (2.2). In [?], the authors use a ferromagnetic core to increase the value of L and therefore increase the amount of force generated. LN2 cooling is different and increases the generated field by increasing the value of $I(t)$.

$$\Phi(t) = L \cdot I(t) \quad (2.2)$$

It is technically difficult to scale up magnetic manipulators [?]. Air-cooled human-size manipulators would use comparatively large electromagnets to produce the magnetic field. This technical challenge could potentially be solved using LN2 cooling. This type of cooling enables a significant decrease in the system price, size, mass and/or improve its energy efficiency.

This tendency is illustrated in Fig. 2.5 and Table 2.1, which compares the flux density magnitude produced by different EM designs along their revolution axis. The comparison was computed using the magnetic modeling software Finite Elements Method Magnetics (FEMM [?]). EM1 and EM3 correspond to the EM present in our experimental setup when respectively cooled with LN2 (EM1) and forced air (EM3). The gain of flux density and magnetic gradient when cooling with LN2 is +435% and is relatively equal to the gain in current (see Table 2.2). EM2 and EM4 correspond to the same EM as 1 and 3 except that ferromagnetic cores were added to each. The gain in flux density is small, approximately 10%. EM6 is a forced-air-cooled EM with a ferromagnetic core designed to generate the same magnetic field as EM1. EM6 must be 2.5× longer than

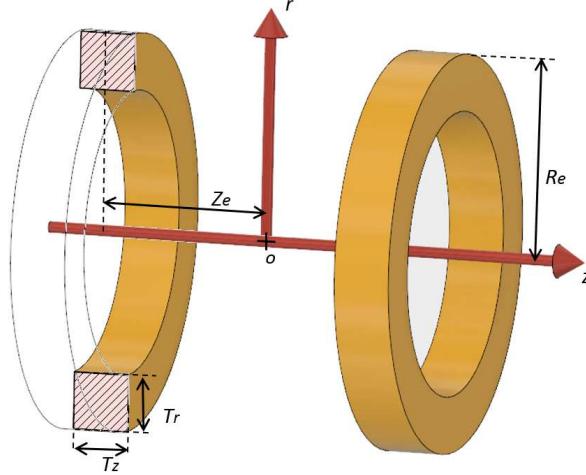


Figure 2.4: 3D representation of a dual-coil collinear EM assembly.

EM1. EM5 is the same as EM6 except that the ferromagnetic core was removed. The relative gain in flux density obtained by adding a ferromagnetic core is larger for EM5 and EM6, suggesting that the gain is related to the aspect ratio of length/diameter of the coil.

Table 2.1: Description of the electromagnet designs compared in Fig. 2.5.

Electromagnet ID	1	2	3	4	5	6
External radius R_e	110 mm	110 mm	110 mm	110 mm	110 mm	110 mm
Coil width T_r	20 mm	20 mm	20 mm	20 mm	20 mm	20 mm
Coil length T_z	20 mm	20 mm	20 mm	20 mm	50 mm	50 mm
Number of turns	795	795	795	795	1987	1987
Copper wire	AWG 22	AWG 22	AWG 22	AWG 22	AWG 22	AWG 22
Core	Air	Hyperco-50	Air	Hyperco-50	Air	Hyperco-50
Cooling method	LN2	LN2	Forced air	Forced air	Forced air	Forced air
Max. cont. current	7.5 A	7.5 A	1.4 A	1.4 A	1.4 A	1.4 A
Schematic						

2.3.2 Scaling Law

This section uses analytical equations to derive the gradient produced by two concentric electromagnets on their revolution axis (see Fig. 2.4). The electromagnets have an external radius R_e . They have a rectangular $T_z \times T_r$ cross-section. The filling factor

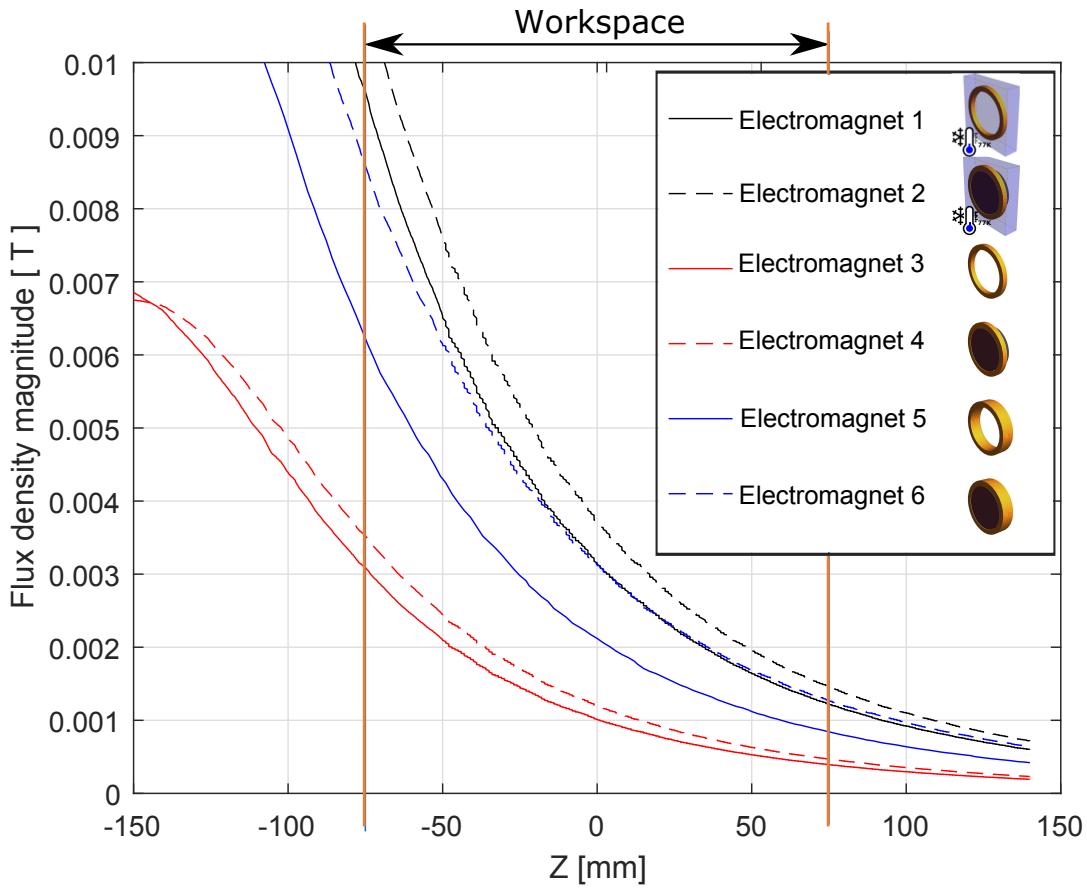


Figure 2.5: Maximum sustainable magnetic flux density by different EM designs. Magnet specifications are given in Table 2.1.

Table 2.2: Comparison between EM properties at room temperature (300K) and at LN2 temperature (77K).

	300K	77K	Difference
Copper electrical resistivity	1.68E-8 Ω·m	2.15E-9 Ω·m	-87%
Coil electrical resistance	27.3 Ω	3.5 Ω	-87%
Max continuous current	1.4 A	7.5 A	+435%
Max cont. current density	4.3 A/mm ²	23 A/mm ²	+435%

F_{Fill} is assumed to be equal to 0.7. LN2 cooled magnets have an additional insulation that has a thickness T_{insul} which reduces the internal bore diameter D_b .

The flux density B_u produced by a single current loop can be calculated on its revolution axis using the Biot-Savart law:

$$B_u(R_u, Z_u) = \frac{\mu_0 \cdot I_u}{2} \cdot \left(\frac{R_u^2}{R_u^2 + Z_u^2} \right)^{\frac{3}{2}}, \quad (2.3)$$

where R_u is the radius of the loop and Z_u is the distance along the axis between the calculation point and the center of the loop. The axial flux density created by a single finite EM is obtained by integrating this equation over the EM cross section:

$$\begin{aligned} B_e(R_e, Z_e) &= \frac{J\mu_0 F_{\text{Fill}}}{2} \int_{Z_e - T_z/2}^{Z_e + T_z/2} \int_{R_e - T_r}^{R_e} \left(\frac{R_u^2}{R_u^2 + Z_u^2} \right)^{\frac{3}{2}} dR_u dZ_u \\ &= \frac{J\mu_0 F_{\text{Fill}}}{8} \left((T_z + 2Z_e) \left(\sqrt{4R_e^2 + (T_z + 2Z_e)^2} - \sqrt{4(R_e - T_r)^2 + (T_z + 2Z_e)^2} \right) \right. \\ &\quad \left. + (T_z - 2Z_e) \left(\sqrt{4R_e^2 + (T_z - 2Z_e)^2} - \sqrt{4(R_e - T_r)^2 + (T_z - 2Z_e)^2} \right) \right) \end{aligned} \quad (2.4)$$

where R_e is the external radius of the coil, Z_e is the distance along the axis between the calculation point and the center of the coil, and J is the current density inside the copper wire. The quantity $J \cdot F_{\text{Fill}}$ is the current density averaged over the winding $T_z \times T_r$ cross section. The magnetic gradient G_e is calculated by the derivative of this equation with respect to Z_e :

$$G_e = \frac{dB_e}{dZ_e} \quad (2.5)$$

The total power P_e dissipated inside the EM can be computed with:

$$P_e = \iiint_V F_{\text{Fill}} \cdot \rho \cdot J^2 dv, \quad (2.6)$$

over the volume V :

$$V = \pi \left(2R_e (R_e - T_{\text{insul}} - D_b) - (R_e - T_{\text{insul}} - D_b)^2 \right). \quad (2.7)$$

These equations were implemented in MATLAB with $T_z = T_r$. The function `fminsearch` was used to inverse this equation and find the value of T_z that produces the desired gradient G_e for a given R_e and Z_e .

Fig. 2.6 presents simulation results obtained using equations 2.3 to 2.6. Plot (a) graphs the EM winding volume as a function of its external diameter. The blue and red curves present the T_z that produces a gradient of 20 mT/m and 45 mT/m respectively. The value of T_z and therefore the volume of the winding changes along these curves to produce the desired gradient strength.

Black curves have been added to locate the functioning points for a human-size system. They represent systems that have a 0.7 m bore diameter D_b (similar to the bore diameters of MRI scanners). Dashed lines are plotted for an air cooled magnet while solid lines are for EM cooled with LN2.

To summarize, colored curved show systems that are able to produce a given gradient (20 mT/m for red curves and 45 mT/m for blue curves). The value of D_b is changing along these curves. Black curves represent systems that have D_b values of 0.7 m. The produced gradient strength is changing along these curves. The intersections of the black and colored curves represent functioning points of systems able to produce a given gradient strength with a D_b value of 0.7 m. Plot (b) in Fig. 2.6 is similar to plot (a) except that the results are presented in terms of power consumption. These data were obtained from the results presented in plot (a) and calculated using eq. 2.6.

Results show that the windings of LN2 cooled systems are always smaller than air cooled windings. They also always use less power. A human-size system producing 45 mT/m would require EMs with a volume of 0.0117 m³ and 0.0585 m³ for LN2 and air-cooled EM respectively. Their power consumptions are respectively 17.2 kW and 8.56 kW. The use of LN2 therefore allows a reduction of 80% of the winding volume and a decrease of 50% of the power consumption.

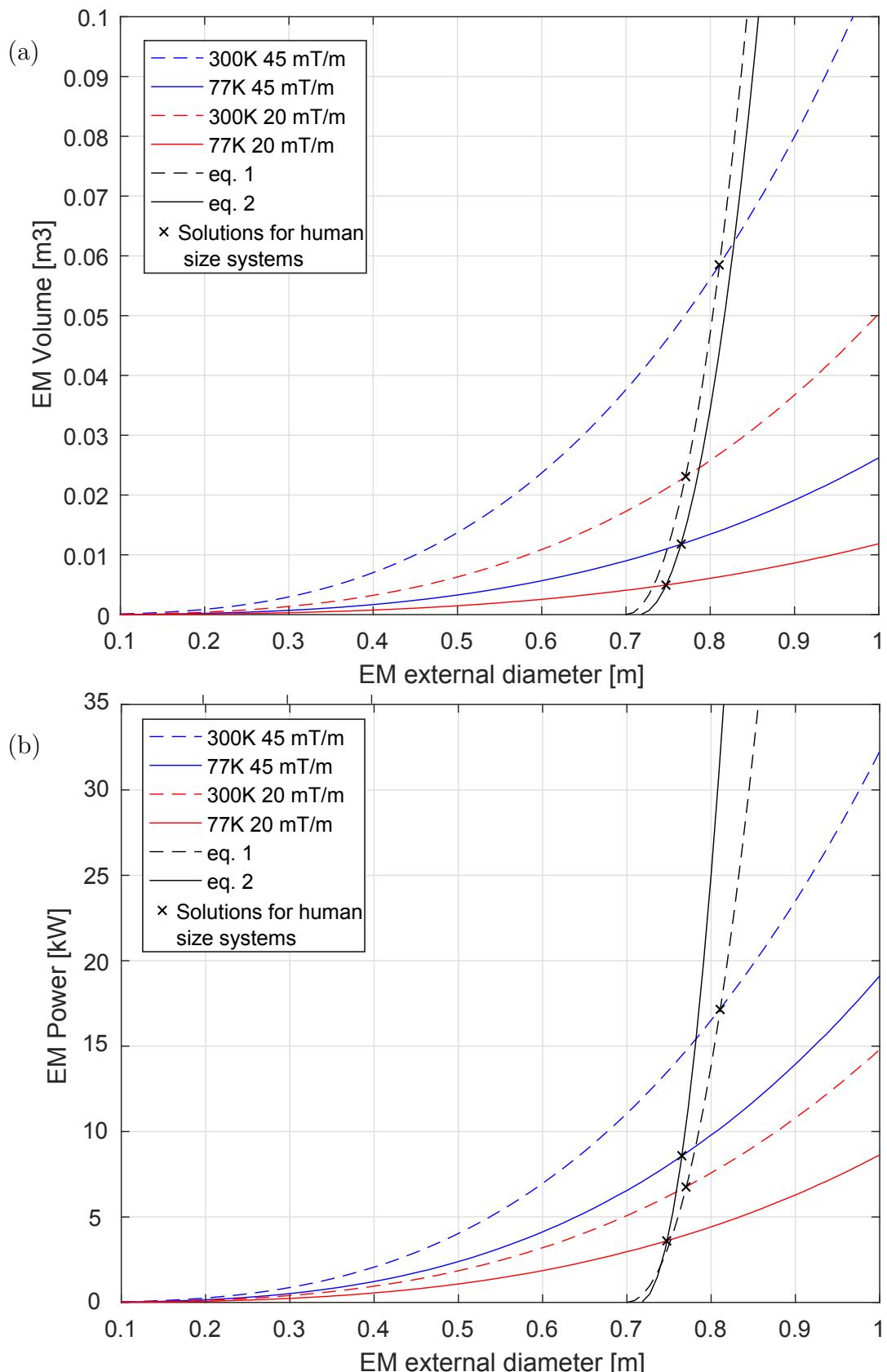


Figure 2.6: Plot of the winding volume (a) and power consumption (b) of air-cooled (300K) and LN₂-cooled (77K) EMs. Red curves correspond to magnetic assemblies able to produce a gradient strength of 20 mT/m at the system

2.4 Inverse Magnetics

This section analyzes 3D manipulation of a single robot having a magnetization \mathbf{m} . The magnetic manipulator is composed of six EM controlled by independent current sources. The total magnetic field is the sum of the field produced by each coil. This section calculates the coil current values to produce the desired force and torque on the robot or the desired force and magnetic field orientation.

2.4.1 Forward problem

To simplify analysis, we first solve the *forward problem* which computes the force and flux density or force and torque using the coil currents.

The system has six EMs, numerated from 1 to 6. The magnetic flux density $B_{k(\mathbf{P})}$ produced by EM number k at location \mathbf{P} can be calculated using eq. 2.8 where I_k is the current in the magnet and $\mathbf{B}_{k(\mathbf{P})}$ is a function that depends on the geometry of the system and the position of the magnet. The function $\mathbf{B}_{k(\mathbf{P})}$ is computed assuming that the coils are equivalent to current loops. The flux density produced by a current loop is calculated using the semi-analytical equations presented in [?].

$$\mathbf{B}_{k(\mathbf{P})} = \begin{bmatrix} B_{kx} \\ B_{ky} \\ B_{kz} \end{bmatrix} = \tilde{\mathbf{B}}_{k(\mathbf{P})} \cdot I_k = \begin{bmatrix} \tilde{B}_{kx(\mathbf{P})} \\ \tilde{B}_{ky(\mathbf{P})} \\ \tilde{B}_{kz(\mathbf{P})} \end{bmatrix} \cdot I_k \quad (2.8)$$

The flux density produced by the complete system can be computed by summing the field produced by each magnet (see eq. 2.9) where \mathbf{I} is a vector containing the current of each coil.

$$\mathbf{B}(\mathbf{P}) = (\tilde{\mathbf{B}}_{1(\mathbf{P})} + \tilde{\mathbf{B}}_{2(\mathbf{P})} + \tilde{\mathbf{B}}_{3(\mathbf{P})} + \tilde{\mathbf{B}}_{4(\mathbf{P})} + \tilde{\mathbf{B}}_{5(\mathbf{P})} + \tilde{\mathbf{B}}_{6(\mathbf{P})}) \mathbf{I} \quad (2.9)$$

It is now necessary to define three new vectors:

$$\tilde{\mathbf{B}}_x(\mathbf{P}) = \left[\tilde{B}_{1x}(\mathbf{P}) \ \tilde{B}_{2x}(\mathbf{P}) \ \tilde{B}_{3x}(\mathbf{P}) \ \tilde{B}_{4x}(\mathbf{P}) \ \tilde{B}_{5x}(\mathbf{P}) \ \tilde{B}_{6x}(\mathbf{P}) \right] \quad (2.10)$$

$$\tilde{\mathbf{B}}_y(\mathbf{P}) = \left[\tilde{B}_{1y}(\mathbf{P}) \ \tilde{B}_{2y}(\mathbf{P}) \ \tilde{B}_{3y}(\mathbf{P}) \ \tilde{B}_{4y}(\mathbf{P}) \ \tilde{B}_{5y}(\mathbf{P}) \ \tilde{B}_{6y}(\mathbf{P}) \right] \quad (2.11)$$

$$\tilde{\mathbf{B}}_z(\mathbf{P}) = \left[\tilde{B}_{1z}(\mathbf{P}) \ \tilde{B}_{2z}(\mathbf{P}) \ \tilde{B}_{3z}(\mathbf{P}) \ \tilde{B}_{4z}(\mathbf{P}) \ \tilde{B}_{5z}(\mathbf{P}) \ \tilde{B}_{6z}(\mathbf{P}) \right] \quad (2.12)$$

Equation 2.9 can be re-written as follows:

$$\mathbf{B}_{(\mathbf{P})} = \begin{bmatrix} \tilde{\mathbf{B}}_{x(\mathbf{P})} \\ \tilde{\mathbf{B}}_{y(\mathbf{P})} \\ \tilde{\mathbf{B}}_{z(\mathbf{P})} \end{bmatrix} \cdot \mathbf{I} \quad (2.13)$$

The force \mathbf{F} is calculated with:

$$\mathbf{F} = \begin{bmatrix} F_{x(\mathbf{P})} \\ F_{y(\mathbf{P})} \\ F_{z(\mathbf{P})} \end{bmatrix} = \nabla \cdot (\mathbf{m} \cdot \mathbf{B}_{(\mathbf{P})}), \quad (2.14)$$

which can be re-written as:

$$\mathbf{F} = \begin{bmatrix} m_x \cdot \frac{\partial \tilde{\mathbf{B}}_{x(\mathbf{P})}}{\partial x} + m_y \cdot \frac{\partial \tilde{\mathbf{B}}_{y(\mathbf{P})}}{\partial x} + m_z \cdot \frac{\partial \tilde{\mathbf{B}}_{z(\mathbf{P})}}{\partial x} \\ m_x \cdot \frac{\partial \tilde{\mathbf{B}}_{x(\mathbf{P})}}{\partial y} + m_y \cdot \frac{\partial \tilde{\mathbf{B}}_{y(\mathbf{P})}}{\partial y} + m_z \cdot \frac{\partial \tilde{\mathbf{B}}_{z(\mathbf{P})}}{\partial y} \\ m_x \cdot \frac{\partial \tilde{\mathbf{B}}_{x(\mathbf{P})}}{\partial z} + m_y \cdot \frac{\partial \tilde{\mathbf{B}}_{y(\mathbf{P})}}{\partial z} + m_z \cdot \frac{\partial \tilde{\mathbf{B}}_{z(\mathbf{P})}}{\partial z} \end{bmatrix} \cdot \mathbf{I} \quad (2.15)$$

The torque \mathbf{T} is calculated with:

$$\mathbf{T} = \begin{bmatrix} T_{x(\mathbf{P})} \\ T_{y(\mathbf{P})} \\ T_{z(\mathbf{P})} \end{bmatrix} = \mathbf{m} \times \mathbf{B} \quad (2.16)$$

which can be re-written as:

$$\mathbf{T} = \begin{bmatrix} m_y \cdot \tilde{\mathbf{B}}_{z(\mathbf{P})} - m_z \cdot \tilde{\mathbf{B}}_{y(\mathbf{P})} \\ m_z \cdot \tilde{\mathbf{B}}_{x(\mathbf{P})} - m_x \cdot \tilde{\mathbf{B}}_{z(\mathbf{P})} \\ m_x \cdot \tilde{\mathbf{B}}_{y(\mathbf{P})} - m_y \cdot \tilde{\mathbf{B}}_{x(\mathbf{P})} \end{bmatrix} \cdot \mathbf{I} \quad (2.17)$$

2.4.2 Inverse problem

Two inverse methods are studied. The first one aims at controlling the flux density and the force applied on the robot using the actuation matrix \mathbf{A}_0 :

$$\mathbf{A}_0 = \begin{bmatrix} \tilde{\mathbf{B}}_x(\mathbf{P}) \\ \tilde{\mathbf{B}}_y(\mathbf{P}) \\ \tilde{\mathbf{B}}_z(\mathbf{P}) \\ m_x \cdot \frac{\partial \tilde{\mathbf{B}}_x(\mathbf{P})}{\partial x} + m_y \cdot \frac{\partial \tilde{\mathbf{B}}_y(\mathbf{P})}{\partial x} + m_z \cdot \frac{\partial \tilde{\mathbf{B}}_z(\mathbf{P})}{\partial x} \\ m_x \cdot \frac{\partial \tilde{\mathbf{B}}_x(\mathbf{P})}{\partial y} + m_y \cdot \frac{\partial \tilde{\mathbf{B}}_y(\mathbf{P})}{\partial y} + m_z \cdot \frac{\partial \tilde{\mathbf{B}}_z(\mathbf{P})}{\partial y} \\ m_x \cdot \frac{\partial \tilde{\mathbf{B}}_x(\mathbf{P})}{\partial z} + m_y \cdot \frac{\partial \tilde{\mathbf{B}}_y(\mathbf{P})}{\partial z} + m_z \cdot \frac{\partial \tilde{\mathbf{B}}_z(\mathbf{P})}{\partial z} \end{bmatrix} \quad (2.18)$$

The force and flux density are equal to:

$$\begin{bmatrix} \mathbf{B} \\ \mathbf{F} \end{bmatrix} = \mathbf{A}_0 \cdot \mathbf{I} \quad (2.19)$$

\mathbf{A}_0 is a square matrix and can be inverted provided that it is not singular. The current \mathbf{I} is calculated with:

$$\mathbf{I} = \mathbf{A}_0^{-1} \cdot \begin{bmatrix} \mathbf{B} \\ \mathbf{F} \end{bmatrix} \quad (2.20)$$

The second method controls the force and torque applied on the robot using the actuation matrix \mathbf{A}_1

$$\mathbf{A}_1 = \begin{bmatrix} m_y \cdot \tilde{\mathbf{B}}_z(\mathbf{P}) - m_z \cdot \tilde{\mathbf{B}}_y(\mathbf{P}) \\ m_z \cdot \tilde{\mathbf{B}}_x(\mathbf{P}) - m_x \cdot \tilde{\mathbf{B}}_z(\mathbf{P}) \\ m_x \cdot \tilde{\mathbf{B}}_y(\mathbf{P}) - m_y \cdot \tilde{\mathbf{B}}_x(\mathbf{P}) \\ m_x \cdot \frac{\partial \tilde{\mathbf{B}}_x(\mathbf{P})}{\partial x} + m_y \cdot \frac{\partial \tilde{\mathbf{B}}_y(\mathbf{P})}{\partial x} + m_z \cdot \frac{\partial \tilde{\mathbf{B}}_z(\mathbf{P})}{\partial x} \\ m_x \cdot \frac{\partial \tilde{\mathbf{B}}_x(\mathbf{P})}{\partial y} + m_y \cdot \frac{\partial \tilde{\mathbf{B}}_y(\mathbf{P})}{\partial y} + m_z \cdot \frac{\partial \tilde{\mathbf{B}}_z(\mathbf{P})}{\partial y} \\ m_x \cdot \frac{\partial \tilde{\mathbf{B}}_x(\mathbf{P})}{\partial z} + m_y \cdot \frac{\partial \tilde{\mathbf{B}}_y(\mathbf{P})}{\partial z} + m_z \cdot \frac{\partial \tilde{\mathbf{B}}_z(\mathbf{P})}{\partial z} \end{bmatrix} \quad (2.21)$$

The torque and force are equal to:

$$\begin{bmatrix} \mathbf{T} \\ \mathbf{F} \end{bmatrix} = \mathbf{A}_1 \cdot \mathbf{I} \quad (2.22)$$

\mathbf{A}_1 is a square matrix and can be inverted provided that it is not singular. The current \mathbf{I} is calculated with:

$$\mathbf{I} = \mathbf{A}_1^{-1} \cdot \begin{bmatrix} \mathbf{T} \\ \mathbf{F} \end{bmatrix} \quad (2.23)$$

2.5 Trajectory control

Equation 2.19 shows that \mathbf{F} is decoupled from \mathbf{B} and \mathbf{T} . The control of \mathbf{B} enables the control of the orientation of the robot. The robot can be assumed to be oriented along the magnetic field direction, as in [?]. The magnitude of the flux density $|\mathbf{B}|$ is set to a constant value and the individual components are calculated to obtain the desired field orientation. Another alternative is to use eq. 2.23 to control the torque directly.

The trajectory is controlled using the controller presented in Fig. 2.3(b). It uses a nested control structure. The inner PID control loop regulates velocity, which is limited by a saturation function to avoid excessive speeds and instabilities. The outer loop generates a velocity setpoint.

The trajectory is defined by the user as a set of viapoints and the corresponding desired velocities V_t . The outer loop first searches for the point of the trajectory that is the closest to the robot position and determines V_t . An additional velocity component V_s is added to this value to steer the robot toward the trajectory centerline.

When the calculated current is above the maximum value (I_{\max} is the maximum current the power supplies can generate) the vector \mathbf{I} is scaled down so that the largest element of \mathbf{I} is equal to I_{\max} . This reduces the force and flux density values applied on the robot, but has the advantage of keeping the correct field and force orientation.

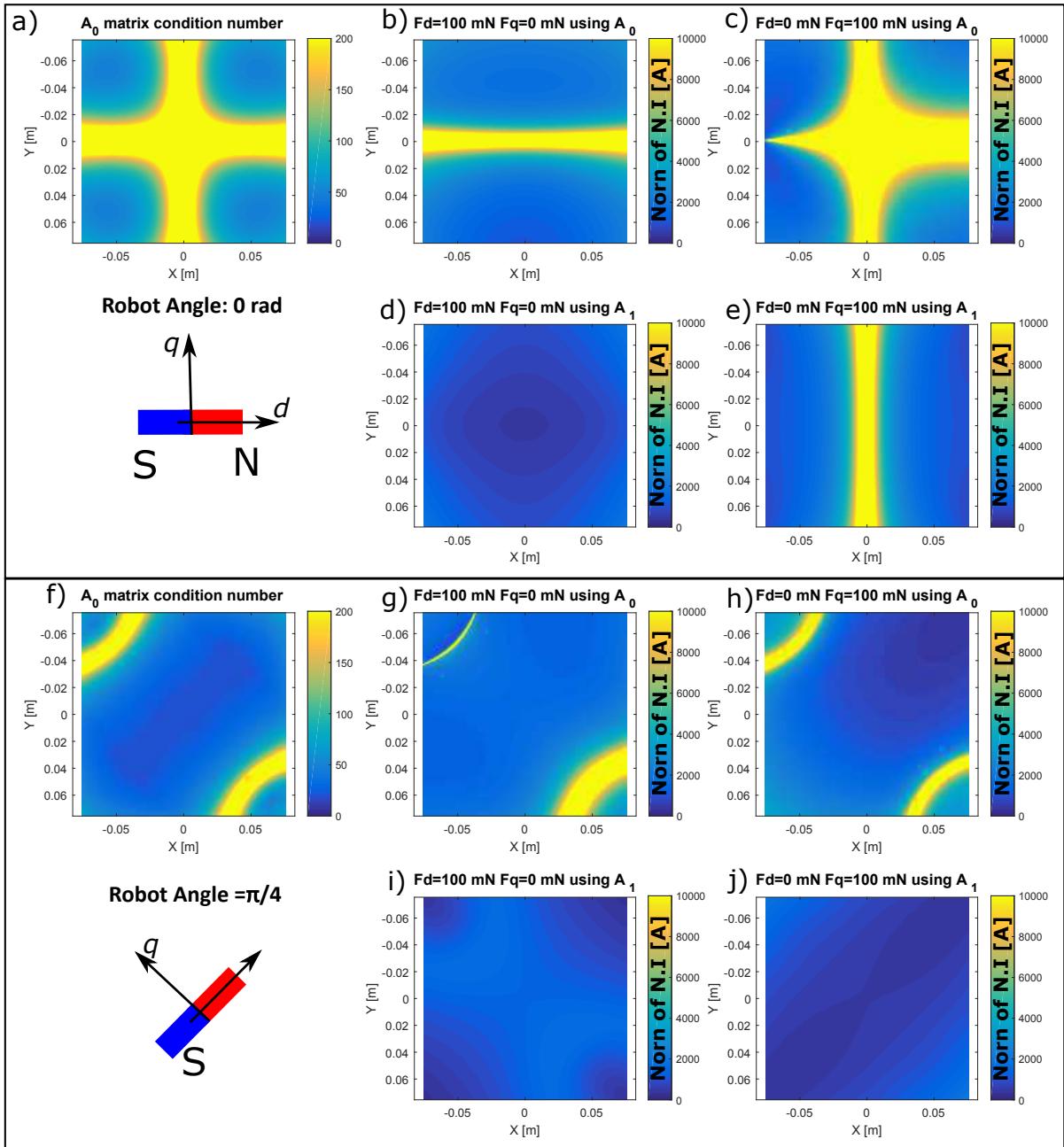


Figure 2.7: Map of the conditioning of the actuation matrix \mathbf{A}_0 for (a) $\theta = 0$ and (f) $\theta = \pi/4$. Map of the norm of the total current needed to produce a force of 100 mN along the d axis (plots b, d, g and i) and along the q axis (plots c, e, h and j). Plots b, c, d and e are plotted for $\theta = 0$ and plots g, h, i and j are plotted for $\theta = \pi/4$. Plots b, c, g and h were calculated using \mathbf{A}_0 and plots d, e, i and j were calculated using \mathbf{A}_1 .

2.6 Singularities analysis for a 3-DOF robot

As explained in §2.5, the orientation of the robot can be controlled using two different methods. The first one uses actuation matrix \mathbf{A}_0 and allows applying a desired flux density vector \mathbf{B} and a force \mathbf{F} to the robot. The second uses actuation matrix \mathbf{A}_1 and allows applying a desired torque \mathbf{T} and a force \mathbf{F} to the robot.

When expressed in the manipulator coordinate system (x, y, z) , \mathbf{A}_1 is square. However, the robot is symmetric around its revolution axis d and therefore no torque can be applied on the d axis. Defining a new coordinate system (d, q, w) allows removing one dimension from \mathbf{A}_1 (i.e. \mathbf{A}_1 has a size 5×6 when expressed in the (d, q, w) coordinate system). The system is therefore underdetermined and the least square solution is calculated using eq. 2.24.

$$\mathbf{I} = \mathbf{A}_1^T (\mathbf{A}_1 \cdot \mathbf{A}_1^T)^{-1} \cdot \begin{bmatrix} \mathbf{T} \\ \mathbf{F} \end{bmatrix} \quad (2.24)$$

The actuation matrices \mathbf{A}_0 and \mathbf{A}_1 are functions of the magnetic manipulator geometry and the robot pose.

The effect of singularities was studied for a 2D-3DOF control. The magnet is horizontally placed on a flat surface corresponding to the x - y plane. The robot is able to move along x and y and rotate around the z -axis.

A 4×4 \mathbf{A}_0 and a 4×3 \mathbf{A}_1 actuation matrices can be calculated for each robot position and angle. The map of the conditioning of the \mathbf{A}_0 matrix is shown in Fig. 2.7(a) and (f). On this figure is also presented the norm of the total current $N \cdot \mathbf{I}$ required to produce a force of 100 mN along the d -axis (central-column) and q -axis (right-column). Plots for $\theta = 0$ and $\theta = \pi/4$ are presented. Results for other angles are shown in the attached video.

For $\theta = 0$, \mathbf{A}_0 has two singularity lines on the x and y axis. For $\theta = \pi/4$ singu-

larities take a rounded shape and are located on two opposite angles of the workspace. No solution is available when the matrix is singular. The singularities are situated on infinitely thin lines, but the system is ill-conditioned near the singularity lines. A large condition number produces a large magnitude for the \mathbf{I} vector. The capabilities of the power supplies limit the maximum current and when the condition number becomes too large, the current saturates. This saturation decreases control authority over the robot in these regions.

The norm of the total current $N \cdot \mathbf{I}$ needed to produce a force of 100 mN along the d and q axis was calculated as a function of the robot position and angle. Plots b, c, e and f of Fig. 2.7 present these results. When using the \mathbf{A}_0 matrix, there are areas where the current becomes larger than 10,000 A close to the singularities when the force is generated along the d axis as well as when the force is generated along the q axis. When using \mathbf{A}_1 , these large current densities are present only when a force is generated along the q axis. When the force is produced along the d axis, the current always stays within relatively small values. This observation is valid for any robot orientation.

2.7 Experimental Results

The control of the velocity and orientation of a NdFeB permanent magnet in a 2D plane described in §2.5 was implemented and tested using the \mathbf{A}_1 actuation matrix. As shown in §2.6 it is preferable to apply the force along the magnetization axis of the robot (d -axis). The program was therefore configured to orient the robot magnetization in the same direction as the applied force. This allows minimizing the value of the force applied along the q axis and therefore improve stability by avoiding current saturations. The permanent magnet was cylindrical, with a diameter of 2.5 mm and a length of 10 mm. It was encapsulated in a black shrink tube to facilitate computer vision tracking. The magnet was then attached horizontally on a Styrofoam disk having a diameter of 18

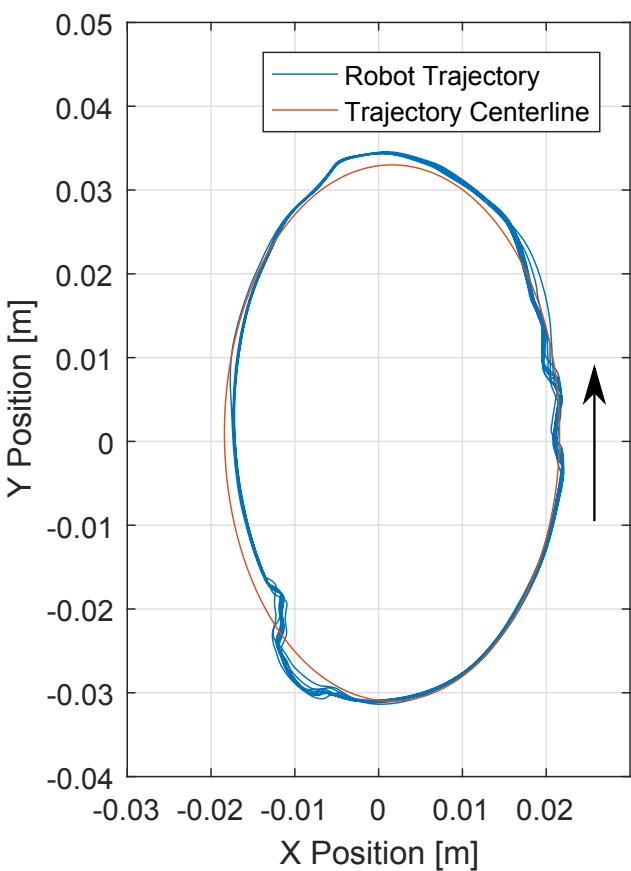


Figure 2.8: Plot of the trajectory of the robot obtained experimentally. For this dataset, the robot completed the path ten times. The robot and the workspace are shown at right.

mm and a thickness of 6 mm. Pictures of the robot are shown in Fig. 2.8.

A robot navigating inside fluid-filled cavities of the human body may be designed to have neutral buoyancy to reduce the amount of force needed. To simulate neutral buoyancy in this 2D control experiment, the magnet-Styrofoam assembly floated at the surface of a water-filled tank. The magnet was able to move freely in the x and y directions as well as rotate around the z axis.

The method from §2.5 was used with the \mathbf{A}_1 actuation matrix from (2.21). The trajectory was an ellipse having a major axis of 63 mm and a minor axis of 40 mm.

Tests were performed with and without LN2 cooling. The trajectory obtained experimentally without LN2 is presented in Fig. 2.8. For these plots, the robot followed

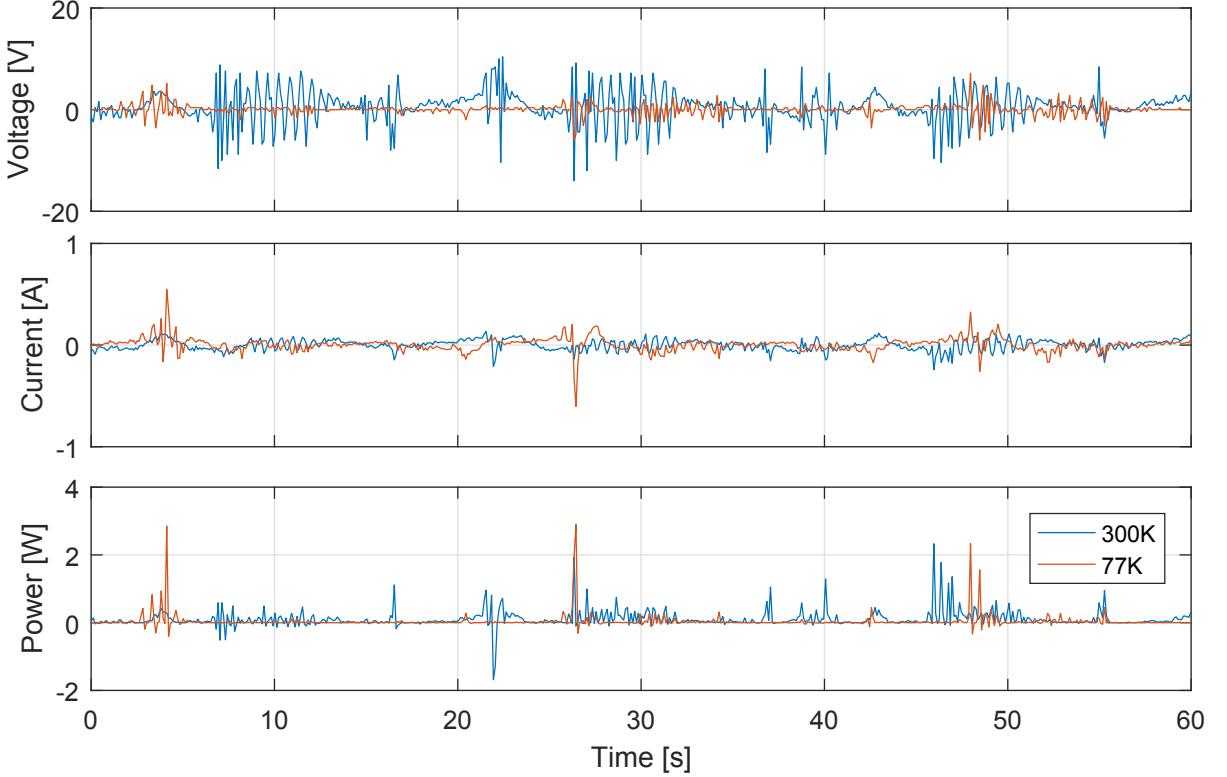


Figure 2.9: Voltage, current, and power used during robot navigation. The curves show 60 seconds of data, which correspond to the robot completing the trajectory in Fig. 2.8 three times.

the trajectory ten times. Trajectories obtained with LN2 cooling are similar. The robot was stable during the navigation, but small deviation from the centerline were observed at several locations.

The current and voltage applied on the $-x$ EM was recorded during one minute for both cooling methods. The instantaneous power was calculated from these data and the results are presented in Fig. 2.9. The average power used for the air cooled case is equal to 0.218 W whereas it is only equal to 0.037 W when magnets are cooled with LN2. This decrease of average power consumption is enabled by the decrease of copper electrical resistivity when cooled at 77K and produces a decrease of the applied voltage as shown in Fig. 2.9(a). The parameters of the controller were not changed between the tests, however, the peak power is increased when LN2 is added. This behavior could be explained by an increase of the current regulation dynamics performed by the power

supply when the electrical resistance of the EM is decreased by the cooling.

The power used was low because the floating robot required little force. Applications that navigate against flow or perform surgery require larger forces and correspondingly more power.

2.8 Conclusion and Future Works

This paper presented a magnetic manipulator using EM cooled with LN2. Liquid nitrogen cooling allows increasing the current circulating inside an EM up to 435%. This cooling enables reducing the size of the EM to produce a given magnetic field. The required electromagnets to achieve a given flux density are cheaper to build, and the complete system is more compact.

A desktop-size prototype was built and tested. The robot, a cylindrical permanent magnet, was manipulated on a 2D plane. Three DOF were controlled: the orientation and the x and y positions.

The system was not designed to produce a uniform magnetic field. Instead, inverse magnetic calculations account for field non-uniformities. The current inside each coil is computed to generate the desired force on the robot and produce the desired field orientation.

The system's matrix conditioning was analyzed. The matrix is sometimes ill-conditioned, depending on the position and orientation of the robot. Issues with ill conditioning can be avoided by controlling the torque directly and avoiding the production of forces along the axis perpendicular to the robot magnetization.

More control inputs can be added to improve the controllability of the robot. A possibility is to use additional EM, as in [?] where eight EMs were used to control a five DOF robot, but using additional EMs makes the system more complicated and expensive

to build.

Future study will focus on the addition of a high-frequency component to the magnetic field to increase the controllability of the robot. A permanent magnet could be encapsulated in a conductive shell such as copper. If the robot is electrically conductive, the AC field would induce currents in it, and generate an additional torque as in an induction electric motor. This AC magnetic field could also be used to control resonating magnetic actuators as in [?]. Finally, the controller could include a temperature management feature that calculates the heat dissipated in the windings and avoids overheating by preventively reducing power.

Chapter 3

Exploiting Non-Slip Wall Contacts to Position Two Particles Using The Same Control Input

This chapter is based on a journal submission with Shiva Shahrokhi, Jingang Shi, Benedict Isichei, and Aaron T. Becker. My contribution was focused on ??????????. This work was supported by the National Science Foundation under Grant No. [IIS-1553063] and [IIS-1619278].

Steered particles offer a method for targeted therapy, interventions, and drug delivery in regions inaccessible by large robots. Magnetic actuation has the benefits of requiring no tethers, being able to operate from a distance, and in some cases allows imaging for feedback (e.g. MRI). This paper investigates particle control with uniform magnetic gradients (the same force is applied everywhere in the workspace). Given three orthogonal magnetic fields, steering one particle in 3D is trivial. Adding additional particles to steer makes the system underactuated because there are more states than control inputs. However, the walls of *in vivo* and artificial environments often have surface roughness such that the particles do not move unless actuation pulls them away from the wall. In previous work, we showed that the individual 2D position of two particles is controllable in a square workspace with non-slip wall contact [?]. Because *in vivo* environments are usually not square, this paper extends the previous work to all convex workspaces and to 3D positioning. This paper also implements the algorithms using a hardware setup inspired by the gastrointestinal tract.

3.1 Introduction

Particle swarms propelled by a uniform field, where each particle receives the same control input, are common in applied mathematics, biology, and computer graphics [?, ?, ?].

The small size of these robots makes it difficult to perform onboard computation. Instead, these robots are often controlled by a broadcast signal. The tiny robots themselves are often just rigid bodies, and it may be more accurate to define the robot as the *system* that consists of particles, a uniform control field, and sensing. Such systems are severely underactuated, having 2 degrees of freedom in the shared planar control input, but $2n$ degrees of freedom for the n -particle swarm. Techniques are needed that can handle this underactuation.

Positioning is a foundational capability for a robotic system, e.g. placement of brachytherapy seeds. In previous work, we showed that the 2D position of each particle in such a swarm is controllable if the workspace contains a single obstacle the size of one particle [?]. However, requiring a single, small, rigid obstacle suspended in the middle of the workspace is often an unreasonable constraint, especially in 3D. This paper relaxes that constraint, and provides position control algorithms that only require non-slip wall contacts. We assume that particles in contact with the boundaries have zero velocity if the uniform control input pushes the particle into the wall.

The paper is arranged as follows. After a review of recent related work in Sec. 3.2, Sec. 3.3 introduces a model for boundary interaction and two shortest path results for representative workspaces. We provide an algorithm to arbitrarily position two particles in Sec. 3.4. Section 3.5 describes implementations of the algorithms in simulation and Sec. 3.6 describes hardware experiments, as shown in Fig. 3.1. We end with directions for future research in Sec. 3.7.

This paper is an elaboration of preliminary work in a conference paper [?] which

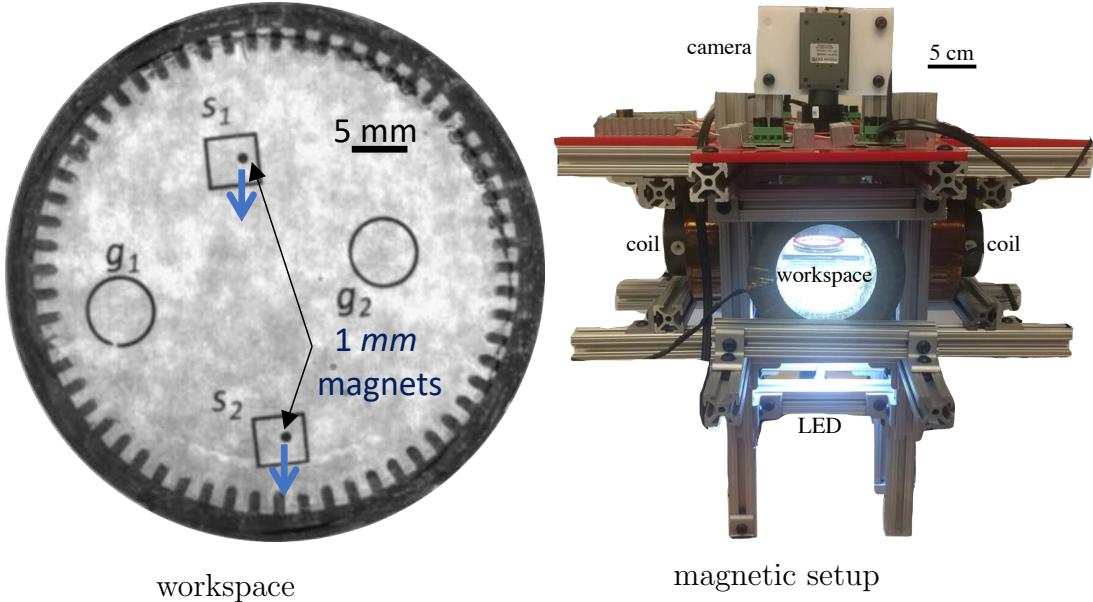


Figure 3.1: Workspace and magnetic setup for an experiment to position particles that receive the same control inputs, but cannot move while a control input pushes them into a boundary.

considered only square workspaces. This work extends the analysis to convex workspaces and 3D positioning. This paper also implements the algorithms using a hardware setup inspired by the anatomy of the gastrointestinal tract.

3.2 Related Work

Controlling the *shape*, or relative positions, of a swarm of robots is a key ability for a range of applications. Correspondingly, it has been studied from a control-theoretic perspective in both centralized and decentralized approaches. For examples of each, see the centralized virtual leaders in [?], and the gradient-based decentralized controllers using control-Lyapunov functions in [?]. However, these approaches assume a level of intelligence and autonomy in individual robots that exceeds the capabilities of many systems, including current micro- and nano-robots. Current micro- and nano-robots, such as those in [?, ?, ?] lack onboard computation.

This paper focuses on centralized techniques that apply the same control input to

both particles. Precision control requires breaking the symmetry caused by the uniform input. Symmetry can be broken using particles that respond differently to the uniform control signal, either through agent-agent reactions [?], or engineered inhomogeneity [?, ?, ?]. The magnetic gradients of MRI scanners are *uniform*, meaning the same force is applied everywhere in the workspace[?]. This work assumes a uniform control with homogenous particles, as in [?], and breaks the control symmetry using obstacles in the workspace.

Alternative techniques rely on non-uniform inputs, such as artificial force-fields. Applications have included techniques to design shear forces for sensorless manipulation of a single object by [?]. [?] demonstrated a collection of 2D force fields generated by six degree-of-freedom vibration inputs to a rigid plate. These force fields, including shear forces, could be used as a set of primitives for motion control to steer the formation of multiple objects.

Similarly, much recent work in magnet control has focused on exploiting inhomogeneities in the magnetic field to control multiple micro particles using gradient-based pulling [?, ?]. Unfortunately, using large-scale external magnetic fields makes it challenging to independently control more than one microrobot unless the distance between the electromagnetic coils is at the same length scales as the robot workspace [?, ?, ?]. In contrast, this paper requires only a controllable constant gradient in orthogonal directions to position the particles.

If a control input causes the particles to collide with obstacles at different times, inverting the control input does not undo the action. Due to this lack of time-reversibility, techniques that require a bidirectional graph, e.g. PRM [?] and RRT* [?] are not suitable. Instead, this paper employs a greedy search strategy.

3.3 Theory

In the absence of obstacles, uniform inputs move a swarm identically. Independent control requires breaking this symmetry. The following sections examine using non-slip boundary contacts to break the symmetry caused by uniform inputs. Our algorithms rely on holding one particle stationary by pushing it into the boundary while moving the other particle. We start with a boundary interaction model in subsection 3.3.1.

For some configurations, we can obtain the optimal solution. Subsections 3.3.2 and 3.3.3 provide shortest-path results for two representative workspaces, squares and disks.

3.3.1 Boundary Interaction Model

The system dynamics represent particle swarms in low-Reynolds number environments, where viscosity dominates inertial forces and so velocity is proportional to input force [?]. In this regime, the input force command $\mathbf{u}(t)$ controls the velocity of the particles. If the i^{th} particle has position $\mathbf{x}_i(t)$ and velocity $\dot{\mathbf{x}}_i(t)$, we assume the following system model:

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}(t) + F(\mathbf{x}_i(t), \mathbf{u}(t)), \quad i \in [1, n]. \quad (3.1)$$

$$F(\mathbf{x}_i(t), \mathbf{u}(t)) = \begin{cases} -\mathbf{u}(t) & \mathbf{x}_i(t) \in \text{boundary and} \\ & \mathbf{N}(\text{boundary}_{\mathbf{x}_i(t)}) \cdot \mathbf{u}(t) \leq 0 \\ 0 & \text{else.} \end{cases}$$

Here $F(\mathbf{x}_i(t), \mathbf{u}(t))$ is the frictional force provided by the boundary, and $\mathbf{N}(\text{boundary}_{\mathbf{x}_i(t)})$ is the normal to the boundary at position $\mathbf{x}_i(t)$.

The same model can be generalized to particles moved by fluid flow where the vector direction of fluid flow $\mathbf{u}(t)$ controls the velocity of particles, or for a swarm of particles that move at a constant speed in a direction specified by a uniform input $\mathbf{u}(t)$ [?]. As in our model, fluid flowing in a pipe has zero velocity along the boundary. Similar mechanical systems exist at larger scales, e.g. all tumblers of a combination lock move

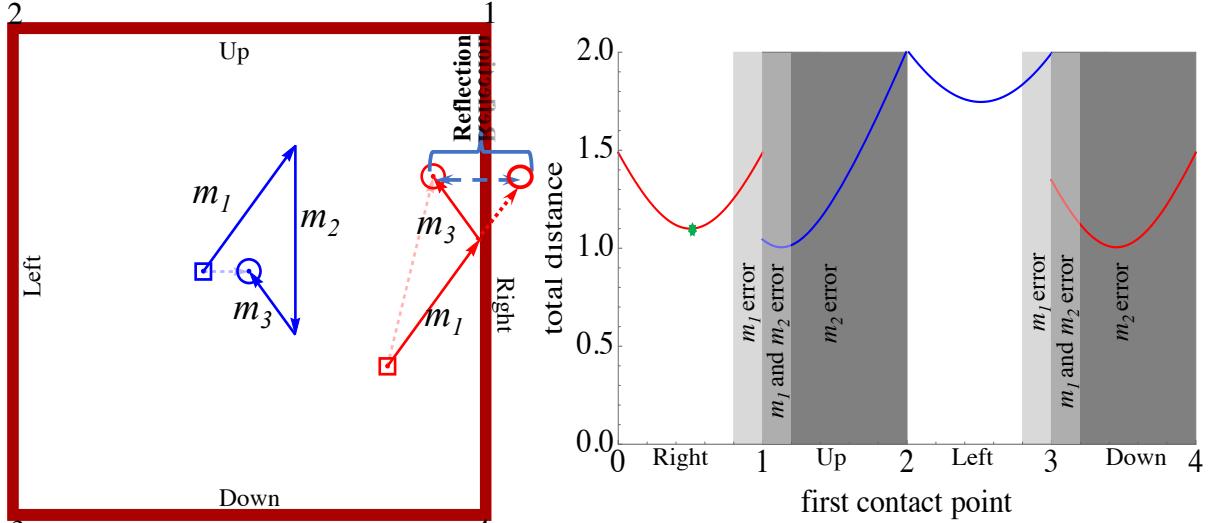


Figure 3.2: The shortest three-move path that reconfigures two particles has the property that the incident angle equals the reflected angle.

uniformly unless obstructed by an obstacle. Our control problem is to design the control inputs $\mathbf{u}(t)$ to deliver two particles to goal positions.

3.3.2 Example: Shortest Path in a Square Workspace

Changing the relative positions of particles in any workspace requires making one particle contact the boundary. If the goal configuration cannot be reached in one move but can be reached in three moves, the shortest path has a simple solution. The first move, m_1 , makes one particle contact a wall, m_2 adjusts the relative spacing error to zero, and m_3 takes the particles to their final positions. m_2 cannot be shortened, so optimization depends on choosing the location where the particle contacts the wall. Since the shortest distance between two points is a straight line, reflecting the goal position across the boundary wall and plotting a straight line gives the optimal contact location, as shown in Fig. 3.2. There are four walls, and four candidate solutions, but some candidate solutions may be invalid because a different boundary is hit before the desired first contact position in move m_1 (light grey regions) or invalid because m_2 cannot generate the goal relative spacing (dark grey regions).

This optimization result was implemented in our previous work using a square

workspace [?]. Fig. 3.3 shows solutions from a *Mathematica* implementation in a square workspace for six representative configurations.

3.3.3 Shortest Path in Unit Disk that Intersects Circumference

The shortest path between two points in the unit disk that intersects the circumference is composed of two straight line segments and has an optimal contact point, as shown in Fig. 3.4. The problem can be simplified by choosing the coordinate system carefully. We define the x -axis along the line from the circle center to the starting point: $S = (s, 0)$, and define the point of intersection by the angle θ from the x -axis: $P = (\cos \theta, \sin \theta)$. Define the final point E by a radius e and angle β : $E = e(\cos \beta, \sin \beta)$. Then the length of the two line segments is

$$\sqrt{(s - \cos \theta)^2 + \sin^2 \theta} + \sqrt{(e \cos \beta - \cos \theta)^2 + (e \sin \beta - \sin \theta)^2}, \quad (3.2)$$

which is minimized by choosing an appropriate θ value.

The length of the two line segments as a function of θ is drawn in the right plot of Fig. 3.4. There are several simple solutions. If s is 1 or e is 0 or β is 0, the optimal angle θ^* is 0. If e is 1 or s is 0, the optimal angle is β . Label the origin O . The optimal path satisfies the law of reflection off the unit circle, with angle of incidence equal to angle of reflection. The angle $\angle OPS$ (from the origin to P to S) is the same as the angle $\angle OPE$ (from the origin to P to E). We name these angles α . This can be proved by drawing an ellipse whose foci are S and E . When the ellipse is tangent to the circle, the point of tangency is P . Since the distance from the origin to P is always 1, we can set up three equalities using the law of sines: From triangle OSP : $\frac{\sin \alpha}{s} = \frac{\sin(\alpha+\theta)}{1} = \frac{\sin \theta}{\|SP\|}$, and from triangle OEP : $\frac{\sin \alpha}{e} = \frac{\sin(\beta-\theta)}{\|EP\|}$. If we mirror the point S about line \overline{OP} and label this point C , from triangle CEO : $\frac{\sin(\alpha+\theta)}{e} = \frac{\sin(2\theta-\beta)}{\|CE\|}$.

Simplifying this system of equations results in $s = e \csc \theta (s \sin(2\theta - \beta) + \sin(\beta - \theta))$. Solving this last equation results in a quartic solution that has a closed-form solution

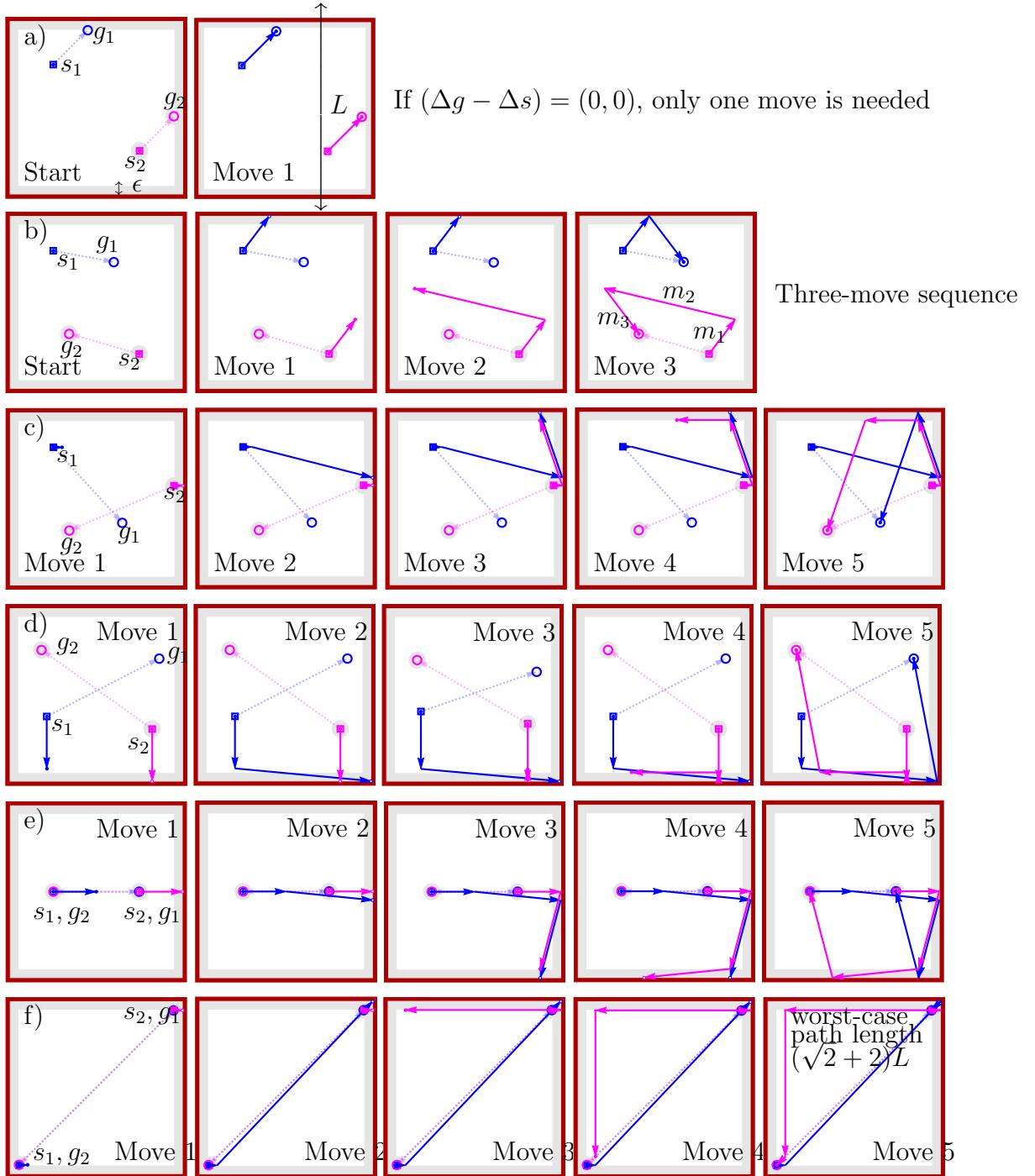


Figure 3.3: Frames from an implementation of Alg. 1: two particle positioning using walls with non-slip contacts. Particle start positions are shown by squares, and goal positions by circles. Dashed lines show the shortest route if particles could be controlled independently. Solid arrows show path given by Alg. 1.

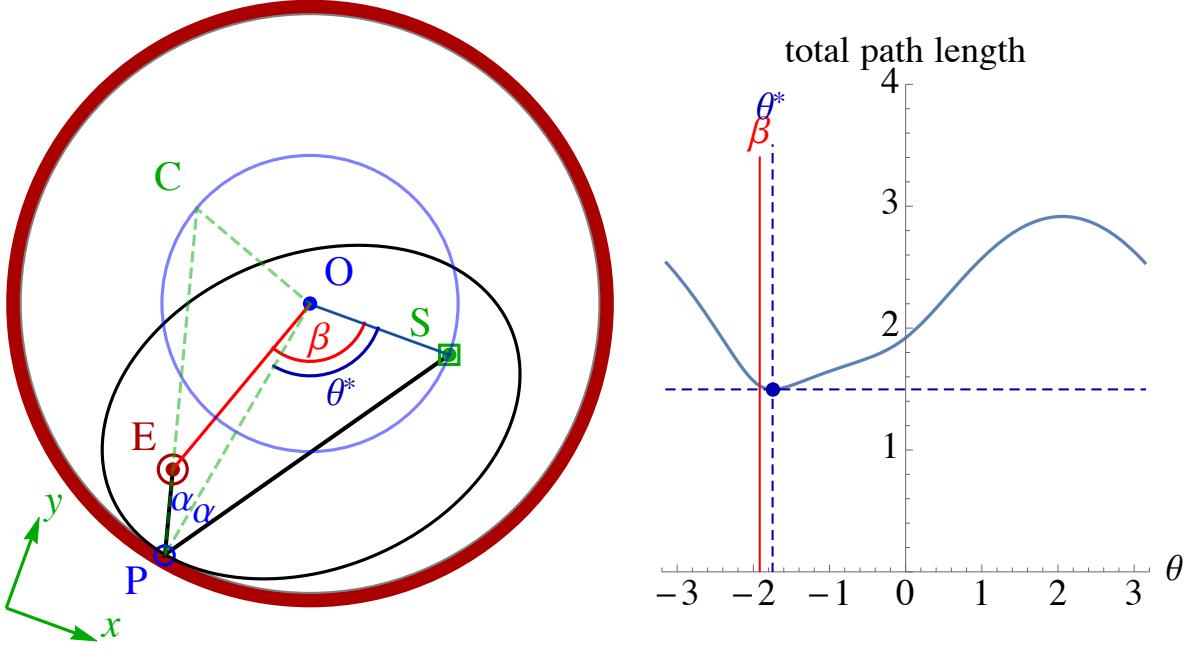


Figure 3.4: The shortest path between two points S to E in the unit disk that intersects the circumference. The path length as a function of intersection point, $P = (\cos \theta, \sin \theta)$ is shown at right. See [?].

with four roots, each of which can be either a clockwise or a counterclockwise rotation θ , depending on the sign of β , with $-\pi \leq \beta \leq \pi$. We evaluate each and select the solution that results in the shortest length path. For an interactive Mathematica demonstration of this shortest path, see [?]. Because the closed form solution is long, it is included in the paper attachments.

3.4 Position Control of Two Particles Using Boundary Interaction

This section presents algorithms that use non-slip contacts with walls to arbitrarily position two particles in a convex workspace. Workspaces are 2D convex polygons with no internal obstacles. Assume two particles are initialized at s_1 and s_2 with corresponding goal destinations g_1 and g_2 . Denote the current positions of the particles p_1 and p_2 . Values $.x$ and $.y$ denote the x and y coordinates, i.e., $p_1.x$ and $p_1.y$ denote the x and y

locations of p_1 . Algorithm 1 can now handle any convex workspace, including the special limit case of a circular workspace. In the last subsection we present techniques to control 3D positioning of two particles.

3.4.1 Δ Configuration Space

The configuration space for two particles is a four dimensional manifold. Translating both particles the same amount is a trivial operation, but changing the relative positions requires boundary interaction. For this reason, our algorithms use the two dimensional Δ configuration space. The Δ configuration space is a set of all possible Δp values, defined as the difference in position of the first particle from the second particle: $\Delta p = p_2 - p_1$. We use the Δ configuration space to plan move sequences that achieve the desired relative spacing. Once the particles have the correct relative spacing, they can be delivered to the goal configuration in one move.

The Δ configuration space for an n -sided convex polygon P can be constructed in a method analogous to computing configuration space obstacles for polygons [?]. Translate n copies of P so that each copy moves a different vertex of P to $(0, 0)$. Because P is convex, the convex-hull of all these translated vertices is the boundary of the Δ configuration space. For an n -sided convex polygon, the Δ configuration space is a $2n$ -sided convex polygon. Even-sided regular polygons are a special case in which half the sides align and the Δ configuration space is n -sided. An example Δ configuration space construction is shown in Fig. 3.5: a four-sided workspace is on the left, the four translated copies with dashed lines outlining the convex hull is in the middle, and the resulting Δ configuration space is on the right.

3.4.2 Two Particle Path Planning

The *2-move reachable set* is the locus of points in the Δ configuration space corresponding to any two-move sequence where the first move brings one particle into contact with the boundary, and the second move translates the second particle without moving the first. For the given Δs (starting configuration), the rightmost image of Fig. 3.5 draws the 2-move reachable sets in transparent blue. Figure 3.6 shows the starting and ending relative positions as Δs and Δg in the Δ configuration space. The next subsections give procedures to compute the 2-move reachable set.

The goal is to move the particles within δ of the goal positions using a shared control input where δ is an arbitrary small number. We do this by first moving them within δ of the correct relative position and then translating the particles to the goal. The relative position is $\|\Delta g - \Delta p\| = \|(g_2 - g_1) - (p_2 - p_1)\|$.

Algorithm 1 assigns a uniform control input at every instance. It first computes the 2-move reachable set. If the goal relative position is in the 2-move reachable set, we move particles to achieve that relative position. If it is not in the 2-move reachable set, we move particles to achieve the closest point on this reachable set from Δg , which is Δg_c .

Achieving a Δg_c configuration requires two-moves, the first to move until one particle touches a wall, and the second to adjust the relative spacing. Once the correct *relative* position has been achieved, a final translation delivers both particles to their goal destinations. Otherwise, we iterate until we reach the goal.

3.4.3 Convex Polygonal Workspaces: 2-Move Reachable Set

Figure 3.6 shows six workspaces, their Δ configuration spaces, and the 2-move reachable sets that correspond to representative initial conditions. Figure 3.7 highlights the construction of the 2-move reachable sets for a square workspace. There are four 2-

Algorithm 1 2-PARTICLEPATHPLANNING($s_1, s_2, g_1, g_2, P, \epsilon$)

Require: knowledge of starting (s_1, s_2) and goal (g_1, g_2) positions of two particles. P is a description of the workspace. ϵ is a positive error bound.

- 1: $(p_1, p_2) \leftarrow (s_1, s_2)$ ▷ p_1, p_2 are current positions
- 2: moves $\leftarrow \{\}$
- 3: $\Delta p \leftarrow p_2 - p_1$
- 4: $\Delta g \leftarrow g_2 - g_1$
- 5: **while** $\|\Delta p - \Delta g\| > \epsilon$ **do**
- 6: $R_{SET} \leftarrow$ Compute 2-move reachable set ▷ use Alg. 2 or 3
- 7: $\Delta g_c \leftarrow$ nearest point in R_{SET} to Δg
- 8: $m \leftarrow$ move-to-wall corresponding to Δg_c
- 9: moves \leftarrow Append m to moves
- 10: $(p_1, p_2) \leftarrow$ ApplyMove m to (p_1, p_2)
- 11: $\Delta p \leftarrow p_2 - p_1$
- 12: **end while**
- 13: moves \leftarrow Append $g_2 - p_2$ to moves ▷ translate to goal
- 14: **return** moves

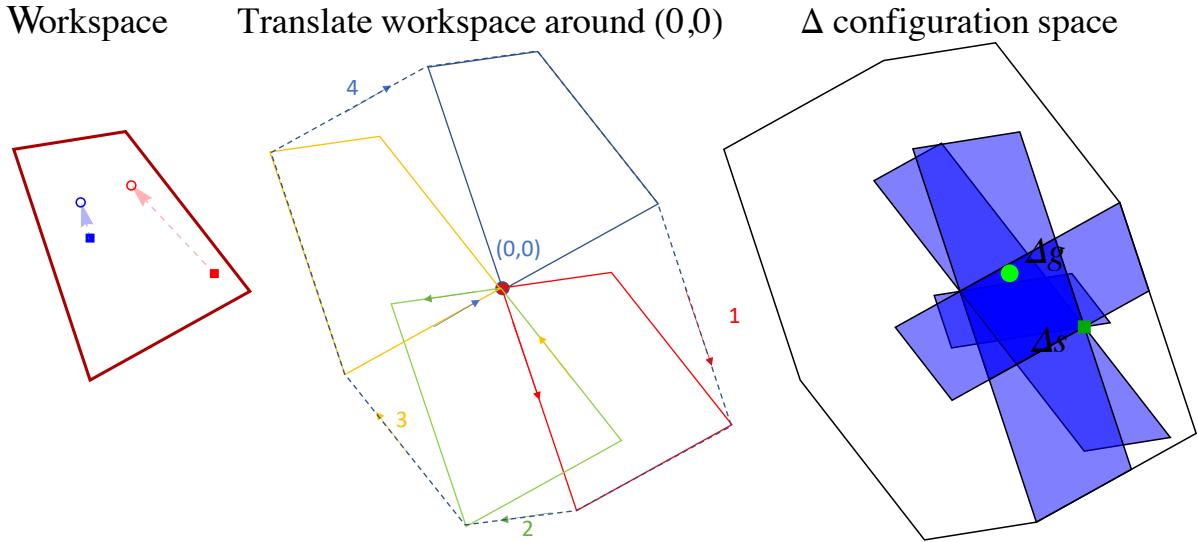


Figure 3.5: Workspace and Δ configuration space is shown for an arbitrary convex polygon with $n = 4$ sides. The 2-move reachable sets for this initial configuration (shown by square markers) are drawn in transparent blue.

move reachable sets, but the horizontal (and vertical) reachable sets are equivalent in the Δ configuration space so we can plan in this space and choose between the two options to minimize the total distance. Algorithm 2 computes the 2-move reachable set for any convex workspace. The set is constructed by considering each edge of the workspace. We name each vertex as p_i where $1 \leq i \leq n$. If one particle contacts edge $\overline{p_i p_{i+1}}$ before the

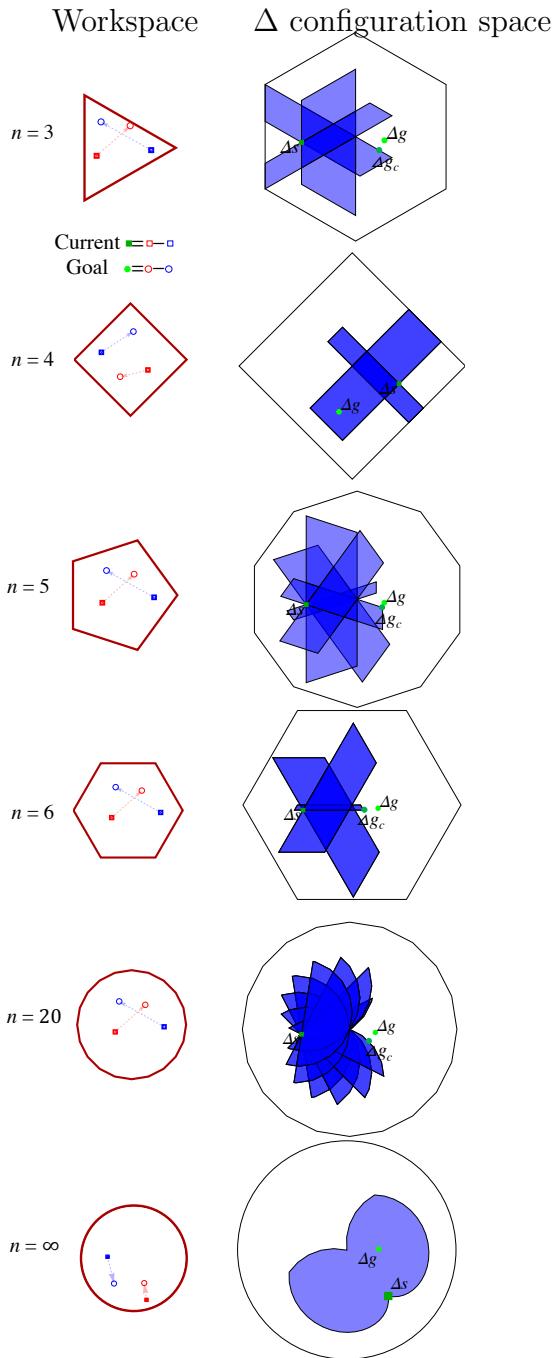


Figure 3.6: The Δ configuration space is all possible configurations of $p_2 - p_1$. The sets reachable in two moves, called *2-move reachable sets*, are drawn with transparent blue polygons. A polygon with n sides has n 2-move reachable sets, but if n is even and the polygon is regular, half the reachable sets overlap. If Δg is in the 2-move reachable sets, we can achieve the required relative position in two moves. If Δg is not in the 2-move reachable set, we define a temporary goal Δg_c (the closest point on the 2-move reachable set to Δg) and apply two moves to achieve Δg_c . We repeat this process until the relative goal position is achieved.

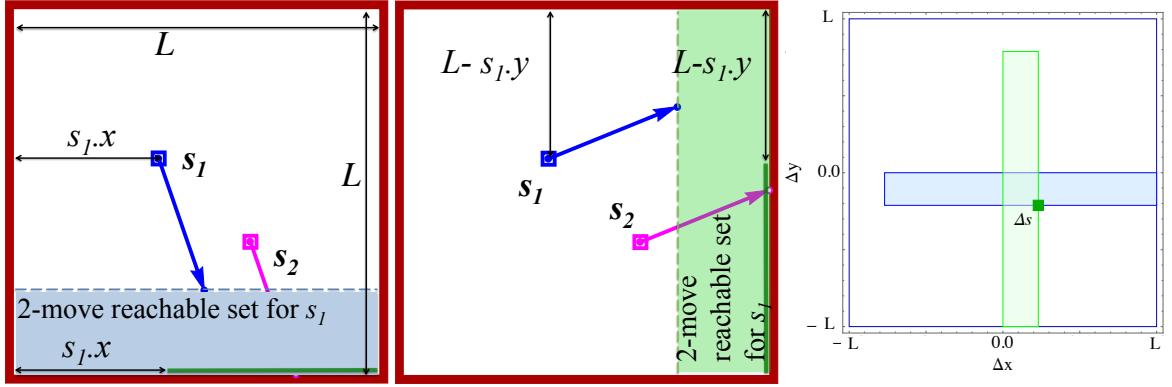


Figure 3.7: Boundary interaction is used to change the relative positions of the particles. Each particle gets the same control input. (left) If particle 2 contacts the bottom wall before particle 1 reaches a wall, particle 2 can reach anywhere along the green line, and particle 1 can move to anywhere in the shaded area. (middle) Similarly, if particle 2 contacts the right wall before particle 1 reaches a wall, particle 2 can reach anywhere along the green line, and then particle 1 can move to anywhere in the shaded area. (right) The 2-move reachable sets in the Δ configuration space is shown.

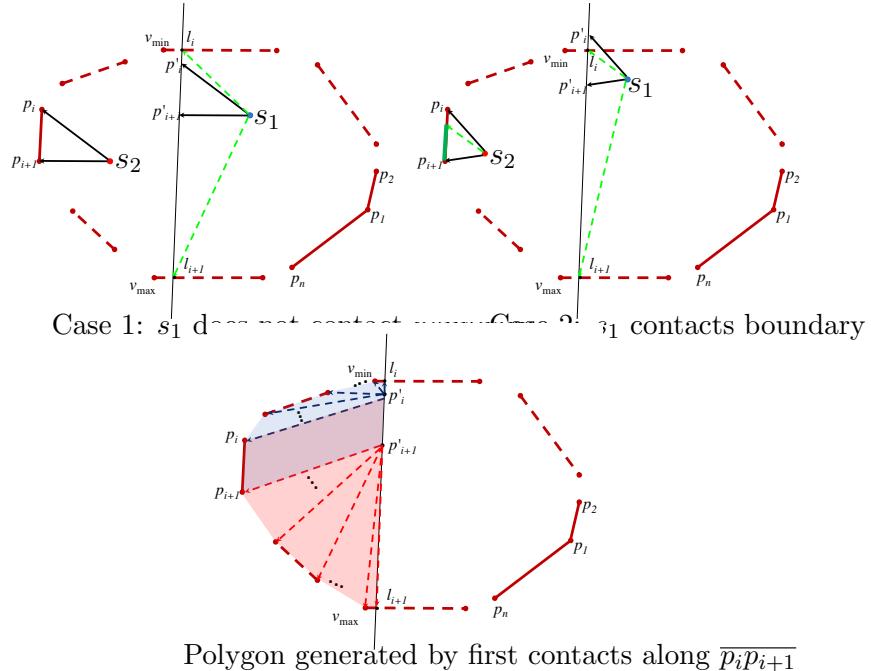


Figure 3.8: Steps to generate the 2-move reachable set when one particle collides with edge $\overline{p_i p_{i+1}}$ of a convex polygonal workspace.

other (one particle will always contact before the other unless the particles are parallel to the wall), the corresponding 2-move reachable set is a polygon, constructed in lines 2-13 of Alg. 2. The union of these polygons for all n sides is the 2-move reachable set of Δ configurations. Figure 3.8 illustrates the procedure to construct the 2-move reachable set generated by collisions with the $\overline{p_i p_{i+1}}$ edge.

Algorithm 2 REACHABLESETPOLYGON(s_1, s_2, g_1, g_2, P)

Require: knowledge of starting (s_1, s_2) and goal (g_1, g_2) positions of two particles. P is a list of the vertices of a convex polygon.

```

1:  $R_{\text{SET}} \leftarrow \{\}$ 
2: for  $p_i$  in  $P$  do
3:    $p'_i \leftarrow s_1 + s_2 - p_i$ 
4:    $p'_{i+1} \leftarrow s_1 + s_2 - p_{i+1}$ 
5:    $L \leftarrow \overline{p'_i p'_{i+1}}$                                  $\triangleright$  line  $(p'_i, p'_{i+1})$ 
6:    $l_i, l_{i+1} \leftarrow$  intersections of  $L$  and polygon  $P$ 
7:   if  $p'_i$  not inside polygon  $P$  then
8:      $p'_i \leftarrow l_i$ 
9:   end if
10:  if  $p'_{i+1}$  not inside polygon  $P$  then
11:     $p'_{i+1} \leftarrow l_{i+1}$ 
12:  end if
13:   $D \leftarrow s_2 - s_1 - ([l_i, v_{\min}, \dots, p_i] - p'_i,$ 
    $[p_{i+1}, p_{i+2}, \dots, v_{\max}, l_{i+1}] - p'_{i+1})$ 
14:   $R_{\text{SET}} \leftarrow$  Append polygon  $D$  to  $R_{\text{SET}}$ 
15: end for
16: Return  $R_{\text{SET}}$ 

```

3.4.4 Circular Workspaces: 2-Move Reachable Set

To compute the 2-move reachable set for a circular workspace, first consider all possible first contact locations. The set of boundary points that a particle can touch before the other particle touches are two arcs from ψ_{\min} to ψ_{\max} and from $\pi + \psi_{\min}$ to $\pi + \psi_{\max}$:

$$\psi \in [\psi_{\min}, \psi_{\max}] = \theta + [\sin^{-1}(\frac{d_{12}}{2r}) - \frac{\pi}{2}, \frac{\pi}{2} - \sin^{-1}(\frac{d_{12}}{2r})], \quad (3.3)$$

where $d_{12} = \|s_1 - s_2\|$, r is the radius of the workspace, and the angle between two particles is $\theta = \arctan(\frac{p_1.x - p_2.x}{p_1.y - p_2.y})$.

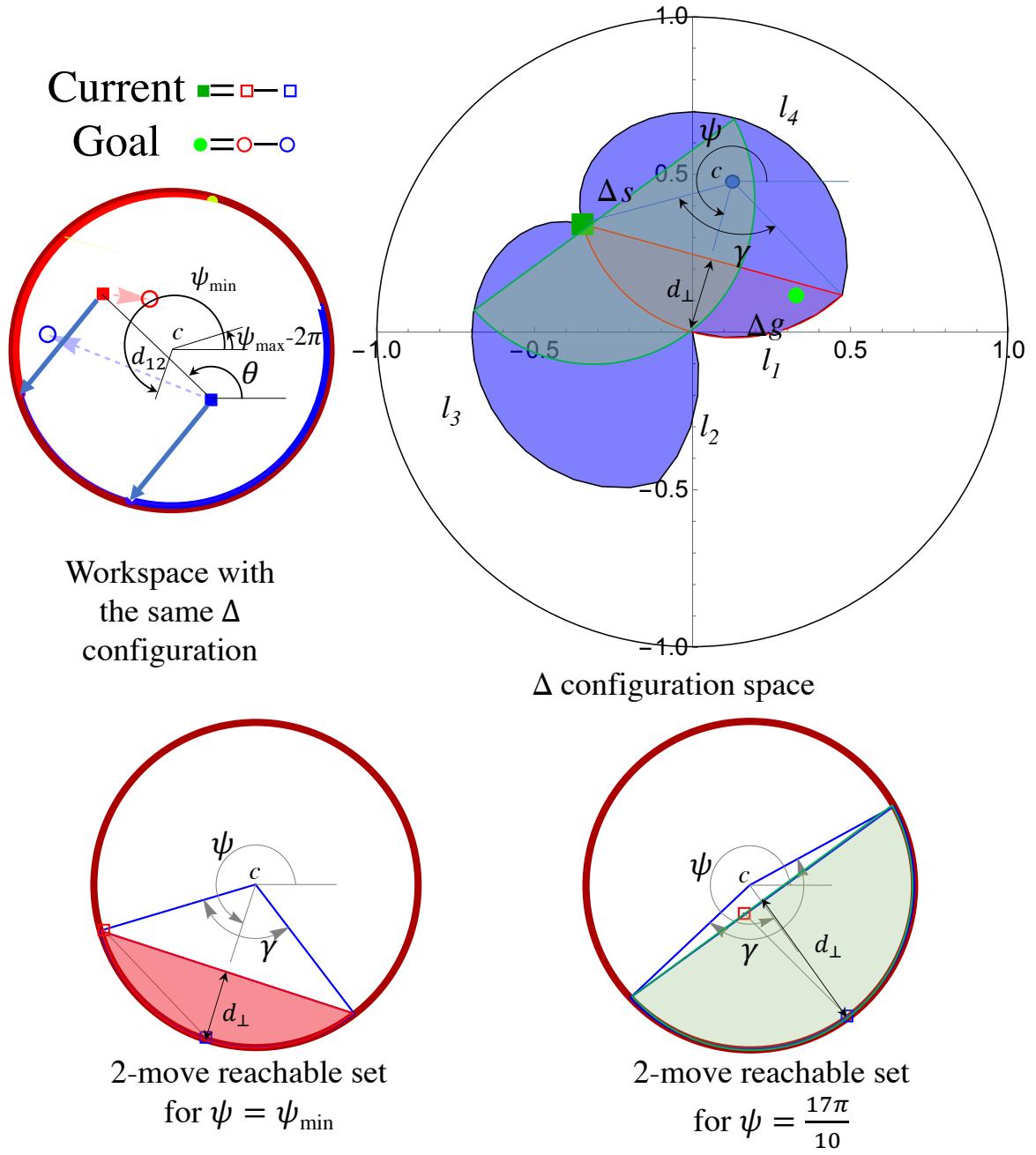


Figure 3.9: Left top: The possible first contact points for the blue and red particles are shown with blue and red arcs. Left bottom: if the blue particle touches the wall at ψ_{\min} (blue square) the other particle (red square) can move anywhere in the red region. Right bottom: if the blue particle touches the wall at $\psi = \frac{17\pi}{10}$ (blue square) the other particle (red square) can move anywhere in the green region. Right: The Δ configuration space for the corresponding starting positions of the particles is shown. The possible 2-move reachable sets before contact are shown in the Δ configuration as a blue region. If the blue particle contacts the boundary at ψ_{\min} , the reachable Δ configuration is the red set, or the green set if $\psi = \frac{17\pi}{10}$.

A circle has an infinite number of sides, thus infinite reachable sets. However, the 2-move reachable set can be parameterized by the angle of first contact location ψ , as shown in Fig. 3.9.

Each ψ value generates a 2-move reachable set that is a chord of the disk, with interior angle γ parameterized by ψ :

$$\gamma(\psi) = \cos^{-1} \left(1 - \frac{d_{\perp}(\psi)}{r} \right), \text{ where:} \quad (3.4)$$

$$d_{\perp}(\psi) = 2||s_1.p_{\psi}(\psi) - s_2.p_{\psi}(\psi)||, \quad (3.5)$$

$$p_{\psi}(\psi) = r[\cos(\psi), \sin(\psi)]. \quad (3.6)$$

The 2-move reachable sets with π difference in ψ value are equivalent in the Δ configuration space. The reachable Δ configuration set for any first contact point defined by ψ is the area under a chord from angle $\psi - \frac{\gamma(\psi)}{2}$ to $\psi + \frac{\gamma(\psi)}{2}$, for a circle of radius r centered at $c = r(\cos(\psi - \pi), \sin(\psi - \pi))$. Two such chords are drawn in red and green in Fig. 3.9.

The equations for the four lines outlining the union of two-move reachable sets are as follows:

$$l_1 = r \left(\cos \psi_{\min} - \cos(\gamma + \psi_{\min}) \right) \quad (3.7)$$

$$+ \sin \psi_{\min} - \sin(\gamma + \psi_{\min}) \quad 0 < \gamma < \gamma(\psi_{\min}),$$

$$l_2 = r \left(\cos \psi_{\max} - \cos(\gamma + \psi_{\max}) \right)$$

$$+ \sin \psi_{\max} - \sin(\gamma + \psi_{\max}) \quad \gamma(\psi_{\max}) < \gamma < 0,$$

$$l_3 = r \left(\cos \psi - \cos(\psi + \gamma(\psi)) \right)$$

$$+ \sin \psi - \sin(\psi + \gamma(\psi)) \quad \psi_{\min} < \psi < \psi_{\max},$$

$$l_4 = r \left(\cos \psi - \cos(\psi - \gamma(\psi)) \right)$$

$$+ \sin \psi - \sin(\psi - \gamma(\psi)) \quad \psi_{\min} < \psi < \psi_{\max}.$$

We combine these boundaries to compute the 2-move reachable set summarized in Alg. 3.

The motion-planner finds a ψ that would enable us to reach Δg_c , the nearest point in the 2-move reachable set to Δg . We first check if Δg_c is in the Δ configuration space chords defined by either ψ_{\min} or ψ_{\max} using the following two tests:

$$(\Delta g_c.x - c.x)^2 + (\Delta g_c.y - c.y)^2 > r^2 \text{ and} \quad (3.8)$$

$$(c.x - \Delta g_c.x) \cos \psi + (c.y - \Delta g_c.y) \sin \psi > r \cos \gamma.$$

If Δg_c is not in either chord, we draw a line from Δg_c to the current relative position, Δp . This line is a chord of the circle centered at c . The ψ to this chord obeys:

$$\psi = \tan^{-1} \left(\frac{\Delta p.x - \Delta g_c.x}{\Delta p.y - \Delta g_c.y} \right). \quad (3.9)$$

The particles achieve Δg_c in two moves. The first move causes one particle to touch the wall at p_ψ , (3.6). The second move achieves the required relative position.

Algorithm 3 REACHABLESETCIRCLE(s_1, s_2, g_1, g_2)

Require: knowledge of starting (s_1, s_2) and goal (g_1, g_2) positions of two particles.

- | | |
|---|----------------------------|
| 1: Calculate ψ_{\min} and ψ_{\max} | \triangleright use (3.3) |
| 2: Calculate $\gamma(\psi)$ | \triangleright use (3.4) |
| 3: Calculate l_1, l_2, l_3, l_4 | \triangleright use (3.7) |
| 4: Return the union of (l_1, l_2, l_3, l_4) | |
-

3.4.5 3D workspaces: Cylinders and Prisms

Extending path planning to 3D is possible only if the two particles do not initially have the same x and y positions. For ease of analysis, we assume the workspace boundaries extend in the $\pm z$ direction to form either right cylinders or right prisms. If the 3D projection is at a different angle, redefine the 2D workspace as a region perpendicular to the projection. First, we move the closest particle to the boundary, which prevents its z -coordinate from changing. We next apply actuation in either the $\pm z$ direction to achieve the desired Δz . Then the particles are actuated away from the boundary and

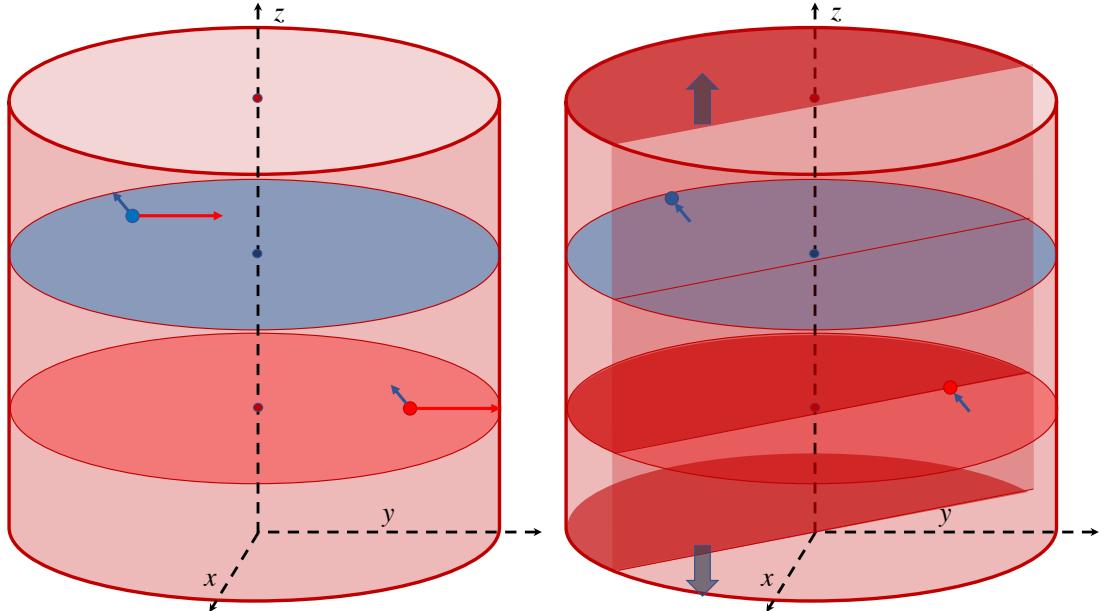


Figure 3.10: Illustration on how boundary contacts enable 3D positioning. Once one particle contacts a boundary, the other particle’s 2-move reachable set is a prism formed by extending the 2D 2-move reachable set in the $\pm z$ direction.

to the appropriate z positions. Path planning continues using Alg. 1 to position the particles to the desired x and y positions. As an example, consider Fig. 3.10 which shows a cylindrical workspace. The blue particle starts in the blue disk and the red particle starts in the red disk. The two candidate shortest-length paths that touch the wall are shown with parallel arrows. Each arrow will cause one of the particles to touch the wall, enabling the other particle to move freely in the z -axis to achieve the required relative position. This can be extended to other 3D workspaces if the workspace can be locally approximated as a 3D prism or cylinder. Other workspaces may be better handled by other path planners, such as [?], which used collisions with protrusions of the workspace to rearrange particles.

3.5 Simulation

Algorithm 1 was implemented in Mathematica using particles with zero radius. Figure 3.11 shows frames of the algorithm in two representative workspaces, square and

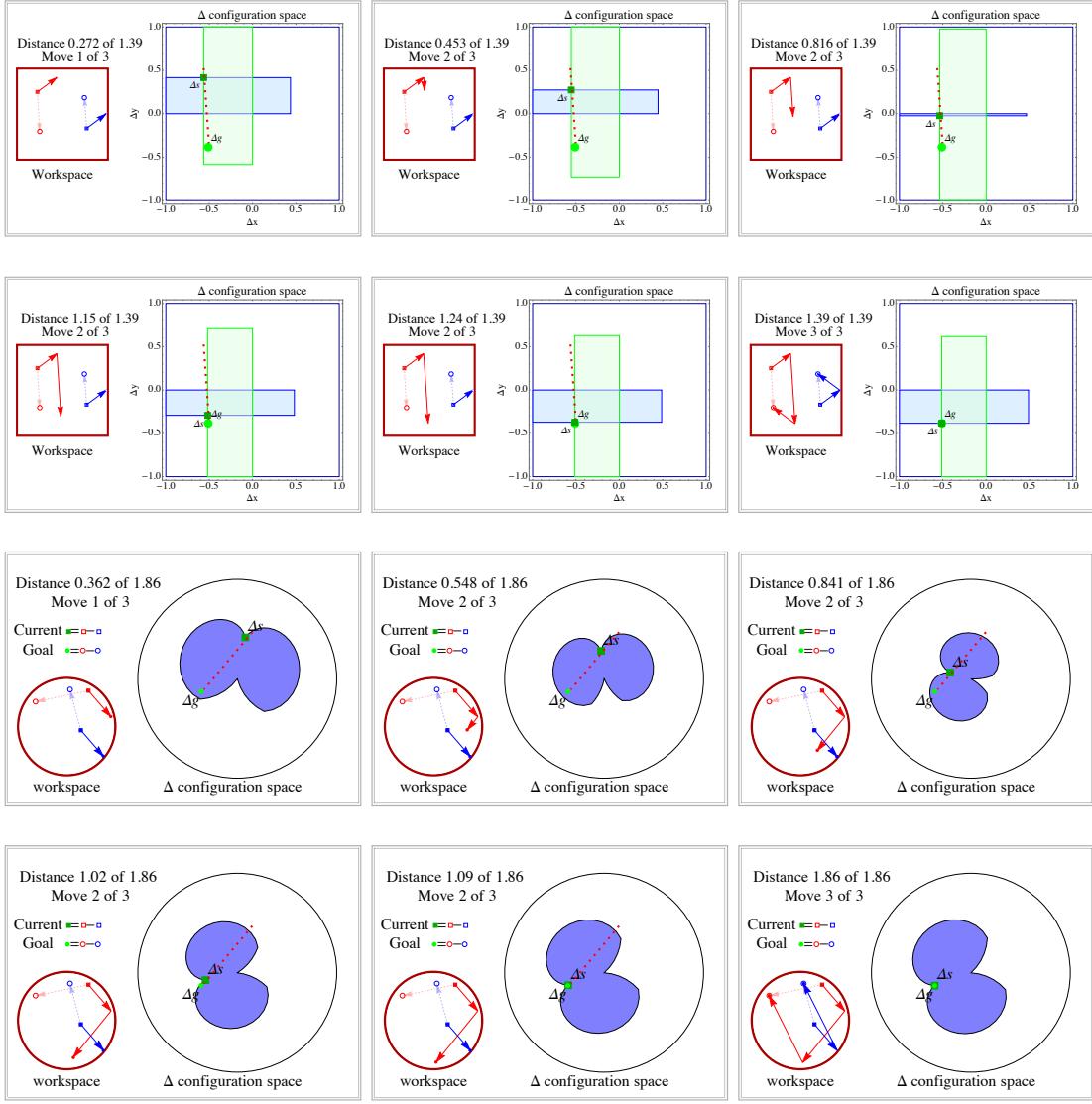


Figure 3.11: Frames from reconfiguring two particles. Top six images show a polygonal workspace and the corresponding Δ configuration space with its 2-move reachable sets. Bottom six images show a disk-shaped workspace and the corresponding Δ configuration space with its 2-move reachable sets. For each the move 1 and 3 are simple translations of both particles and so the reachable sets do not change. The reachable set morphs during move 2 because one particle is held stationary by the boundary. See multimedia attachment for animations of each.

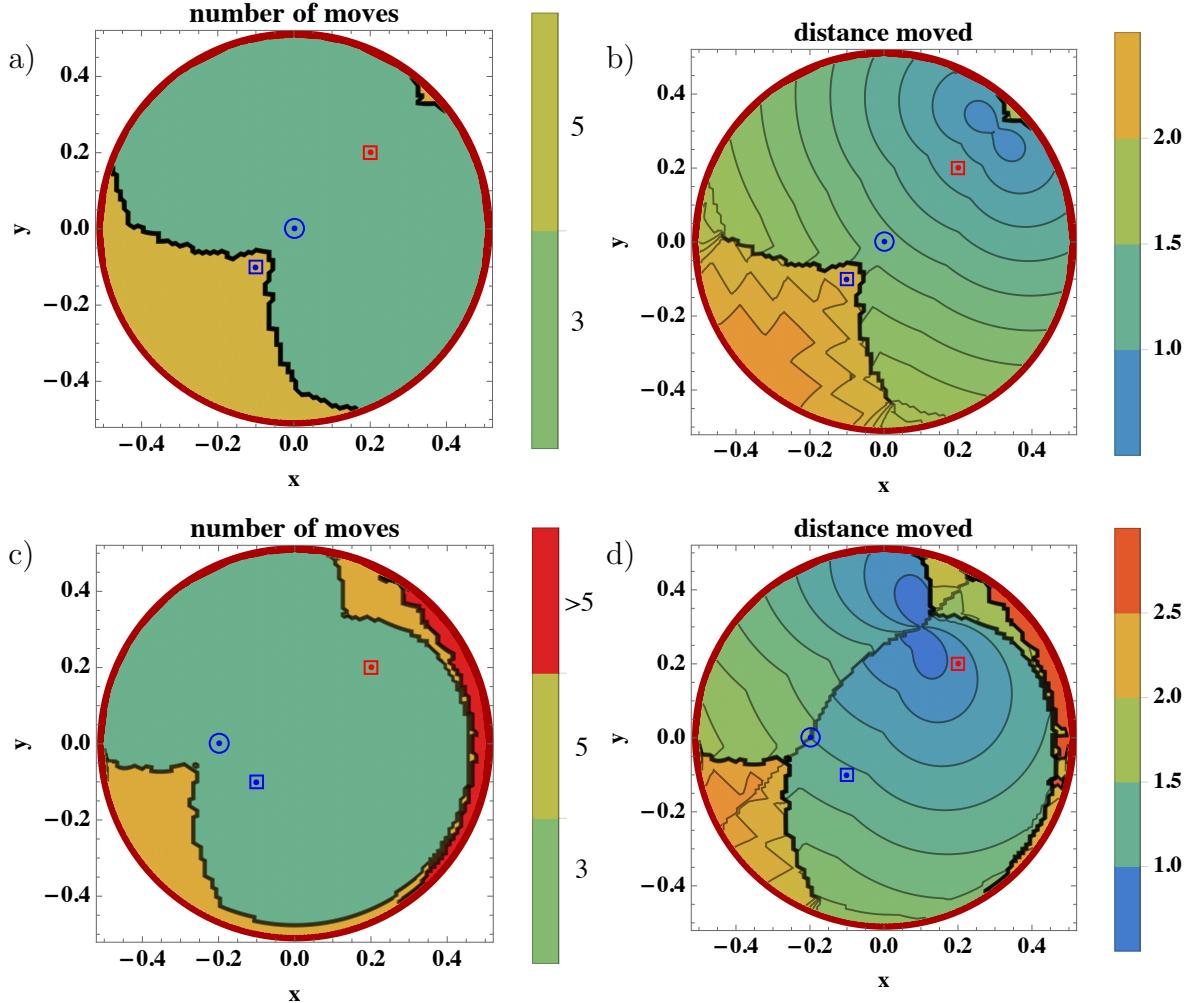


Figure 3.12: Plots show performance with one goal on the boundary.

disk, with two arbitrary starting and goal configurations.

The contour plots in Fig. 3.12 left show the length of the path for two different settings. Top row considers $\{s_1, s_2, g_1\} = \{(0.2, 0.2), (-0.1, -0.1), (0, 0)\}$ and bottom row considers $\{s_1, s_2, g_1\} = \{(0.2, 0.2), (-0.1, -0.1), (-0.2, 0)\}$ each in a workspace with $r = 0.5$, and g_2 ranging over all the workspace. Fig. 3.12 right shows the number of moves and left shows the total distance of the path. This plot shows the nonlinear nature of the path planning. When the goal is in the middle of the workspace, a symmetry in the path length is expected as the top row shows. The bottom row shows a shift in the goal position which breaks the symmetry of the path length in the workspace.

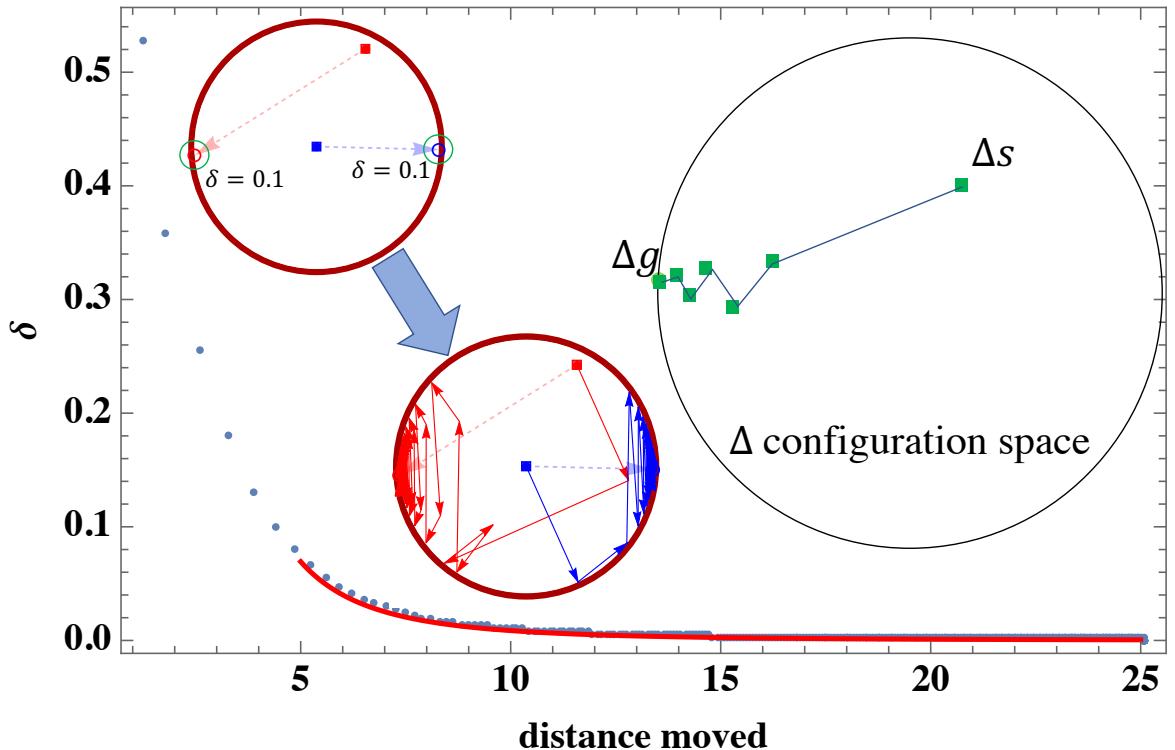


Figure 3.13: The worst-case path length occurs when particles must swap antipodes. This can never be achieved but can be asymptotically approached. Plot shows decreasing error as the number of moves grows.

The worst-case occurs when the ending points are at antipodes along the boundary (π angular distance). This can never be achieved but can be asymptotically approached as shown in Fig. 3.13. Figure 3.14 shows the same concepts in a square workspace. Figure 3.14 top and middle row considers the particles for three arbitrary starting and goal positions for the particles. Thus far, this paper has considered the particles to be unique. If particles are interchangeable, the path lengths often decrease. The bottom row of Fig. 3.14 considers interchangeable particles with the same configuration as the middle row with unique particles. The worst-case path lengths decrease by 33%, 60%, and 30% for the three test cases shown.

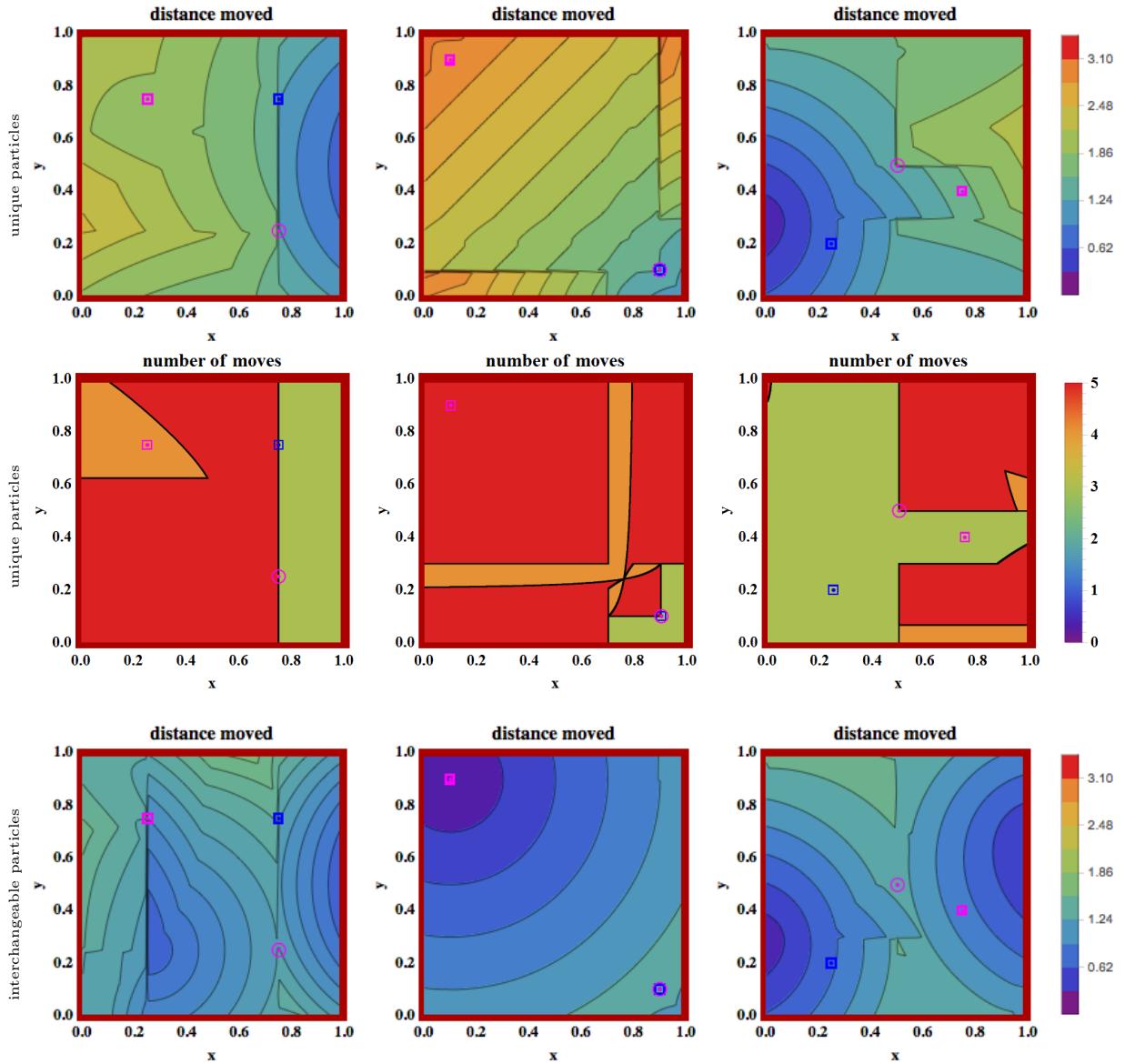


Figure 3.14: Starting positions of particles 1 and 2 and goal position of particle 2 are fixed, and $\epsilon = 0.001$. The top row of contour plots show the distance if particle 1's goal position is varied in x and y . The middle row shows the number of moves required for the same configurations. The bottom row shows the same configuration but when the particles are interchangeable.

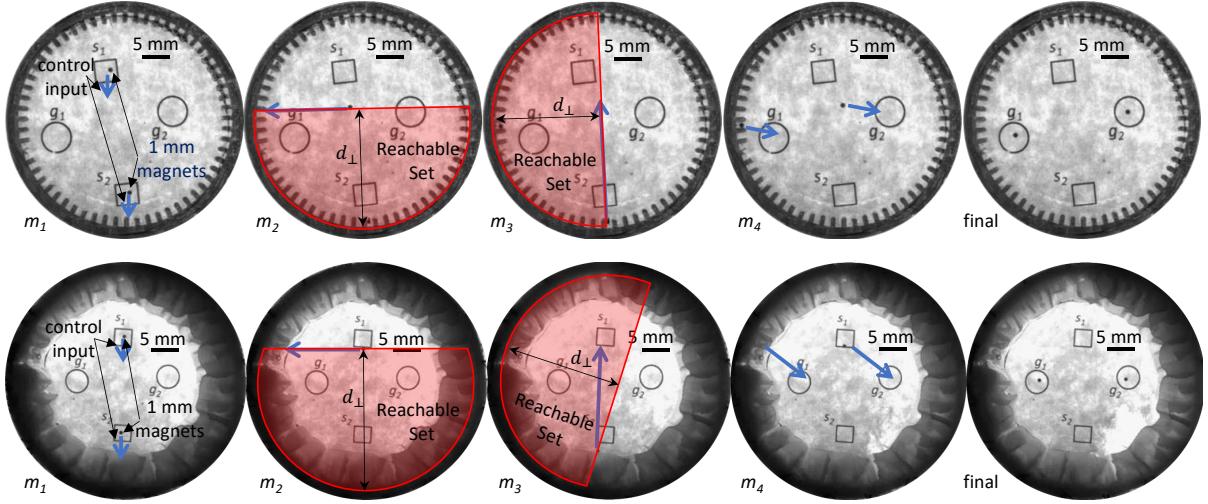


Figure 3.15: Frames showing particle positions before and after control inputs. Top row: small intestine phantom. Bottom row: cow stomach tissue.

3.6 Experimental Results

To demonstrate Alg. 1 experimentally, we performed several tests. Each used the same magnetic setup shown in Fig. 3.1. Two different intestine models were employed, the first a 3D-printed cross-section representation of a small intestine, and the second a cross-section of a bovine stomach.

3.6.1 Magnetic Manipulation Setup

The magnetic manipulation system has two pairs of electromagnetic coils, each with iron cores at their centers, and arranged orthogonal to each other. The iron core at the center of each coil concentrates the magnetic field towards the workspace. An Arduino and four SyRen regenerative motor drivers were used for control inputs to the coils. Finally, a FOculus F0134SB 659 x 494 pixel camera was attached to the top of the system, focusing on the workspace which was backlit by a 15 W LED light strip.

To obtain experimental data, the test samples (the phantom intestine model and the bovine cross section) were placed in laser-cut acrylic discs and then immersed in corn syrup. Corn syrup was used to increase the viscosity to 12000 cP for the experiments.

Spherical 1 mm magnets (supermagnetman #SP0100-50) were used as our particles. Our experimental setup did not perfectly implement the system dynamics in (3.1). In particular, the magnetic field in this setup is only approximately uniform. The magnetic force varies in both magnitude and orientation. As shown in the video attachment, this non-uniformity causes the particle closer to the coil to move faster than the other particle. This phenomenon makes it easier to increase particle separation than to decrease separation, but this can be compensated because boundary collisions easily decrease the separation. Also, magnetic forces are not exactly parallel, but point toward the center of the activated coil. Algorithm 1 still works despite these non-uniformities, but sometimes requires additional iterations.

3.6.2 Intestine Phantom Model

The intestine phantom model was used first and was made to mimic the geometry of an intestine and its villi. The model consists of a circular ring with an outer diameter of 50 mm, an inner diameter of 46 mm, and 60 2 mm long protrusions on its inner surface cut out of 6 mm thick acrylic to model the geometry of intestinal villi. Figure 3.15 top row shows an experiment. Starting and ending positions were printed beneath the workspace on transparency film. Our algorithm successfully delivered the particles to goal positions in 10 out of 10 trials.

3.6.3 Bovine Stomach Cross-section

Strips of cow stomach approximately 5 mm thick were cut and sewn to acrylic cylinder and then glued to an acrylic substrate using cyanoacrylate (superglue). This assembly was then filled with corn syrup. The experiment is shown in Fig. 3.15 bottom row. Our algorithm successfully delivered the particles to goal positions in 5 out of 5 trials. A video showing one trial of this experiment is available in the supplementary materials.

3.7 Conclusion and Future Work

This paper presented techniques for controlling the positions of two particles using uniform inputs and non-slip boundary contacts. The paper provided algorithms for precise position control. The algorithms relied on calculating reachable sets in a 2D Δ configuration space. Extending Alg. 1 to 3D was straightforward, but increased the complexity. Hardware experiments illustrated the algorithms in ex vivo and in artificial workspaces that mimic the geometry of biological tissue.

There are several avenues for future work. This paper assumed friction was sufficient to completely stop particles in contact with the boundary. The algorithms would require retooling to handle small friction coefficients. The techniques in [?] and [?] could be applied to extend the analysis to more than two particles.

Chapter 4

Low-Cost Robot Manipulator Arms

The course Introduction to robots, focuses on robotic manipulators. As a TA for this class, I designed a series of labs using a cheap (under US \$50) robot arm kit.

The kit, as purchased, is not a robot. It is instead a device with five motors and the user has a small controller that can turn motors on and off. The main aim of the labs are to introduce the students to robotics concepts and control system. A GUI was also implemented for lower level students to introduce them to programming and cultivate interest in robotics.

My labs are arranged as follows:

1. Lab 1: Assembly and familiarization
2. Lab 2: Open-loop control
3. Lab 3: Closed-loop control with Potentiometer sensors
4. Lab 4: Closed-loop control with image processing



Figure 4.1: The unmodified Robotic arm



Figure 4.2: The parts of the robot arm before assembly

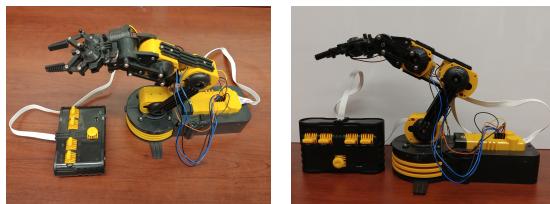


Figure 4.3: The Assembled Robotic arm

4.1 Lab 1: Assembly and familiarization

For this lab, the students were tasked with assembling the robot and testing it with its default controller. They were then tasked with moving around little objects within the robot's workspace to get an idea of the robot's capabilities. By assembling the robot themselves, the students become intimately familiar with the limits of the robots, and its various possible orientations.

4.2 Lab 2: Open loop control

For this lab, the students were tasked with altering the robot to include a microprocessor for open loop control. Using a motor driver and an arduino microprocessor, the students were instructed on how to include a degree of automation to the robot's movements. The main aim of this project was to showcase the ease to which automation can be achieved, but also reveal the need for feedback.

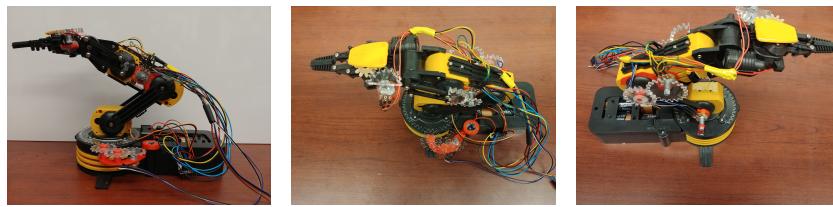


Figure 4.4: The Augmented Robotic arm

4.3 Lab 3: Closed-loop control with Potentiometer sensors

For this lab, the students were tasked with altering the robot further to include potentiometers to read its joint values and thus implement closed loop control with forward kinematics. The students were provided with instructions on how to attach the potentiometers to the robot with some custom designed 3D printed parts and lasercut gears. The aim of this lab was to help students understand the importance of implementing feedback in robotics control systems, and to introduce them to parts fabrication to a certain extent.

4.4 Lab 4: Closed-loop control with image processing

For this lab, the students were tasked using camera inputs to control the robot in a simple tracking task. This illustrates closed loop control with inverse kinematics.

4.5 Pre-semester prep

Before the semester begins, the acrylic gears are cut, the 3D printed parts are made, the jumper cables, the custom PCB circuit board and a few other miscellaneous parts for the augmentation are ordered.

4.6 Results

At the end of the labs, the students leave the class with experience constructing and programming a robot arm. In the course of doing this, they learn robotics terminology, and have an idea about different means of controlling robots and implementing sensor feedback.

Chapter 5

Conclusion

This is where you draw your conclusions.

