Next-Generation 6-DoF Handheld Manipulator for Microsurgery

Christian Berger

CMU-RI-TR-24-42

July 30, 2024

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Cameron Riviere, *Chair*
Zackory Erickson
Arpita Routray

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my parents, John and Caroline, who supported me every step of the way.

Abstract

Robotic assistance is used today in a variety of surgeries as a means of precise, dexterous, and minimally-invasive manipulation. However, practical use in microsurgical environments such as vitreoretinal surgery remains a challenge for the most common mechanically-grounded robotic platforms. Microsurgery requires micron-level accuracy and the ability to manipulate with interaction forces in millinewtons. Vitreoretinal surgery specifically requires a line of sight through the lens of the eye and maintaining a remote center of motion (RCM) at the incision point in the eye.

A handheld manipulator, Micron, has been presented to address the unique challenges of microsurgery. It enables the manipulator to scale down a surgeon's motion, actively cancel their involuntary tremor, and resist moving the tooltip into unsafe areas. Since the manipulator is handheld, it does not significantly obstruct the operating room and allows the surgeon to still feel haptic feedback during surgery. The manipulator is also capable of moving around an RCM at the sclerotomy. However, the manipulator's mechanical design necessitated unreliable motors that have a low stall force and a more reliable iteration was needed.

This thesis describes the construction of a new iteration of Micron based on the KIST microsurgical robot and integrated with our group's control systems and optical sensor. It begins by investigating the applicability to Micron of a design based on parallel continuum manipulators. The kinematics and dynamics of a parallel continuum manipulator are simulated and a further simulation is built to evaluate the stiffness of the designs. Although the design ultimately proves infeasible, the analysis settles the question for future research. Two simulations of the rigid-link Micron, built in Gazebo and PyBullet, are also created and evaluated as first steps toward a full kinematics and dynamics simulation of the manipulator. The PyBullet simulation is simplified and automated to pick up a small object on a model of the retina. Most importantly, the physical manipulator is constructed, including all its mechanical and electrical parts, as well as firmware to integrate it into our group's control software and optical sensor. The manipulator is actuated and its pose measurements are analyzed.

Acknowledgments

I've benefited from the support of countless people through my academic journey, so I'd like to take this opportunity to thank a few of them.

First, I want to thank my advisor, Dr. Cameron Riviere, for his mentorship and support over the last two years. His depth of experience in the surgical robotics field has been an invaluable resource, enabling me both to perform research in the field and to start my career in the industry. I'm deeply grateful for the kindness he has shown me.

I'd also like to thank Dr. Sungwook Yang for the information he provided us about his microsurgical robot and Rob MacLachlan for his assistance integrating the new manipulator into the control systems he designed. This project would not have been possible without their help. I want to thank Dr. Zackory Erickson and Arpita Routray, as well, for agreeing to serve on my thesis committee.

I've collaborated with a number of people on this project, none more than Cara Jaroenkunathum. Her mechanical experience made this project possible. More than that, though, her presence and personality made the work enjoyable and saved me from toiling alone in our basement lab. I can't say thanks enough for her help. The list of collaborators is long, but I'll do my best to name them here. My thanks to Yurika Yamada, Sterling McTee, Damian Cross, Sreeram Thirupathi, Zulekha Karachiwalla, Abena Boadi-Agyemang, Haoru Xue, and Arturo di Girolamo.

When I arrived in Pittsburgh, I had the great fortune to be accepted by a wonderful community of friends. I especially want to thank Sayan, Oliver, and Iqui for the many dinners, excursions, and those late-night study sessions. I couldn't have survived that first semester without them. Numerous other experiences, from skiing and kayaking with Aman to driving lessons with Adi and lunches with Viekash, were a welcome reprieve from research.

My biggest thanks goes out to my family and to my partner. None of this would have been possible without the support of my parents, John and Caroline, and my siblings, Lauren and Evan. Whether it was sending me cookies, bouncing around ideas on the phone, or flying out immediately to support me after emergency surgery, they were there for me every step of the way. Finally, I want to thank my partner, Jacob. Your constant push to approach life with greater joy and ease has made these past few years the happiest of my life.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Hand tremor is an inescapable part of any standard surgical procedure. However, surgeries in the eye often require the manipulation of handheld tools within a workspace on the order of micrometers [16]. Such workspaces are significantly smaller than the peak-to-peak range of hand tremor, which is typically on the order of hundreds of micrometers [70], [69]. This low signal-to-noise ratio makes it very difficult to distinguish voluntary hand motions from involuntary hand tremor by amplitude alone. However, several techniques make it possible to identify and limit the impact of unsafe motions of the tooltip. First, the motions can be distinguished and by their different frequency ranges. The human eye-hand feedback loop has a critical frequency in the range of 0.5-2 Hz [53], while involuntary hand tremor generally lies around the range of 7-17 Hz [76]. Second, computer vision algorithms can identify regions inside the eye that should be out of bounds [7]. Third, scaling down the motion of a surgeon's hand reduces the impact of accidental motions [65]. Microsurgical procedures could be made safer and more consistent if inadvertent or unsafe motions applied by a surgeon's hand were actively canceled prior to reaching the tooltip of the manipulator.

Several robotic platforms have been developed to minimize tremor during microsurgical procedures. Given their prominence in other medical robotics applications, teleoperated platforms were investigated early. The PRECEYES Surgical System, for example, was used to perform the first ever robot-assisted intraocular surgery [55].

However, mechanically grounded platforms entail significant risk whenever a patient moves unexpectedly [60]. Their large range of motion and high inertia [75] is another potential risk. Tissue in the eye does not regenerate if injured [36].

The problems inherent in mechanically grounded instruments can be avoided if the instrument is handheld. Such a handheld tremor-cancelling device, called Micron, has been developed by our group. An early 3-degree-of-freedom (3-DoF) version of Micron, shown in Figure 1.1, is described in [53]. The device aims to mimic a handheld tool with interchangeable end-effectors, the exception being that it can control the pose of the tooltip within a small workspace. The tooltip sits on the top platform of a parallel platform manipulator. Various configurations of this parallel platform have been developed. The first 6-DoF version is described in [94] and is also shown in Figure 1.1. Its tiny motors are placed in the links of the parallel manipulator. Having 6 degrees of freedom allowed this version of Micron to establish a remote center of motion at the incision point in the sclera to avoid tearing it as the instrument moved around in the eye. However, both this version of Micron and a subsequent version of Micron designed by New Scale Technologies [37] use tiny SQUIGGLE ultrasonic piezoelectric linear actuators, which had reduced force capability and were unreliable in operation.

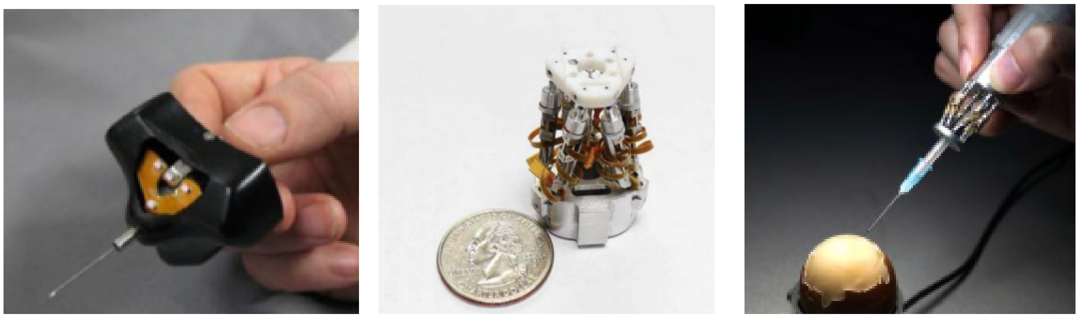


Figure 1.1: Iterations and descendants of Micron, as described in [53], [94], and [46], respectively.

A descendant of Micron has been developed in the ANSUR Laboratory at the Korean Institute of Science and Technology (KIST) and is shown in Figure 1.1. The KIST microsurgical robot converts to a 6-Prismatic-Universal-Spherical (6-PUS) joint system with new, more reliable motors from Maxon [46]. These motors and their custom lead screws can withstand an order of magnitude greater side load on the intraocular shaft. The increased stiffness and reliability of the motors made the manipulator an appealing basis for the next iteration of Micron. Fortunately, Dr. Sungwook Yang at the ANSUR Laboratory agreed to collaborate in this project. However, the electromagnetic (EM) tracker that the manipulator uses is less accurate and provides less information than the optical sensor our group has used on previous iterations of Micron. That optical sensor, the Apparatus to Sense Accuracy of Position (ASAP), has a resolution of 4 μm within its workspace and can track the 6-degree-of-freedom pose of both the handle and end-effector [52]. A new iteration of Micron that merges the mechanical design of the KIST microsurgical robot with the tracking abilities of the ASAP sensor would be a significant step forward in this research.

1.2 Contributions

The contributions in this thesis are as follows:

1. The feasibility of a new mechanical design for Micron based on prior work on parallel continuum manipulators is thoroughly investigated.
2. Preliminary simulations of the new iteration of Micron are created and a first step is taken towards automating a surgical sub-task in simulation.
3. The next iteration of Micron is constructed and actuated. The feedback from the ASAP optical sensor is analyzed.

The first contribution stems from a body of work that applies parallel continuum manipulators, which actuate parallel flexible legs, to various surgical instruments such as colonoscopes [10]. Small flexures have been used in multiple iterations of Micron [22, 53, 94]. However, they have not been applied to a 6-PUS system such as the KIST microsurgical robot. The kinematics and dynamics of parallel continuum manipulators were investigated and a simulation was run to test how stiff the manipulator could be made while still maintaining a sufficiently large workspace. Although the design ultimately proved infeasible, the analysis settles the question for future research.

Micron had never been simulated in a full kinematics and dynamics simulation. With the surgical robotics field increasingly moving towards the automation of surgical sub-tasks such as suturing [62], it was an opportune time to take the first steps. Two simulations are created and evaluated, one in Gazebo and one in PyBullet, with the Gazebo simulation showing more promise. Hardware delays and other time constraints prevented incorporation of realistic dynamics information into the manipulator's joints. Taking these constraints into account, the first step toward automating a surgical sub-task is taken with the PyBullet simulation. Specifically, a heavily simplified version of the manipulator is automated to grasp a small object on a model retina.

Although the hardware was delayed, it arrived soon enough for the third contribution to be made. The mechanical parts of the manipulator were given tolerances, ordered, and then assembled in the lab. Several custom PCBs were designed and assembled to route signals to and from the manipulator. Additional mechanical parts were designed and constructed to mount onto the manipulator and hold the electronics. Firmware was written to connect the motor drivers to our group's Micron control systems and the ASAP optical sensor. ASAP was calibrated for the new device and its measurements were analyzed while the manipulator was moved.

Chapter 2

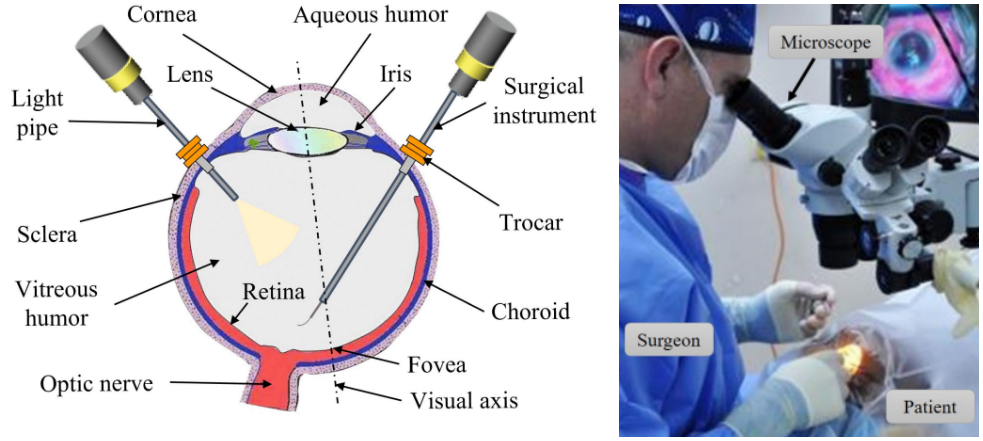
Background

2.1 Retinal Surgery

Vision impairment is associated with a reduction in social participation, educational performance, and opportunities for employment [17, 50, 71]. Globally, the World Health Organization estimates that more than 2.2 billion people have a vision impairment, with at least 1 billion of these being preventable [4]. Of those with preventable impairment, 8 million people have vision-threatening age-related macular degeneration and 3.9 million people have vision-threatening diabetic retinopathy [78]. Both of these conditions are expected to afflict more people in the future as life expectancies increase and more people develop diabetes [51]. Diabetic retinopathy alone has been projected to afflict 3.4 million Americans by 2050, a threefold increase from 2005 [68].

Vitreoretinal surgery is used to treat vision impairment caused by conditions such as diabetic retinopathy and macular degeneration. The most common method involves a pars plana vitrectomy, shown in Figure 2.1 [91]. Three ports are placed into the sclera: one for illumination inside the eye, one for a surgical instrument, and one for an infusion cannula. The cannula is used to remove the the vitreous humor and replace it with another liquid that provides better visibility of the retina. The light pipe and surgical instrument are actively operated by the surgeon. Depending on the specific procedure, the surgical instrument could be a needle, forceps, or laser probe. The surgeon views the procedure through a stereo microscope.

Typical treatments in vitreoretinal surgery include intraocular laser photocoagu-



(a)

(b)

Figure 2.1: Pars plana vitrectomy. (a) An illustration of the vitrectomy seen in a human eye cross section. (b) Typical surgical operating setup during a vitrectomy. Photo credit [38, 43].

lation and epiretinal membrane peeling. Epiretinal membrane peeling is the most common vitreoretinal surgery performed [33] and is required in the treatment of advanced diabetic retinopathy [73]. After a pars plana vitrectomy, the surgeon must lift and peel the epiretinal membrane away from the retina [21]. This membrane is incredibly thin, measured at $61 \pm 28\mu$m, making the peel very challenging [89]. Laser photocoagulation can impede the growth of abnormal retinal vessels or fix the retina in place to treat diabetic retinopathy and tears in the retina [11]. However, accidental photocoagulation in the wrong place can cause vision loss, making the accuracy of the laser aiming critical [31].

Retinal vessel cannulation is an experimental procedure with significant potential. Cannulation could be used to treat retinal vein occlusion (RVO), arteriovenous malformations, and retinal macroaneurysms [25]. RVO is of particular interest because it is the second most common retinal vascular disease [66]. It currently has no proven effective treatment [8]. Cannulation of retinal vessels afflicted by RVO allows the injection of clot-dislodging or clot-dissolving fluids [19, 25]. However, retinal vessels are often just 40-120 μm in diameter, making them exceptionally tricky to target with a simple handheld tool [6]. This difficulty is compounded by the fact that undesired motion while the cannula is inserted into the vessel can tear the vessel open.

2.2 Robotic-Assisted Retinal Surgery

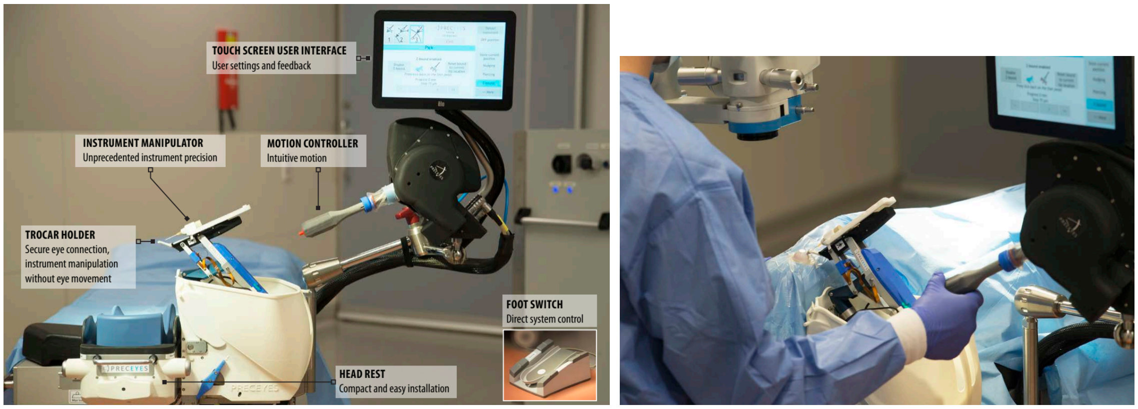
The introduction of robotic assistance to vitreoretinal surgery is intuitive: robotic arms offer a new level of precision and dexterity that human hands struggle to achieve. By far the most common robotic-assisted surgical platform is the da Vinci surgical system, which allows its user to teleoperate a robot positioned over a patient [1, 26]. The system offers improved tremor filtration, motion scaling, enhanced dexterity in small spaces, and high precision [12]. However, it is a highly generalized platform, made to accommodate a wide variety of surgeries and is not designed for microsurgery. The ophthalmic surgical environment has unique qualities, some offering new challenges, that must be considered in the development of a robotic platform:

1. The environment is unusually small and requires a precision on the order of microns [67].
2. Interaction forces between the tool and tissue are on the order of millinewtons; significantly smaller than the interaction forces in most other surgical procedures [34]. These forces can be separated into two categories: scleral and retinal interaction forces. Scleral forces are often an order of magnitude larger than retinal forces, making it difficult to sense retinal forces [44].
3. The fact that the tool punctures the sclera necessitates maintaining a remote center of motion (RCM) at the incision, around which the tool must pivot [29]. This avoids imposing a load on the sclera, which could tear it. It also avoids degrading the positioning performance of the robotic platform as the sclera deflects the tool.
4. Unlike most minimally invasive procedures, surgeons performing intraocular procedures have a direct view of the operating environment through a stereo microscope positioned directly over the lens of the patient's eye.

Several robotic platforms have been developed to better fit into vitreoretinal surgeries and improve their outcomes. By and large, they can be divided into teleoperated, cooperative, and handheld platforms.

Teleoperated surgical platforms have become widespread due in large part to the commercial success of the da Vinci surgical system. This ubiquity led to research investigating its performance in intraocular surgery. Initial evaluations found the da Vinci's precision sufficient, but encountered issues with the location of the RCM [14, 85]. Intraocular procedures require an RCM located at the sclerotomy, where the tool intersects the sclera. However, the da Vinci's RCM is located above its wrist, resulting in motions that are difficult to control. This concern led to the development of a hexapod manipulator that was mounted on the end of the da Vinci's arm [13, 58]. Although this allowed the RCM to be placed at the sclerotomy, it did not address one of the original concerns of using the da Vinci: such a large robot obscures the view that the surgeon usually has through a stereo microscope positioned over lens of the eye. The view offered by the da Vinci was not as detailed [14]. Less obstructive teleoperated platforms have been developed at UCLA [92], Columbia University [88], Vanderbilt University [97], and the University of Tokyo [86].

A teleoperated platform, the PRECEYES Surgical System [2], was used in the first successful human intraocular robotic surgery [27]. It is currently the only robotic platform cleared for retinal surgery on humans. The platform is shown in Figure 2.2. Its end-effector moves with 4 degrees of freedom (DoFs) around a remote center of motion at the sclerotomy: all 3 rotational DoFs (two tilting and one twisting) and 1 axial translation DoF [25]. Another teleoperated Robot, AcuSurgical's Luca, has recently completed its first successful clinical case [30].


TOUCH SCREEN USER INTERFACE
User settings and feedback
INSTRUMENT MANIPULATOR
Unprecedented instrument precision
MOTION CONTROLLER
Intuitive motion
TROCAR HOLDER
Secure eye connection, instrument manipulation without eye movement
FOOT SWITCH
Direct system control
HEAD REST
Compact and easy installation

(a) (b)

Figure 2.2: PRECEYES Surgical System. Photo credit [2].

Cooperative surgical platforms depend on a human to provide high-level actuation, while its control algorithm selectively compensates for unintentional smaller, high-frequency motions. The Steady Hand Eye Robot (SHER) developed at Johns Hopkins University provides this with admittance control [28]. SHER is a 5-DoF device around the RCM, lacking only the ability to twist around the end-effector's axis. Recent versions of SHER are shown in Figure 2.3. Researchers at KU Leuven have developed a similar co-manipulated device [32], which has been evaluated in a clinical feasibility study [90].

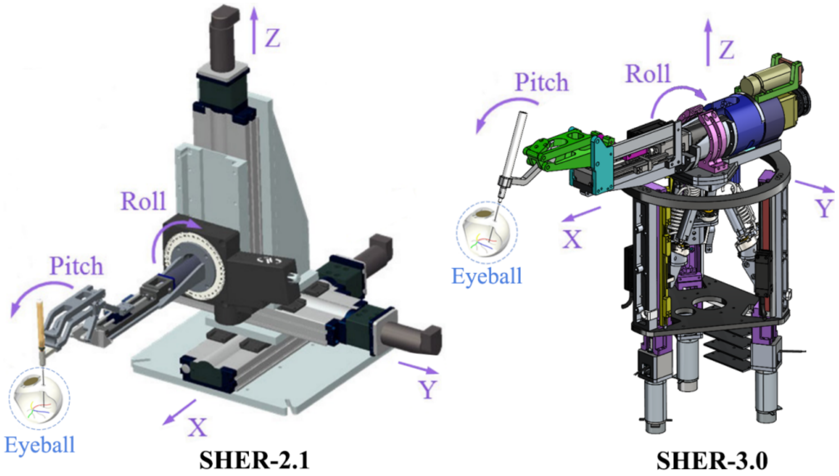
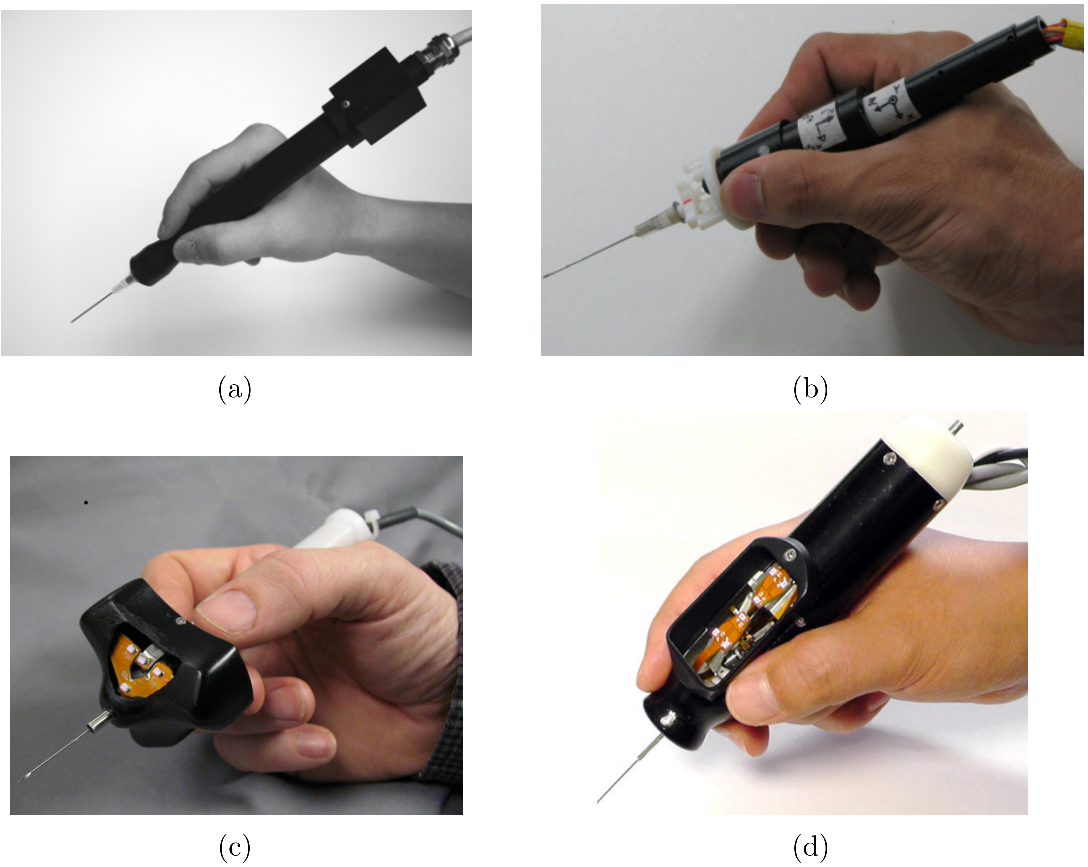


Figure 2.3: Steady Hand Eye Robot (SHER). Photo credit [98].

Handheld surgical platforms are fully held by the surgeon and must compensate for their unintentional motions before they reach the end-effector. They have minimal footprint in the operating room and are intuitive for surgeons who are used to handheld instruments. The foremost handheld platform is Micron, developed over several iterations in the Surgical Mechatronics Laboratory at Carnegie Mellon University. These iterations are described in the following section.

2.3 Micron: A Handheld Robot for Microsurgery

Micron is a handheld tool that contains a parallel manipulator between its handle and its end-effector. This parallel manipulator allows it to actively cancel unintentional or unsafe motions in the handle before they reach the tooltip. The various iterations of Micron are shown in Figure 2.4.



(a)

(b)

(c)

(d)

Figure 2.4: Iterations of Micron and related robots, as described in [5], [47], [53], and [94], respectively.

Micron was first introduced in 2003, as shown in Figure 2.4a [65]. It used 3 actuators, each with seven piezoelectric stacks, to move the end-effector with 3 degrees of freedom: both tilting DoFs and an axial translation DoF. The pose of the handle was tracked with a 6-DoF inertial sensor affixed to its proximal end. The 3-D position of its tip was tracked by an optical sensor that illuminated the workspace with pulsed infrared light and used stereo position-sensitive detectors (PSDs) to sense the reflection of the light off of a small ball placed on the tip of the manipulator. This sensor, the Apparatus to Sense Accuracy of Position (ASAP), was first developed in [40] and became integral to later versions of Micron.

The large number of piezoelectric stacks in Micron's first iteration made it unusually heavy for a retinal surgical tool. A second iteration was created to reduce the weight. It used just 1 piezoelectric stack per actuator, mechanically amplifying each actuator to achieve a sufficient workspace [5]. However, traditional fabrication techniques proved insufficient for the tiny, complex parts in the manipulator, leading to unsatisfactory performance. A third iteration replaced these parts with a system of flexures to remove error due to friction, backlash, and imperfect assemblies [22]. Again, imprecise mechanical fabrication created problems for the manipulator, reducing its workspace. Building on this version of Micron, a more compact robot, ITrem, was developed [47]. Shown in Figure 2.4b, it significantly reduces the size of the inertial sensor on the handle to make the manipulator easier to use in surgery.

Further research found that involuntary tremor often exists below 10 Hz, making it necessary to sense and suppress motions at that frequency [53]. Motion at such frequencies is too low for the inertial sensor used in earlier iterations of Micron, so a fourth iteration was created to utilize the ASAP sensor instead [53]. This manipulator, shown in Figure 2.4c, still moves with 3 degrees of freedom, but has an array of 3 infrared LEDs mounted at the base of the end-effector, as well as a single LED on the handle. Each of these LEDs pulses at a distinct frequency so that they can be differentiated by ASAP. When they are all in ASAP's field of view, the 6-DoF pose of both the handle and the end-effector can be tracked within less than 10 μm. The base of the manipulator was also made wider to enable a larger workspace. These changes led to significant improvements in handheld accuracy.

Every iteration of Micron up to this point moved with 3 degrees of freedom, which does not allow the tool to implement a remote center of motion. Additionally, the workspace of the tooltip was limited to a few hundred microns. To remedy these issues, the fifth iteration of Micron was created, shown in Figure 2.4d. This manipulator was the first 6-DoF iteration of Micron, designed to house a Stewart-Gough parallel platform [94]. The Stewart-Gough platform was chosen for its inherent stiffness and ability to tilt significantly, while maintaining a small footprint. Each leg of the platform is a tiny linear piezoelectric SQUIGGLE motor from New Scale Technologies that lengthens or shortens the leg. Combined with flexures on each end that connect to the top and base platforms, they move the top platform in 6 degrees of freedom. Because the end-effector could now move in 6 DoF, three infrared LEDs needed to be mounted on both the handle and end-effector to sense each part's 6-DoF pose. This iteration of Micron enlarged the workspace of the manipulator by an order of magnitude, enabled it to maintain an RCM, and provided a statistically significant reduction in 3D positioning error [93].

2.4 The KIST Microsurgical Robot

A descendant of the 6-DoF Micron has been developed in the ANSUR Laboratory at the Korean Institute of Science and Technology (KIST) [46]. Its mechanical design, show in Figure 2.5, forms the basis of the research presented in this thesis. Although the manipulator is still a 6-DoF parallel platform, it does not use a Stewart-Gough platform. Instead, it rearranges the joints to create a 6-Prismatic-Universal-Spherical (6-PUS) system. Each leg consists of a prismatic joint that pushes up or down on a universal joint. Each universal joint enables a fixed-length strut to tilt on two axes. This strut connects the universal joint to a spherical joint, which in turn connects to the top platform.

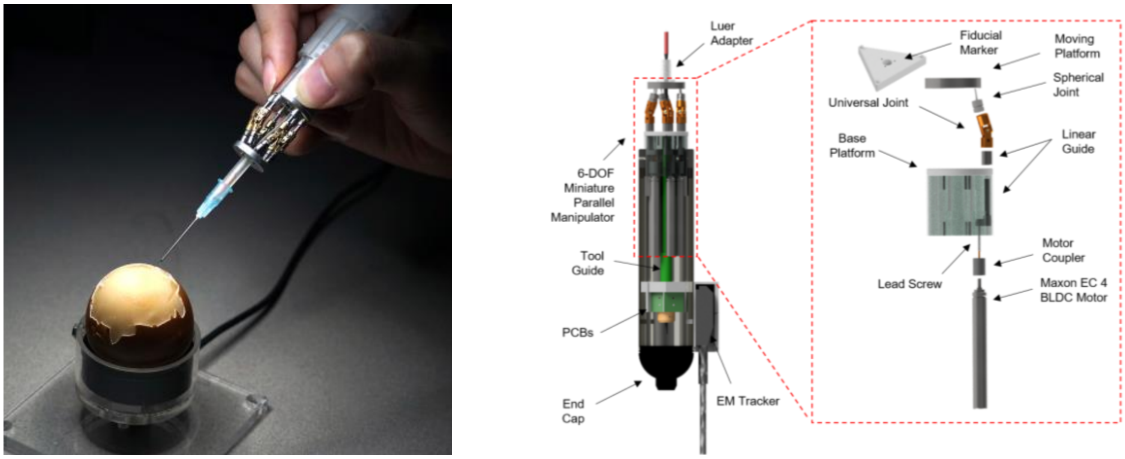


Figure 2.5: The KIST microsurgical robot. Photo credit [46].

The 6-PUS configuration of the manipulator allows it to keep its motors parallel so that they can be larger and still fit inside the handle. This is a crucial change because it enables the use of stiffer motors; specifically, EC4-431182 Maxon motors [46]. The change in joints also removes the small flexures the 6-DoF Micron used.

These flexures were prone to breaking when sudden forces were applied and reduced the stiffness of the manipulator. With these changes, the KIST microsurgical robot is measured to be resistant to side load up to 5.0 N applied at an RCM, an order of magnitude more than with the 6-DoF Micron [46].

To sense its 6-DoF pose, the KIST manipulator includes an electromagnetic (EM) sensor: the Viper FT-Flatsided sensor from Polhemus. It is added to the back end of the manipulator so that it tracks the pose of the handle, but not the end-effector. Specifically, the sensor tracks its pose relative to an EM source placed nearby. When the tracker was within 150 mm of the source, it tracked position with a root-mean-square error (RMSE) below 10 μm [46]. At a distance of 250 mm from the source, it maintained an RMSE below 20 μm.

2.4.1 Inverse Kinematics of a 6-PUS Parallel Manipulator

The inverse kinematics of a 6-Prismatic-Universal-Spherical (6-PUS) parallel manipulator define how the linear motion of its six linear actuators can work in concert to move the tool of the manipulator with six degrees of freedom. The inverse kinematics of the KIST microsurgical robot are derived in [46]. Since they are used heavily in this thesis, they are repeated below.

The kinematic configuration of the 6-PUS parallel manipulator is shown in Figure 2.6. The base of the manipulator, assumed to be stationary, is shown in green. Six prismatic joints are located around the base, from which the red linear rods extend. These linear rods connect to the orange connecting rods (struts) with universal joints. Unlike the linear rods, the struts do not change length. The struts attach to the top platform, in blue, with spherical joints. At the center of the top platform, a needle is attached, orthogonal to the plane of the top platform. The tool tip is located at the end of the needle. At some variable location along the needle, a remote center of motion is defined. The needle pivots around this center.

The origin of all measurements in Figure 2.6 is set at O. The remote center of motion, T_{rcm}, can be assumed to be given with respect to the top platform. We can also assume that we know the dimensions of the base platform and the top platform, as well as the constant lengths of the struts. With that information, an input Cartesian location for T_{tip} with respect to O can be translated to the z-axis

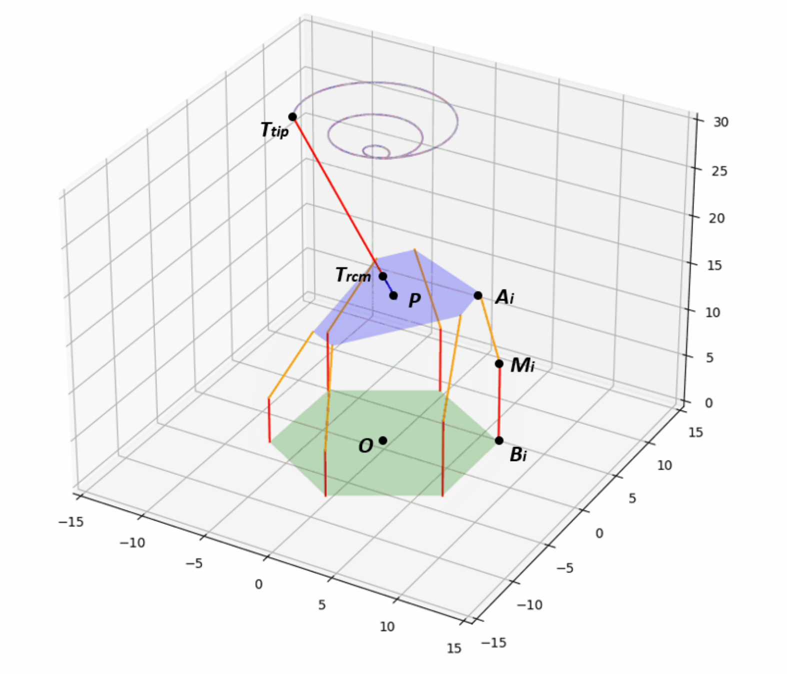


Figure 2.6: The kinematic configuration of the 6-PUS parallel manipulator.

displacement of each of the linear actuator rods, OM_i^z, for $i = 1, 2, ..., 6$.

The base platform is a regular hexagon, so the position of each prismatic joint can be described as follows, given r_b, the radius of the circle bounding the platform:

$$OB_i = \begin{bmatrix} r_b \cos\theta_i^b & r_b \sin\theta_i^b & 0 \end{bmatrix}^T, \ \theta_i^b = -\frac{\pi}{6} + (i-1)\frac{\pi}{3}$$

The position of each spherical joint around the top platform can be described as follows, given r_p, the radius of the circle bounding the platform, and P_0, the resting position of the center of the top platform:

$$P_0A_i = \begin{bmatrix} r_p \cos\theta_i^p & r_p \sin\theta_i^p & 0 \end{bmatrix}^T$$

$$\theta^p = 5\pi/3 + 0.82, \pi/3 - 0.82, \pi/3 + 0.82, \pi - 0.82, \pi + 0.82, 5\pi/3 - 0.82$$

The first step in finding the linear rod lengths is to find the positions of the spherical joints, given nonstandard tool tip and remote center of motion positions. We can first assume that the resting position of the top platform, P_0 leaves the tool vertical, with T_{rcm}, T_{tip}, and P_0 only displaced along the z-axis. In that case, the displacement between the current tool tip and the resting position of the top platform can be written as follows, given l_t, the length of the tool:

$$P_0T_{tip} = \begin{bmatrix} \Delta T_{tip}^{x} & \Delta T_{rcm}^{y} & \Delta T_{tip}^{z} + l_t \end{bmatrix}^T$$

We can also define the orientation of the tool:

$$v_{tool} = \frac{T_{tip} - T_{rcm}}{||T_{tip} - T_{rcm}||}$$

With these definitions, the current center of the top platform, OP, can be defined as follows:

$$OP = OP_0 + P_0T_{tip} - l_t v_{tool}$$

Now that we have the current center of the top platform, we need a rotation matrix to define the orientation along which to offset the positions of the spherical joints, P_0A_i. This rotation matrix is the rotation between the current tool vector, v_{tool}, and the tool vector when the platform is in resting position at P_0, v_{tool}^{init}. The axis of rotation, v_{rot}, and the angle of rotation, θ_{rot}, can be written as follows:

$$v_{rot} = \frac{v_{tool} \times v_{tool}^{init}}{||v_{tool} \times v_{tool}^{init}||}, \; \theta_{rot} = \arccos\left(\frac{v_{tool} \times v_{tool}^{init}}{||v_{tool} \times v_{tool}^{init}||}\right)$$

We can construct the unit quaternion from these values:

$$q = \begin{bmatrix} \cos\left(\frac{\theta_{rot}}{2}\right) \mid \cos\left(\frac{\theta_{rot}}{2}\right) v_{rot} \end{bmatrix}$$

Finally, the rotation matrix, R, can be constructed as the matrix in Figure 2.7, assuming $q = \begin{bmatrix} q_r & q_x & q_y & q_z \end{bmatrix}$.

$$\mathbf{R} = \begin{bmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_z q_r) & 2(q_x q_z + q_y q_r) \\ 2(q_x q_y + q_z q_r) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_x q_r) \\ 2(q_x q_z - q_y q_r) & 2(q_y q_z + q_x q_r) & 1 - 2(q_x^2 + q_y^2) \end{bmatrix}$$

Figure 2.7: Rotation matrix.

We can now define the current positions of the spherical joints:

$$OA_i = RP_0A_i + OP$$

Now that we have the current positions of the spherical joints, we know the z-axis displacement of each spherical joint. In order to determine the z-axis displacement caused by the linear rods, we simply need to subtract the component of that displacement that is caused by the struts. We can find the z-axis displacement caused by the struts with a simple application of the Pythagorean Theorem. The vertical displacement of each linear rod can found as follows:

$$OM_i^z = OA_i^z - \sqrt{||M_iA_i||^2 - \left\| \begin{bmatrix} OA_i^x & OA_i^y \end{bmatrix} - \begin{bmatrix} OB_i^x & OB_i^y \end{bmatrix} \right\|^2}$$

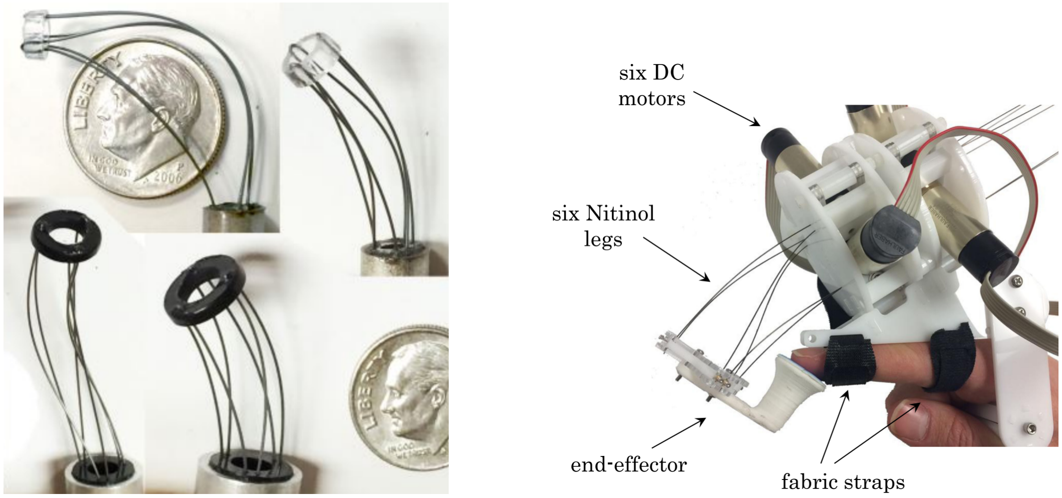
Chapter 3

Investigating the Applicability of Parallel Continuum Manipulators

A significant limitation of prior iterations of Micron has been the complexity of their mechanical parts. Since the device is handheld and contains six legs in a parallel configuration, its 12 links and 18 joints are necessarily tiny. Machining such complex and tiny parts to a precision that allows the joints to function is both costly and challenging. Maintaining those parts over years of use is prohibitively difficult. The costs in time and money that this complexity imposes motivated us to look for a simpler mechanical structure.

Parallel continuum manipulators offer a similar workspace to parallel rigid-link manipulators, but they come with much simpler mechanical structures. In lieu of complicated universal or spherical joints that connect rigid links, they use wire to connect the base platform to the top platform. Using wire would be much cheaper than using precision machined rigid links. Unfortunately, the simplicity that is gained in the mechanical design is lost in the model of the manipulator's behavior. Deformable wires include a natural elasticity and compliance that can be hard to predict and can vary with any load applied to the end-effector. In recent years, the kinematics and statics of a parallel continuum manipulator very similar to that which might exist in Micron have been determined [83]. That procedure enabled us to further investigate the applicability of a parallel continuum manipulator to Micron.

The structure of parallel continuum robots makes them well-suited for miniature manipulators [18] and several such manipulators have been created. Much of the work on parallel continuum manipulators comes from Dr. Caleb Rucker's REACH Laboratory at the University of Tennessee, Knoxville, which has designed manipulators with platform diameters as small as 10 mm [61]. A similarly small device has been created at the University of Pennsylvania to allow for haptic feedback to fingertip motions [96]. These manipulators are shown in Figure 3.1. Both use six linear actuators to increase and decrease the length of each leg of wire between the top and base platforms.



(a) (b)

Figure 3.1: Miniaturized parallel continuum manipulators. Photo credit [83, 96].

These examples and the body of research they are built on provided the inspiration for this investigation of whether Micron could benefit from the structure of a parallel continuum manipulator.

3.1 Design Requirements

Three fundamental hurdles need to be cleared to prove that a parallel continuum manipulator could be used inside Micron:

1. The manipulator must be sufficiently miniaturizable to fit inside a handheld instrument.

2. The manipulator must have a workspace comparable to or larger than previous iterations of Micron.
3. The manipulator must be sufficiently resistant to the side load placed on the end-effector's RCM by the sclera.

The first and second hurdles are demonstrably cleared by prior research into small parallel continuum manipulators. Not only have miniature manipulators been constructed, but the manipulator with a diameter of just 10 mm has been shown to have a workspace much larger than Micron's [10]. If the workspace of such a manipulator were changed, it would still need to have a maximum horizontal radius of 4 mm to keep it in line with the KIST microsurgical robot [46]. However, the third hurdle is a significant cause for concern. Suppose Micron's end-effector were inserted into the eye and the sclera were to move, placing a transverse load on the end-effector at its RCM. An insufficiently rigid manipulator would allow the end-effector to move, potentially causing uncontrolled deflection of the tip inside the eye. Additionally, imposing a sudden side load could cause the manipulator to exhibit wobbly dynamics, which should be minimized. To combat these effects, we require that the manipulator must be able to resist a typical scleral side load applied at the RCM of its end-effector. Side loads imposed by the sclera have been measured around 250 mN, so that is the side load we simulate [44].

3.2 Simulation of a Parallel Continuum Manipulator Using MATLAB Simscape

An attempt was made to create a kinematics and dynamics model of a parallel continuum manipulator in MATLAB Simscape. The simulation was motivated by our desire for a model whose legs we could control and whose end-effector we could physically perturb to view its reaction. Such a model would not only inform us of the deflection of the end-effector with side load, but also of any wobbly dynamics that might immediately follow that side load. In this section, we develop a dynamics model of a standard parallel continuum manipulator and use its inverse kinematics to control the movement of its top platform.

3.2.1 Structure of the Simulated Manipulator

To verify the accuracy of the model, its dimensions were kept similar to those described in [18]. This model was therefore viewed as a precursor to a model small enough to fit inside Micron. The mechanical parameters are listed in Table 3.1.

Table 3.1: Parameters of the Simulated Structure

Initial Rod Length	0.4 m
Rod Radius	3 mm
Platform Radius	80 mm
Platform Major Angle	100 deg
Platform Minor Angle	20 deg
Rod Density	7990 kg/m^3
Rod Young's Modulus	193 kPa
Rod Shear Modulus	77.2 kPa

Images of the manipulator, drawn from the results of the inverse kinematics algorithm, are shown in Figure 3.2. The initial rod length is the length of each rod when the manipulator is in the equilibrium position. The platform radius is the distance from the center of each platform to the point where it connects to each rod. The platform major and minor angles are the angles measured between the points where each rod intersects the platforms. The manipulator moves its top platform with 6 degrees of freedom by lengthening and shortening each of its six elastic rods with feed-through prismatic joints at the point where each rod intersects the base platform.

3.2.2 Inverse Kinematics

The kinematics of a parallel continuum manipulator are necessarily load-dependent, meaning that one must solve for the statics at the same time. If we ignore all forces caused by sources other than our actuators, another solution such as constant-curvature kinematics can be applied [81]. However, the interactive nature of the retinal surgical environment requires that we take external load into account. We chose the method in [83] as an efficient and effective kinetostatic analysis. It uses Cosserat rod theory to model the deformations of the manipulator's legs.

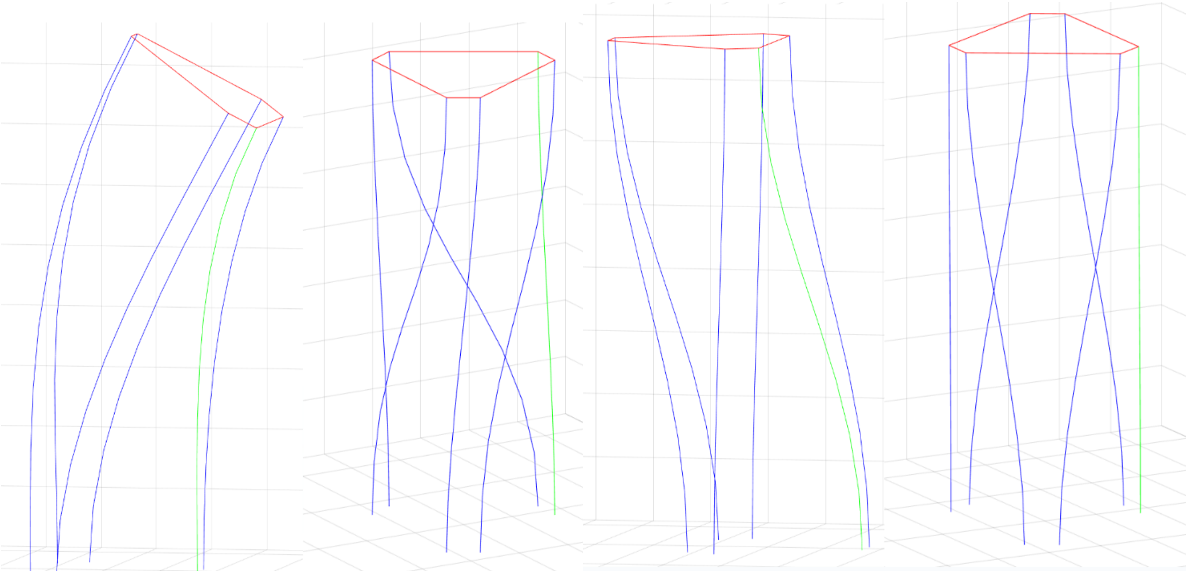


Figure 3.2: Results of the inverse kinematics algorithm. Running left to right, the manipulator is shown tilting, twisting, translating, and at equilibrium.

3.2.2.1 Cosserat Rod Model

The Cosserat rod equations are given in Equations 3.1-3.6.

$$p_i' = R_i v_i \tag{3.1}$$

$$R_i' = R_i \hat{u}_i \tag{3.2}$$

$$v_i = v_i^* + K_{se,i}^{-1} R_i^T n_i \tag{3.3}$$

$$u_i = u_i^* + K_{bt,i}^{-1} R_i^T m_i \tag{3.4}$$

$$n_i' = -f_i \tag{3.5}$$

$$m_i' = -p_i \times n_i - l_i \tag{3.6}$$

Each rod i, for $i = 1...6$, can be defined by its world frame position $p_i(s_i)$ and word frame orientation $R_i(s_i)$ as a function of arc length s_i. Equations 3.1 and 3.2 express the rate of change of world frame position and orientation with respect to the body frame rate of change of position $v_i(s_i)$ and the body frame rate of change of rotation $u_i(s_i)$.

These body frame velocities are obtained from Equations 3.3 and 3.4. The v_i^* and u_i^* vectors represent the rod in an initially straight stress-free state. They are assigned as $v_i^* = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ and $u_i^* = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. The $K_{se,i}$ and $K_{bt,i}$ matrices relate to the stiffness of the rod cross-section. They are defined as $K_{se,i} = diag(A_iG_i, A_iG_i, A_iE_i)$ and $K_{bt,i} = diag(E_iI_i, E_iI_i, J_iG_i)$, where A_i is the area of the cross-section, E_i is the Young's modulus of the rod, G_i is the Shear modulus of the rod, I_i is the second area moment about the local x and y axes, and J_i is the polar area moment about the local z axis.

The vectors $n_i(s_i)$ and $m_i(s_i)$ are the internal force and moment vectors of each rod as a function of arc length. They are obtained in Equations 3.5 and 3.6. The vectors f_i and l_i are the distributed force applied to each unit length of the rod [18].

3.2.2.2 Boundary Conditions

Boundary conditions can be written to solve for different parameters within the Cosserat rod equations. For example, [83] derives boundary conditions for the inverse kinematics of a manipulator with no external load applied. They are given in Equations 3.7-3.10. The conditions assume that at both ends, each rod can rotate about its z axis, rendering the angle of that rotation unimportant. The F and M vectors represent external forces and moments applied to the center of the top platform. They are assumed to be negligible for our purposes. The value L_i is the length of each rod. The desired position and orientation of the center of the top platform are denoted by p_d and R_d, respectively.

$$\sum_{i=1}^{6}[n_i(L_i)] - F = 0 \tag{3.7}$$

$$\sum_{i=1}^{6}[p_i(L_i) \times n_i(L_i) + m_i(L_i)] - p_d \times F - M = 0 \tag{3.8}$$

$$(log(R_i^T(L_i)R_d))^{\vee}|_{xy} = 0 \; for \; i = 1...6 \tag{3.9}$$

$$p_d + R_d r_i - p_i = 0 \; for \; i = 1...6 \tag{3.10}$$

3.2.2.3 Optimization

The combination of the Cosserat rod equations and the boundary conditions presents a boundary value problem that can be solved by optimizing the free variables to fit the boundary conditions. In the inverse kinematics developed in [83], the rod lengths are optimized to get the end-effector pose closest to a desired pose. Starting with guessed boundary conditions, one can integrate along the arc length of each rod using the fourth-order Runge-Kutta method. Once the end-effector position is determined, the Levenberg-Marquardt algorithm is used to update the boundary conditions [48]. Example poses of the manipulator, determined with this inverse kinematics algorithm, are shown in Figure 3.2.

3.2.3 Dynamics Simulation

Our dynamics simulation uses the lumped-parameter model, a simple way to model a flexible body in Matlab Simscape Multibody [56]. Although the Simscape package includes blocks for flexible cylindrical beams, they were not used because they cannot handle large deflections where Hooke's Law no longer applies. Instead, each rod is modeled as a mass-spring-damper system with several rigid cylinders connected by 5 joints. Each joint has 4 degrees of freedom: three rotational and one translational for strain. A picture of the model at its equilibrium pose is shown in Figure 3.3. A small needle is mounted on top of the platform. Prismatic joints at the base of each rod serve to lengthen or shrink the rods.

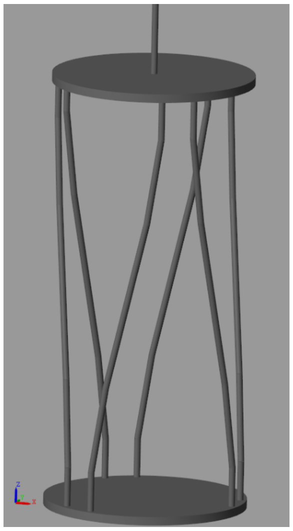


Figure 3.3: The dynamics model at equilibrium.

Since the rods are modeled as rigid links with joints, the local spring constants and damping coefficients of each joint must be established. Characterization of these constants was performed by applying a transverse load step of 20 N to the tip of a rod and measuring displacement. The same load step was applied to a ”Flexible Cylindrical Beam” block that was given all the relevant material and structural properties from Table 3.1. Comparison of the displacement of both rods over the same load step enabled characterization of the spring constants and damping coefficients along the rod’s local x and y axes. The spring constant was first altered until the steady-state displacement of the two rods matched. Then, the damping coefficient was altered. The peaks of the displacement oscillations were determined and the period between them measured. An exponential decay curve was fit to the envelope of the displacement oscillations using MATLAB’s ”fitnlm” function. The damping coefficient that produced a period within 0.0001 Hz of the desired period and had the smallest Euclidean distance between the parameters of the exponential decay curves was chosen.

3.2.4 Control Algorithm

PID control is used to drive the manipulator to desired end-effector poses. Since the manipulator mechanism is lightweight and small-scale, the manipulator dynamics can largely be ignored in the control scheme. Therefore, we focus on the inverse kinematics algorithm to derive the control signals. The full control diagram is shown in Figure 3.4.

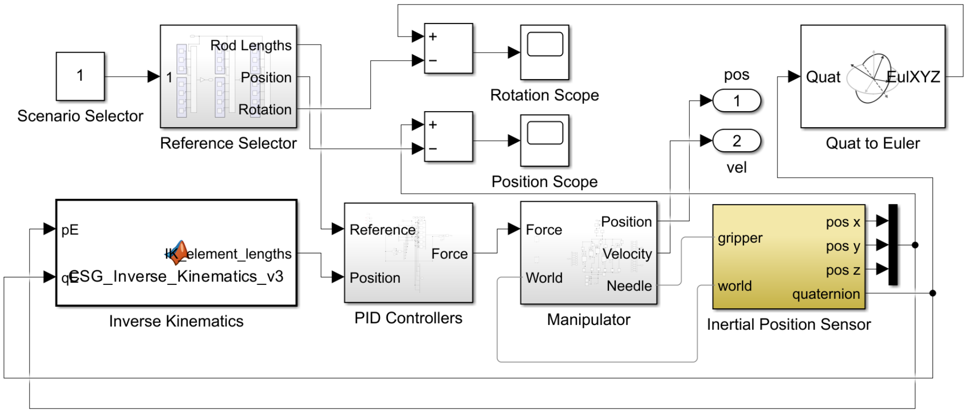


Figure 3.4: The Simulink control diagram of the dynamics model.

A desired pose is run through the inverse kinematics algorithm and the desired length of each rod is obtained. At the same time, the actual pose of the end-effector is measured. This measurement is realistic when a pose estimation device is used, as in [52]. The measured pose is run through the inverse kinematics algorithm at 200 Hz and the measured length is each rod is obtained. The error between each rod's desired length and measured length is input into 6 independent PI controllers. The controllers all have the same gain. Each controller outputs a force, which is applied to the prismatic joint at the base of each rod. These prismatic joints lengthen and shorten the rods to their desired lengths.

3.2.5 Results

The manipulator can be controlled near desired positions with significant steady-state error. For our experiment, we chose a number of references consisting of tilts around the x-axis and translations along the y-axis. The desired positions are shown in Table 3.2. Figure 3.5 presents the evolution of a simulation with the seventh reference position.

Table 3.2: Desired End-Effector Positions

Scenario	Position (m)			Rotation $(°)$		
	x	y	z	x	y	z
1	0.0	0.0	0.4	0.0	0.0	0.0
2	0.0	0.0	0.4	5.0	0.0	0.0
3	0.0	0.0	0.4	7.0	0.0	0.0
4	0.0	0.0	0.4	9.0	0.0	0.0
5	0.0	0.0	0.4	11.0	0.0	0.0
6	0.0	0.0	0.4	13.0	0.0	0.0
7	0.0	0.0	0.4	15.0	0.0	0.0
8	0.0	0.01	0.4	0.0	0.0	0.0
9	0.0	0.03	0.4	0.0	0.0	0.0
10	0.0	0.05	0.4	0.0	0.0	0.0

Figure 3.6 characterizes the result using the rod length error, rotation error, and translation error. Since the manipulator is initialized to the same end effector position as the reference, the positional error stays close to zero. Steady state error exists for

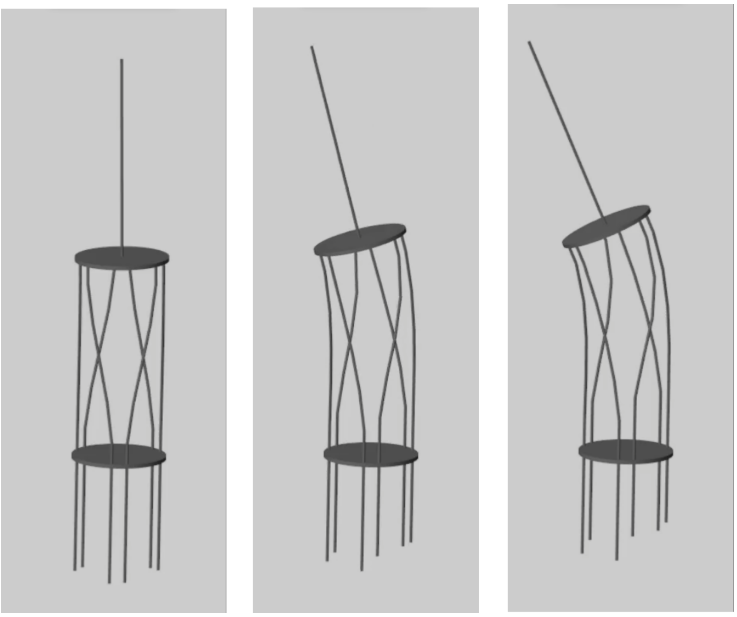
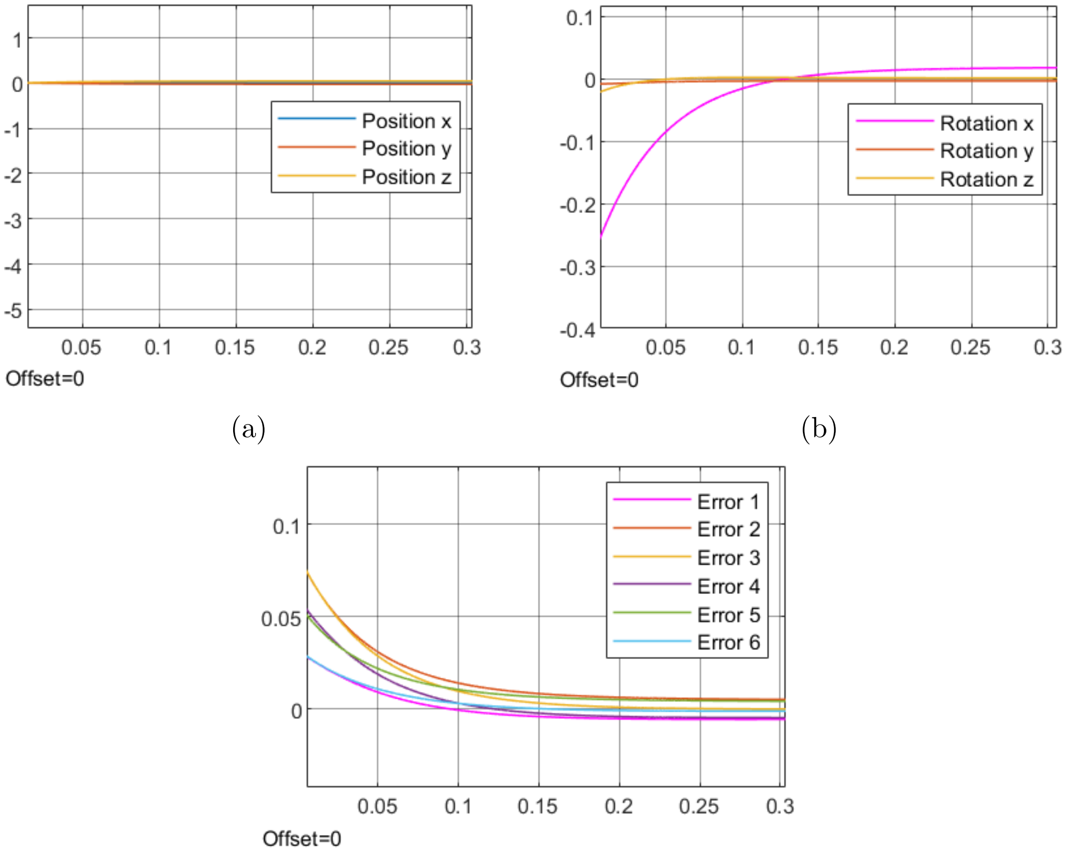


Figure 3.5: Overall mechanism movements during the simulation evolution at $t = 0s, t = 0.1s, t = 0.3s$.

rotational error and rod length error. This error can be attributed to model error. All reference poses from Table 3.2 converged with steady-state error.

The steady-state errors in end-effector pose for all reference poses in Table 3.2 are shown in Figure 3.7. When the manipulator is tilting around the x-axis, position error along the y-axis increases; all other error is negligible. Orientation error stays around zero, indicating that the desired tilt angle was achieved in all cases.

When the manipulator is translating along the y-axis, position error along the y-axis increases dramatically. In fact, the error is almost as large as the desired translation, suggesting that the translation was largely unsuccessful. Orientation error remains close to zero for all translations.

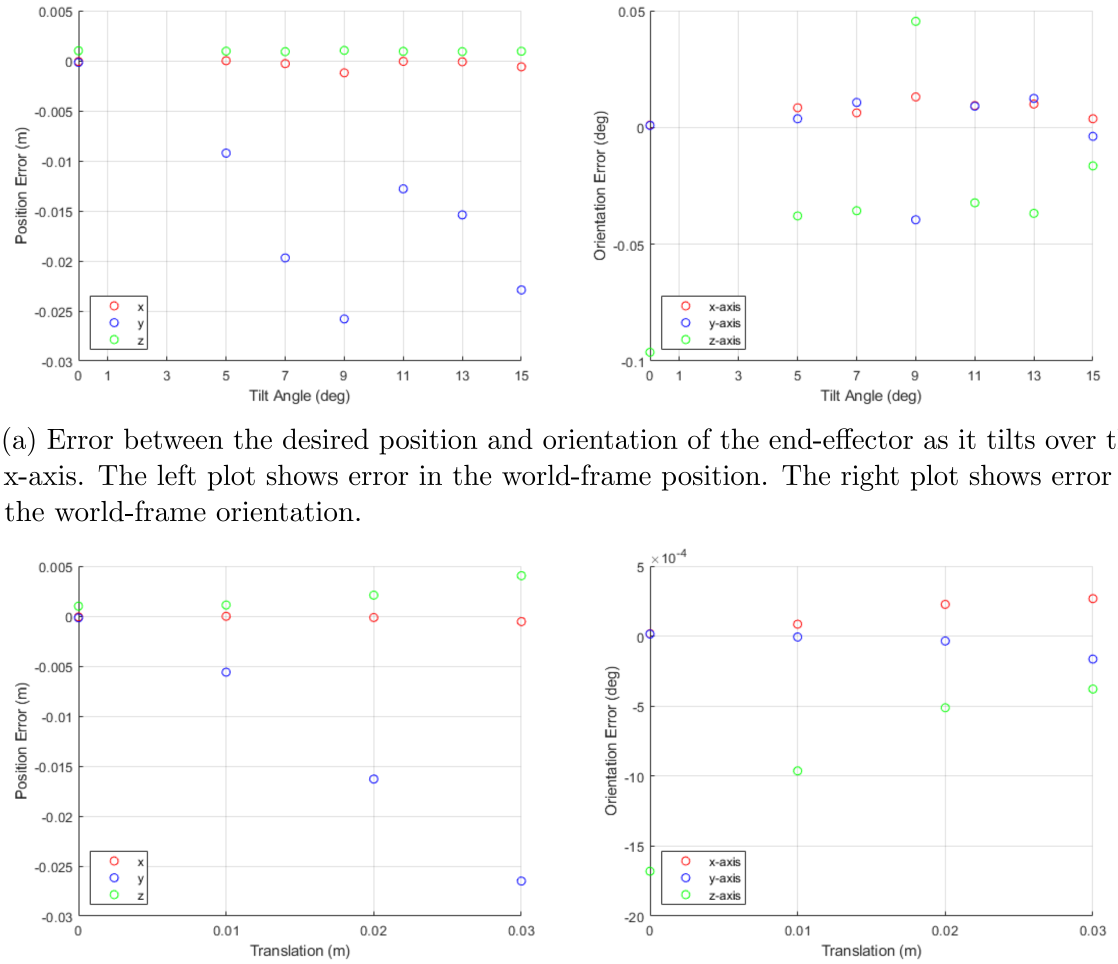


(a)

(b)

(c)

Figure 3.6: Error in the simulation evolution at $t = 0s, t = 0.1s, t = 0.3s$. (a) Error in the position of the top platform. (b) Error in the orientation of the top platform. (c) Error in the positions of the linear actuators.



(a) Error between the desired position and orientation of the end-effector as it tilts over the x-axis. The left plot shows error in the world-frame position. The right plot shows error in the world-frame orientation.

(b) Error between the desired position and orientation of the end-effector as it translates along the y-axis. The left plot shows error in the world-frame position in meters over translation in meters. The right plot shows error in the world-frame orientation in radians over translation in meters.

Figure 3.7: Position and orientation error of the top platform of the dynamic simulation in a variety of poses.

3.2.6 Discussion

Model error as a whole is the main culprit for the steady-state error between desired and measured positions. We implemented the lumped-parameter model for our dynamics simulation using 5 elements to make up each of the 6 manipulator legs. While more elements per leg would result in a more accurate result, this would also increase the computational and time cost of running the dynamics simulation (the maximum we tested was 10 elements per limb). A more robust method of approximating a flexible body is described in the aforementioned Mathworks resource and is implemented by combining a rigid-body model and a deformation model.

Error in the dynamics model is also attributable to the method we used to lengthen and shorten the rods. The structure of the manipulator assumes that feed-through prismatic joints are used to change the lengths of the rods. However, our Simscape model required both that the bases of the rods be fixed and that a fixed number of joints be used for each rod. To best simulate the manipulator's behavior, we placed prismatic joints at the base of each rod and controlled their length. This method does not allow the added rod length to have an orientation other than vertical. It also does not allow the rods to shorten past a certain threshold. A far more detailed and complex dynamics model for a parallel continuum manipulator is presented in [84].

We implemented an inverse kinematics algorithm similar to [83]. From our results, we can observe that our inverse kinematics algorithm reached a solution in an average of 7.8 optimization steps. Compared to prior work, this is a greater number than that of [83], which took an average of 5.5 steps. This algorithm was optimized for manipulator that was 100 times taller than ours, leading to the inefficiency. Additionally, our algorithm was run at just 200 Hz, rather than the kHz ranges in [83], meaning each guess was more stale when it was used to warm-start the next iteration of the algorithm.

The significant model error we witnessed in this simulation, along with the long computation time that each step took, led us to decide against moving forward with this simulation method.

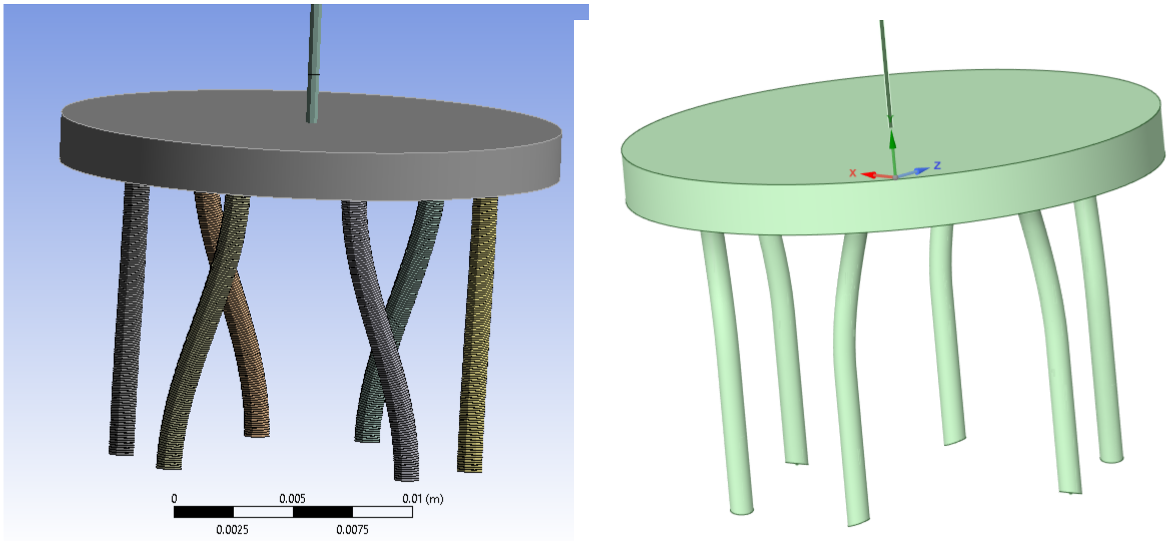
3.3 Simulation of a Stiffer Parallel Continuum Manipulator in Ansys

In parallel to the Simscape simulation, a much shorter and more rigid parallel continuum manipulator was simulated with the Static Structural Analysis feature of Ansys. The purpose of this simulation was more limited in scope: it was made to evaluate the forces required to both actuate and perturb a stiffer parallel continuum manipulator. A stiffer manipulator was chosen to reflect the fact that Micron requires more resistance to side load and a smaller footprint than any of the parallel continuum manipulators developed in prior research. Inability to achieve the required workspace and resistance to side load on such a manipulator would indicate that parallel continuum manipulators were ill-suited for use in Micron.

3.3.1 Structure of the Simulated Manipulator

Two different structures were examined. Both were primarily informed by the structure of the KIST microsurgical robot, whose parts we hoped to adapt. This meant locking in the diameter, major angle, and minor angle of both the top and base platforms to the specifications of that design. Consequently, the only two parameters left variable were the height of the top platform above the base platform and the diameter of the wires. All the wires were modeled as Stainless Steel 316L, to facilitate easy sterilization. Superelastic nitinol was briefly explored, but exhibited few differences with such short wires. Finally, the effect of gravity was disabled in the simulation.

Examples of the two structures are shown in Figure 3.8. The first structure is informed heavily by Cosserat rod theory. Given a height of the top platform and a wire thickness, the inverse kinematics routine from the MATLAB Simscape simulation is used to generate the positions of points along the length of each wire while the manipulator is in its equilibrium position. The 3D points on each leg are then projected as 2D points onto the plane they collectively define. The normal vector of each of the six planes is recorded. A Python script written for use in Ansys SpaceClaim begins with a standard top platform and end-effector, creates the six planes, and draws all of the points onto them. The script then uses the points to define a contour for each leg and extrudes a cylinder along each of the contours. The second structure does the same, except that it offsets the angles that the wires are inserted into the top platform by 5.9 degrees from the vertical, the same as is specified in the KIST microsurgical robot's top platform.



(a)

(b)

Figure 3.8: Examples of the simulated structures. (a) The normal wire entry model. (b) The angled wire entry model.

Each structure includes an end-effector in the form of a thin needle. The needle is 60 mm long and has an RCM 33 mm from its tip, as defined in [46].

Due to the time-intensive nature of the simulations, just 16 different variations of each structure were evaluated. For each structure, height above the base platform was selected from $[5, 7, 9, 12]$ mm and wire diameter was selected from $[0.5, 0.70.9, 1.1]$ mm.

3.3.2 Static Structural Analysis

Ansys Mechanical uses finite element methods to evaluate the static behavior of a structure under load. The simulation explored two separate load scenarios, with the limited nature of the simulation environment restricting it to tilting motions only.

The first scenario investigates the limits of the achievable workspace. This workspace is first constrained by the maximum linear travel of the prismatic joints, 5 mm, and the maximum force that can be applied by the prismatic force, 7.5 N, as defined in [46]. The bases of two adjacent legs are kept stationary, while the other four are allowed to move. A 5 mm upward displacement is induced in each of the two free legs over the course of 100 linear steps. At each step, the force applied to the base of each of the six legs is measured. Any timesteps that have a leg's reaction force going beyond 7.5 N are thrown out. The horizontal deflection of the end-effector tip is also measured at every timestep. The maximum tip deflection that is observed within the force and displacement constraints of the actuators is recorded. An example of this tilt is shown in Figure 3.9.

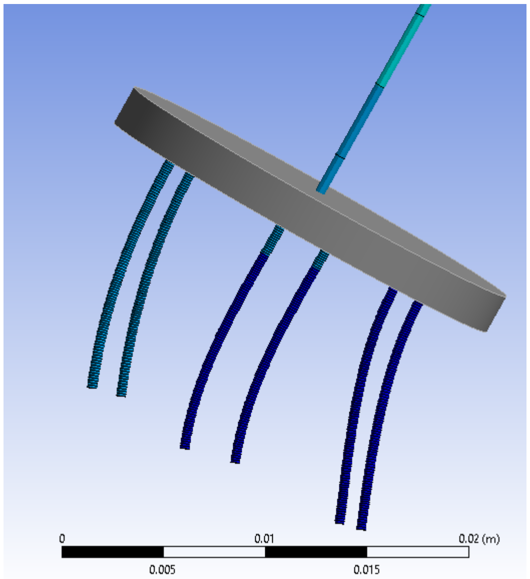


Figure 3.9: Example of the simulated structure tilting.

The second scenario examines the susceptibility of the manipulator to tip deflection under side load applied at the RCM. It fixes the positions of all the legs, applies a 0.25 N side load to the RCM over 100 linear steps, and again measures the horizontal deflection of the tooltip. The deflection of the tooltip at 0.25 N is recorded.

In order to make the optimization tractable, our simulation sets the top platform as a solid structure, but characterizes the wires as Ansys Beams. The mesh of the wires is set to have an element length of 0.1 mm. The simulations are all run with the ”Large Deflection” setting enabled, which forces the solver to model beams without using Hooke's Law. To facilitate execution of the 32 simulations, a Python script was created for use inside Ansys Mechanical. It runs through the SpaceClaim geometries, loading each one, setting meshes, assigning remote points and joint probes, running the simulations, and recording the probed values in a CSV file.

3.3.3 Results

3.3.3.1 Structure with Normal Wire Entry

The first structure examined was that in which all the wires enter both the top and base platforms at an angle normal to those platforms. The maximum achievable horizontal deflection of the end-effector's tooltip, under the force and position constraints of Micron's actuators, is shown in Figure 3.10. The horizontal deflection of that tooltip under a 0.25 N side load applied at the RCM is shown in Figure 3.11.

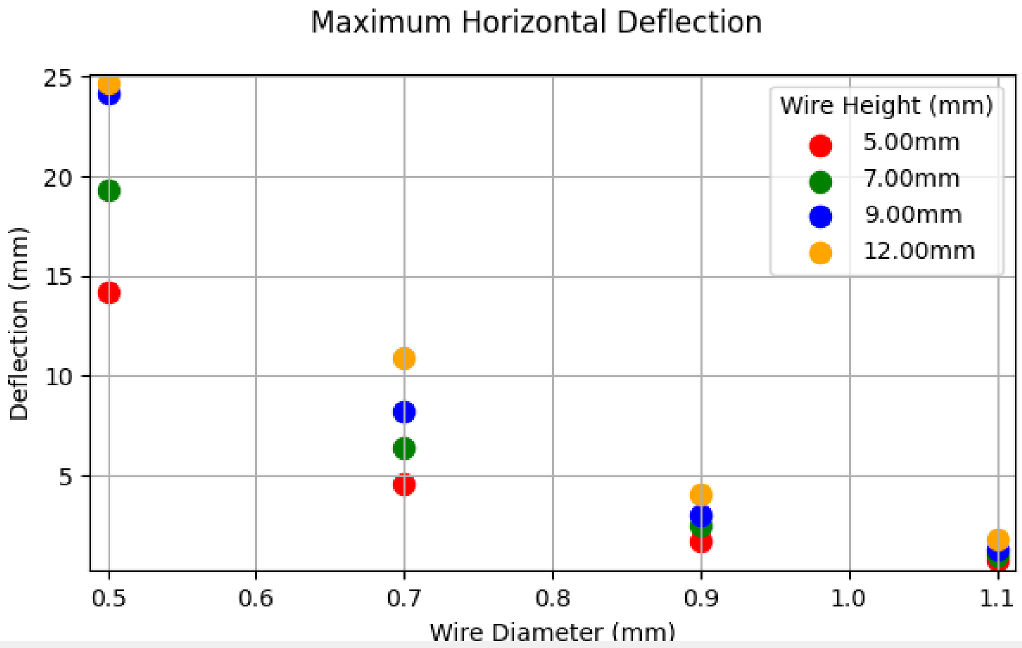


Figure 3.10: Maximum horizontal deflection of 16 different normal wire entry simulated structures.

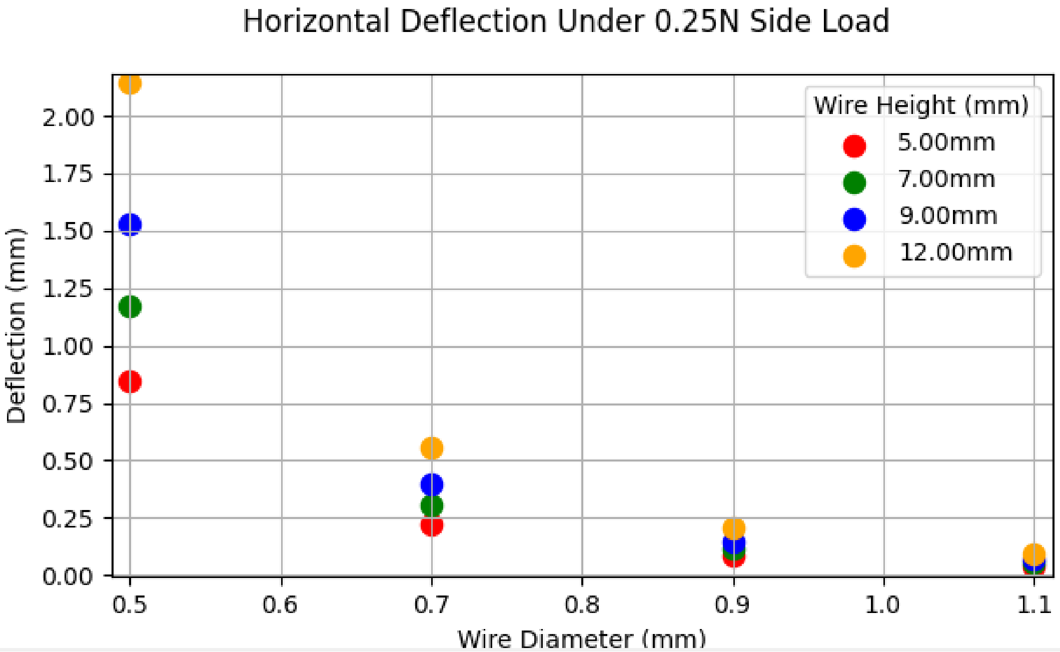


Figure 3.11: Horizontal deflection of 16 different simulated normal wire entry structures under 0.25 N of side load at their RCM.

3.3.3.2 Structure with Angled Wire Entry

The second structure examined was that in which all the wires enter the top platforms at an angle offset from the normal vector. The maximum achievable horizontal deflection of the end-effector's tooltip, under the force and position constraints of Micron's actuators, is shown in Figure 3.12. The horizontal deflection of that tooltip under a 0.25 N side load applied at the RCM is shown in Figure 3.13.

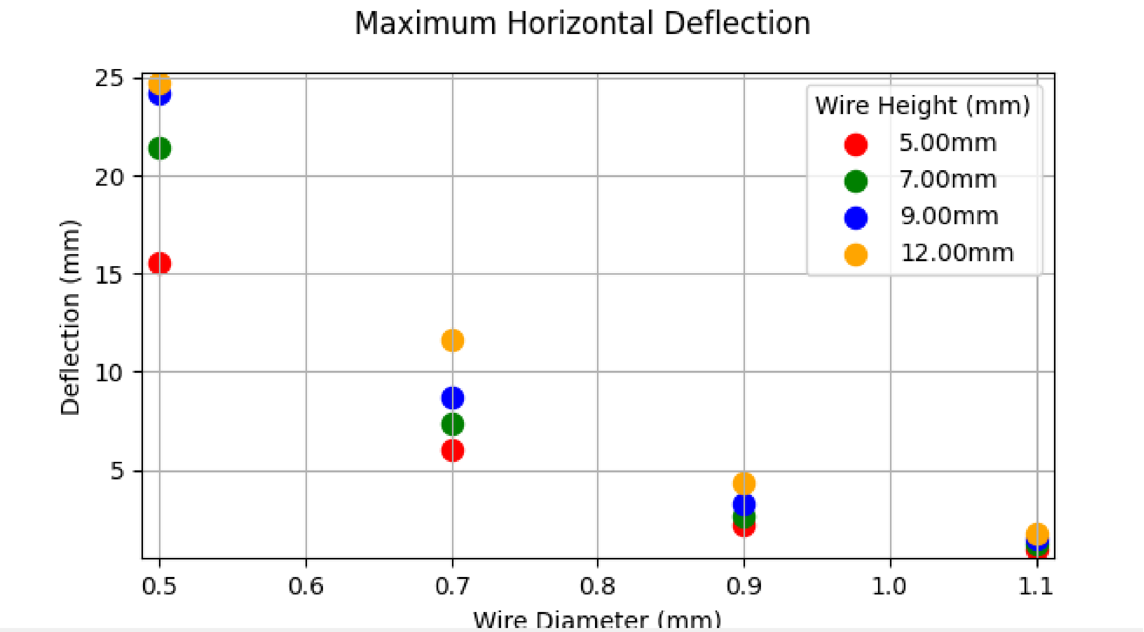


Figure 3.12: Maximum horizontal deflection of 16 different angled wire entry simulated structures.

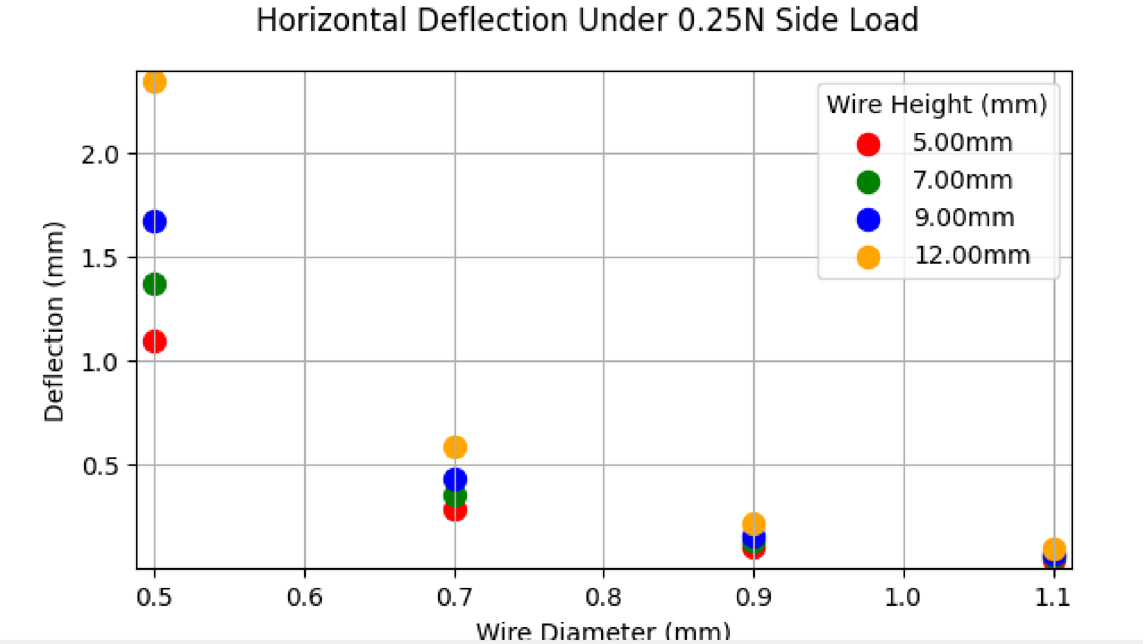


Figure 3.13: Horizontal deflection of 16 different angled wire entry simulated structures under 0.25 N of side load at their RCM.

3.3.4 Discussion

No matter if the wire entry points are angled or not, the results show the same fundamental contradiction: whenever the manipulator is stiff enough to adequately resist side load, it is too stiff to provide an adequate workspace, and whenever the manipulator is compliant enough to provide that adequate workspace, it is too compliant to resist side load. In the structure with normal wire entry, Figure 3.10 shows that a wire of 0.9 mm diameter and 12 mm height produces a maximum horizontal displacement of 4 mm, which is sufficient for our application's workspace. However, that same structure sees 0.23 mm of unwanted horizontal deflection when exposed to the specified side load in Figure 3.11. That would mean a random, uncontrollable position error of potentially hundreds of microns. The structure with angled wire entry has a very similar result: the structure with a wire of 0.9 mm diameter and 12 mm height produces a sufficiently large workspace, but is not stiff enough to resist side load. Although these simulations are limited to tilting motions only, those tilting motions should in fact be ideal conditions for the manipulator to exhibit its rigidity. Along with tilting, Micron's manipulator would be expected to perform small horizontal translations, which have been seen to greatly increase the compliance and instability of parallel continuum manipulators, even leading to buckling [82].

3.4 Applicability Discussion

Neither simulation produced a mechanical structure of a parallel continuum manipulator that would be workable for Micron. The Ansys simulation in particular exposed the fundamental contradiction between the second and third design requirements: by the time the manipulator's workspace is large enough, it is too compliant to side load at its RCM. That contradiction, viewed in ideal tilting manoeuvres, all but eliminates parallel continuum manipulators from being usable in Micron.

The fact that a parallel continuum manipulator's kinematics are closely linked to the forces applied on it implies that knowledge of those forces could permit a control algorithm to compensate for them. Specifically, knowledge of the side load applied on the end-effector's tooltip could be useful. The REACH Laboratory has published research into both deflection-based force sensing and actuation-based force-sensing [9]. Measuring deflection of the end-effector would be trivial, since we use the ASAP optical tracking system. The REACH Laboratory found significant error in this estimation method, but indicated that restricting force sensing to directions in which the manipulator is noticeably compliant, as in a side load, could greatly reduce that error [9]. Actuation-based force sensing is shown to be much more successful in their application, but would suffer in ours, given that Micron is too small to contain force sensors in all six of its legs. Finally, although knowledge of the side load would allow the controller to accurately define the tooltip's behaviour, the side load would still be present and could still prevent the manipulator's mechanical structure from compensating for it.

Throughout this analysis of the applicability of parallel continuum manipulators to Micron, it has been assumed that Micron must move with 6 degrees of freedom. That unspoken requirement has informed decisions regarding the shape, placement, and quantity of the wires. In fact, an early iteration of Micron had only three actuators and so was limited to 3-DoF movement [53]. Specifically, the end-effector could tilt in both directions and translate axially, but could not translate horizontally to produce an RCM. Early Ansys simulations that used straight, vertical wires showed impressive resistance to side load, while still achieving the desired workspace.

Chapter 4

Simulating Micron

Previous iterations of Micron were never simulated in a full kinematics and dynamics simulator. Two fundamental challenges stand in the way of a successful simulation. First, Micron is a very small device with a workspace defined in millimeters, smaller than most simulators are designed for. Second, Micron uses a human-in-the-loop control system, complicating any results from simulating the device without human input. However, significant advancements in computational power over the last decade have enabled more accurate and faster simulations. Additionally, the ability to simulate Micron brings myriad benefits. Experiments run to validate a real-time blood vessel segmentation and reconstruction algorithm used in Micron's control loop often required full days devoted to moving Micron by hand through a field of phantom vessels [87]. A simulated Micron could be moved through a photorealistic simulated environment along a pre-determined path with much less human labor and time. Recent work in surgical robotics has also focused on the automation of simple but delicate surgical sub-tasks with reinforcement learning [59, 72, 74, 80]. Such automation would be most useful in situations like retinal membrane peeling, where the interaction forces are too small for the surgeon to notice [43]. However, any implementation of reinforcement learning must consume large amounts of data collected over many trials. Simulations of the device would provide plentiful and repeatable trials.

4.1 Creating an SDF File

The first step towards a simulation is to model the mechanical properties of the robot and store it in a readable format, such as a Unified Robot Description Format (URDF) file. Micron is distinct from most simulated robots in that it has a parallel configuration with multiple closed kinematic chains. Since URDF files can only define serial kinematic chains, that distinction necessitates that it be described by a more generalized Simulation Description Format (SDF) file.

Prior work in the lab produced a SolidWorks CAD model of the manipulator. This formed the basis of the SDF file. The model was first updated to include far more accurate weights and inertia matrices for each component. These values were determined based on the material and size of each component, as calculated in SolidWorks. The many individual parts in the assembly were then grouped into links: one base platform, six lower links, six universal joint ball bearings, six upper links, six spherical joint ball bearings, and one top platform. The solidworks_to_URDF_exporter add-in within SolidWorks was used to obtain a description of the manipulator in a URDF file.

During this process, a complication arose due to the fact that URDF files can only represent revolute, prismatic, continuous, or fixed joints. To satisfy this condition, the universal joints were deconstructed into two stacked revolute joints each. The structure of the URDF file is as follows. Each lower link is connected to the base platform by a prismatic joint. A revolute joint connects each universal joint's ball bearing to its lower link. A second revolute joint connects each universal joint's ball bearing to its upper link. These two revolute joints capture the roll and pitch rotations, but not the yaw rotation.

Next, the URDF file was converted into an SDF file. Edits were made manually to the resulting file to connect all six legs to the top platform. Conversion to an SDF file also enabled the joints connecting the top platform to the legs to be changed to spherical joints. Unfortunately, SDF does not represent universal joints. A picture of the manipulator model is shown in Figure 4.1.

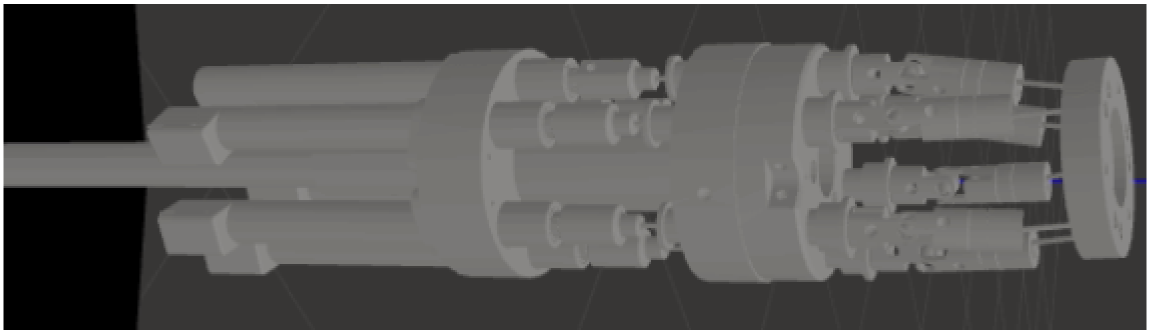


Figure 4.1: Structure of the Gazebo simulation of Micron.

4.2 Initial Simulation in ROS and Gazebo

Gazebo is one of the few simulators that can actuate joints on robots described in SDF [23]. This capability, along with its close relationship to ROS and the rapid development time that it entails, led us to choose Gazebo for our first simulation of Micron. The SDF file was imported into Gazebo and the pose of its handle was fixed to the ground plane. Gravity was disabled.

4.2.1 Controlling the Model

ROS Noetic was used to control the manipulator in simulation. The capability to control joints described in SDF is not native to Gazebo, so we used the ros_sdf plugin [41] to control Micron's prismatic joints. It implements an independent PID controller on every joint that is left as ideal in the current configuration.

The structure of the control code is shown in Figure 4.2. Each red box is a ROS node. Input commands are taken in real time from a Logitech F310 gaming controller to define a desired twist of the top platform. The left joystick controls the twist's roll and pitch orientation, while the right joystick controls the twist's planar translation. The gamepad node converts that input to the desired twists and advertises it as a ROS topic. The inverse kinematics node subscribes to that topic and computes the required position of each of the six prismatic joints to produce the desired top platform twist. Finally, the inverse kinematics node advertises these six position commands and they are input into the Gazebo simulation.

Control of the manipulator required implementing its inverse kinematics in a ROS

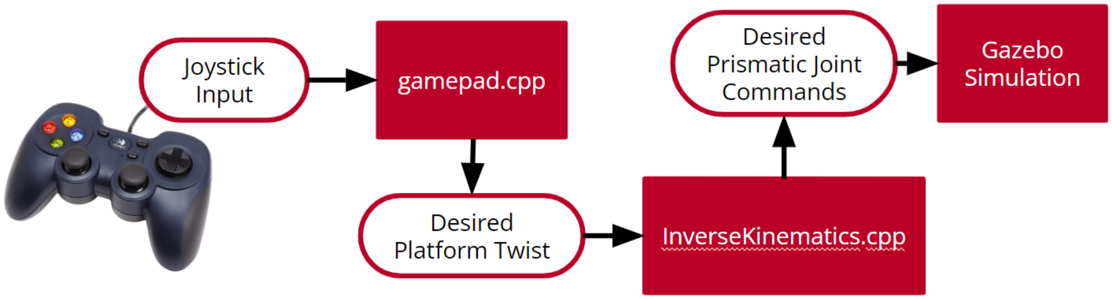


Figure 4.2: Structure of the Gazebo simulation control code.

topic. The inverse kinematics of this 6-PUS manipulator are adopted from [46] and simplified so that they take in the desired twist (position and orientation) of the top platform and output the required lengths of each leg's linear rod.

4.2.2 Results

The game controller was used to command the top platform to a desired pose, using the inverse kinematics described earlier. An additional ROS node was created to log the actual twist of the top platform and the commanded twist of the top platform. The twists were logged for 12 seconds while the manipulator was commanded to tilt its orientation, and another 12 seconds while the manipulator was commanded to translate its position.

The graph of the commanded and actual twists of the top platform over 12 seconds of tilting commands are shown in Figure 4.3. Only tilts in the roll and pitch axes are commanded. No tilt in the yaw axis is commanded.

The graph of the commanded and actual twists of the top platform over 12 seconds of translating commands are shown in Figure 4.4. Only translations in the x and y axes are commanded. No translation in the z axis is commanded.

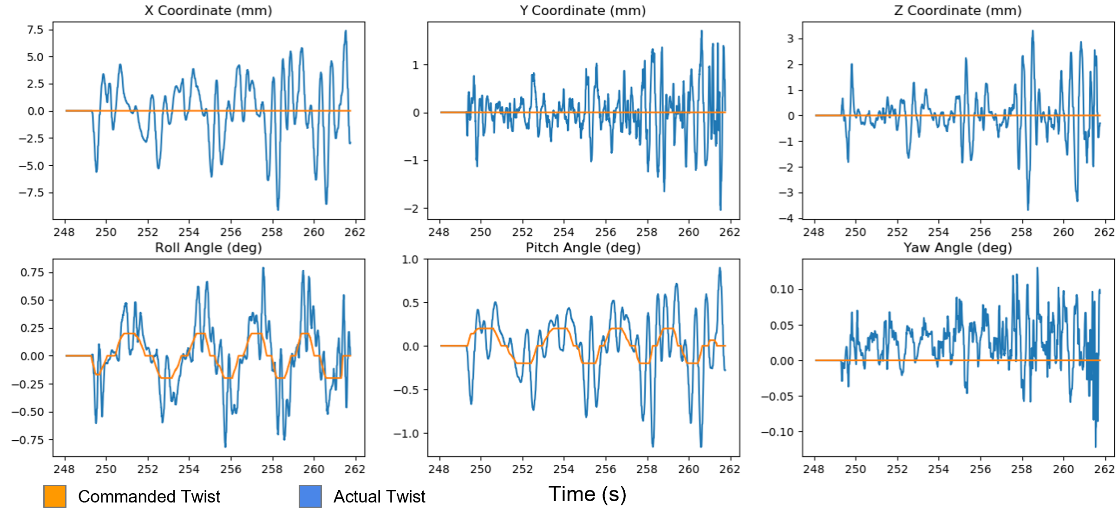


Figure 4.3: Top platform position and orientation over 12 seconds of tilting commands. The orange signal is the commanded twist and the blue signal is the actual twist.

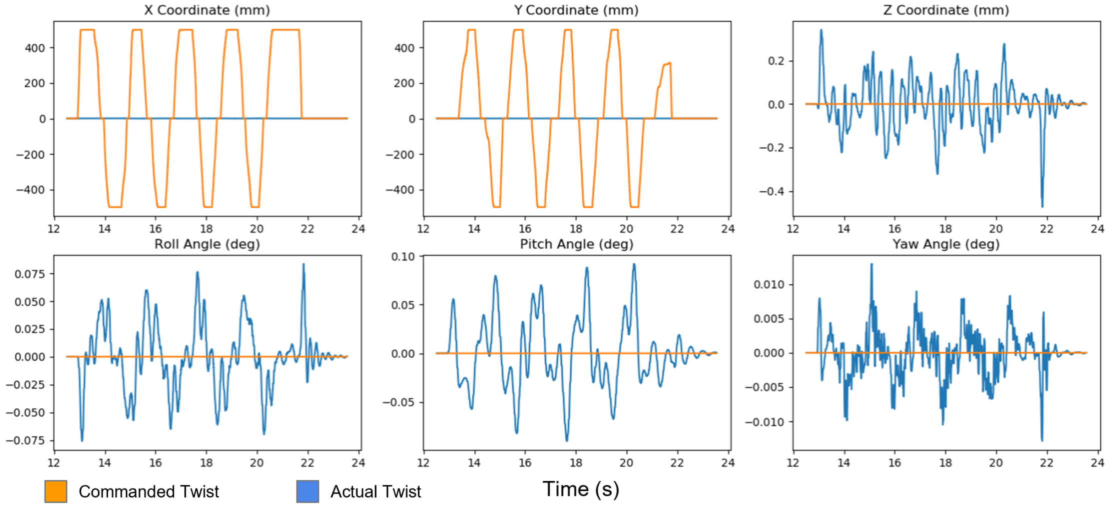


Figure 4.4: Top platform position and orientation over 12 seconds of translating commands. The orange signal is the commanded twist and the blue signal is the actual twist.

4.2.3 Discussion

The Micron model reacted to tilting commands reasonably well. At the very least, the intention of the input was visible in the output. However, the top platform's actual pose jerked around significantly. Additionally, the top platform's pose did not follow translation commands at all, instead seeming to tilt. All of these results can be attributed to modeling error.

Modeling error in the kinematic structure was present throughout the model. Small errors in the SolidWorks assembly continued to be found and corrected for months after these experiments. This is to be expected for such a tiny device that has 82 different metal parts. As a result, the orientation and positions of joints is slightly off. This contributes to the fact that the third leg of the manipulator cannot seem to connect properly to the top platform, despite being defined as connected.

A significant source of error was discovered only recently: the wires that connect the ball joints to the top platform are too long. Longer wires increase the effective radius of the kinematic top platform and push the legs to become more vertical. This in turn makes it easier to tilt and makes it harder to translate. It also pushes the legs closer to the kinematic singularity that results when the legs are all vertical. Proximity to that singularity gives the manipulator a wobbly behaviour.

There is also large modeling error in the dynamics of the manipulator. For the moment, none of the joints have been given spring constants or damping coefficients to define their rigidity or stability. Future work will make the joints far more rigid and stable. However, accurate characterization of these joints will require the ability to experiment with a physical model, which can only be performed once its assembly is completed.

4.3 Toward Automation of a Surgical Sub-task

The initial simulation of Micron in Gazebo provided a roadmap for creating further simulations, but was not the correct setup for attempting to automate a surgical sub-task. Although a lot of its modeling error could only be resolved with a physical model, improvements could be made in the meantime to better interface with standard reinforcement learning environments. With this in mind, the model was moved to

PyBullet and a heavily simplified surgical sub-task was identified and automated.

4.3.1 Background

One of the most common vitreoretinal procedures is epiretinal membrane peeling [33]. This surgery involves puncturing the sclera and using a forceps mounted at the tooltip to peel away at the membrane that covers the retina at the back of the eye. The membrane can be as thin as 10 micrometers [54] and exerts minimal forces [35]. Without perceptible haptic feedback, the surgeon runs a significant risk of placing the forceps too deep into the retina, causing damage to the retina or hemorrhages [43]. Additionally, any manipulation of the tooltip must take into account its interactions with the sclerotomy. As the tooltip moves, it can rotate the eye and tear the sclera, so the interaction force at that point must be limited.

Recently, several sensors have been designed to improve the pose and force data available to vitreoretinal surgical instruments. A depth sensor is in development to describe the depth between the tooltip and the retina. Several force sensors have been developed to measure both scleral and retinal interaction forces [43].

4.3.2 Simulation

A simulation environment of Micron performing a fine manipulation task on the retina is created. The environment is evaluated with a standard reinforcement learning algorithm. The ultimate goal is to allow a surgeon to position Micron's tooltip near the retinal membrane, press and hold a foot-pedal, and have Micron autonomously grasp the membrane so that the surgeon can further peel away at it. In this simulation, under several simplifying constraints, Micron grasps and lifts a small rigid object that is placed on the retina.

4.3.2.1 Infrastructure

Gazebo is unable to simulate deformable bodies, which is of interest in the future, so Micron needed to be ported into a new simulator. PyBullet is a simulator that can model soft bodies [23], [24]. Being open-source, it also has many examples and tutorials to learn from. The fact that it uses Python made it especially quick to use. When combined with a custom OpenAI Gym environment [15], it enabled us to quickly implement Stable Baselines' reinforcement learning algorithms in our environment [39].

4.3.2.2 Simulating Micron

In the time between this simulation and the Gazebo simulation, multiple small kinematic errors were fixed in the SolidWorks model. Additionally, we added a forceps that had been used with an earlier iteration of Micron onto the top platform of the manipulator. The forceps uses tiny, pre-curved, metal fingers that close as they are pulled back into their sheath.

In this simulation, Micron can actuate its six prismatic joints, as well as its forceps. Since we limit the tool to axial translation, all six prismatic joints move in unison. The mechanics of the forceps also dictates that the fingers move in unison.

Micron is a parallel manipulator and contains many closed kinematic chains. However, PyBullet does not allow URDF files to describe closed kinematic chains. Nor does it enable actuation of joints described in SDF. Instead, PyBullet allows the user to create constraints that fix a point on one robot to another point on a different (or the same) robot. We first separated the base platform and the top platform of the manipulator into two different URDF files and connected them with six constraints. However, these constraints were enormously insufficient. Moving the prismatic joints in slow patterns often caused the constraints to become unstuck, separating the top platform from the base. This was likely due to the tiny size of Micron's joints. We repeatedly encountered problems due to the fact that PyBullet was optimized for much larger robots.

In order to actuate the manipulator using its prismatic joints, we implemented its inverse kinematics in separate code. The inverse kinematics of Micron are described in [46]. We could not rely on a library for the inverse kinematics, because the manipulator has closed kinematic chains. Significant modeling error in the inverse kinematics was manually worked out, but error still exists because the behavior of the universal joints and the six constraints was never simulated realistically by PyBullet. We could not solve this issue without changing the simulator, so we decided to limit the manipulator to axial translation only.

We encountered more problems when trying to simulate the contact between the forceps and other small objects. The translation of each gripper is limited to 0.5 mm. However, PyBullet hard codes a collision margin between objects of not less than 1 mm. The only solution we could implement without changing the simulator was to push the collision meshes of each gripper back so that they could hold the objects. This issue significantly impairs the accuracy of the simulation and further demonstrates that PyBullet was not meant to simulate such small interactions.

4.3.2.3 Simulating the Eye

A CAD model of the eye was taken from [64] and bisected. For simplicity, all parts of the eye were kept rigid. Each half was converted into a URDF, with one half having no collision. The half with no collision was pierced by Micron's tooltip. Micron interacted with the half of the eye that had collision enabled.

4.3.2.4 Simulation Task

A small rigid sphere of diameter 0.5 mm is placed on the retina inside the eye. Micron is placed directly above it, inside the eye, so that it could touch it simply by extending axially. The task for the manipulator is to reach down to the ball, grasp it in the forceps, and lift it back to the start state. This is shown here in Figure 4.5.

This is the first simulation of Micron completing a task. As such, several simplifying assumptions were made to keep the task tractable. As previously mentioned, all bodies are rigid. Additionally, we fix the handle of the manipulator. This assumes that the low-level position controllers of the manipulator are able to perfectly cancel out the surgeon's hand tremor. We also only move the tool axially, which negates the

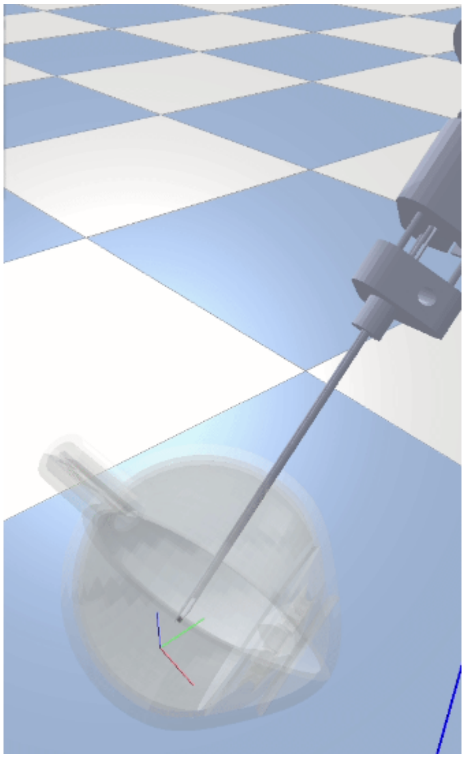


Figure 4.5: Simulation Environment

significant modeling error stemming from the often-unstuck constraints we used to model our closed kinematic chains. Finally, we assume no scleral collision, as this would require us to model the sclera as a soft body.

Several random factors were added to the initialization of the simulation environment. The angle of the tool relative to the retina was allowed to change in the range $[-\pi/8, \pi/8]$ radians and the distance from the ball was allowed to change from 3 to 8 mm. Further random factors, including the friction of the ball and sensor noise were abandoned due to difficulties in training the model.

4.3.2.5 Sensors and Constraints

In creating this simulation, we wanted to use sensors that were as accurate to the use case of the manipulator as possible. This included a literature search for real sensors that we can accurately model. We used a depth sensor, a force sensor to measure interaction forces between the retina and tooltip, force sensors to measure grip forces of the forceps, and a sensor for the position of the grippers. The two force sensors are vital, as we place hard constraints on the forces that can be applied to the retina and membrane.

A 1-DoF visual depth sensor is in development in the Surgical Mechatronics Laboratory at CMU. It uses a combination of the visual information to ascertain the distance from the tooltip to the retina. This information includes the size and intensity of the light emitted by a light pipe, as well as the shadow cast by of the tooltip. Since no findings on this sensor have been published yet, we kept perfect resolution and range for this sensor.

A 6-DOF force sensor was used to measure the interaction force between the retina and tooltip. Based on a sensor in literature, a 1 cm long, 160 μm diameter Fiber Bragg Grating (FBG) strain sensor was embedded in a 0.5 mm diameter tool shaft [42]. The FBG sensor used was the os1100 Fiber Bragg Grating from Micron Optics, Inc. [3], and the sensor was designed to have a wavelength interrogator with four channels and a resolution and scan frequency of 0.001 nm and up to 2 kHz, respectively. Additionally, the force resolution for the sensor was 0.001 mN and the sampling rate was greater than 100 Hz, with confirmed functionality up until 0.25 mN [42].

To measure grip force we utilized grip force sensible forceps with two optical FBGs attached to two grasping locations to measure the pinch forces of the forceps. Commercially available optical FBG sensors with dual 5 mm grating and the ability to measure tensile strain and thermal variation were used [49]. The sensors are assumed attached in two different locations to account for both the mechanical and thermal strain. One grating is assumed to be positioned on the gripper where the strain was most concentrated during gripping in order to measure mechanical tensile strain. The second grate is assumed to be located where strain was not concentrated to compensate for thermal strain. We can utilize the principle that the strain done when gripping occurs will change the peak wavelength of the gratings and the difference in the wavelength between the two gratings will give us the grasping force. The force resolution of this sensor was 1 mN with a confirmed resolution up until 11 mN [49].

In order to ensure minimal damage to the retina, the previously mentioned sensors were constrained with the following conditions: 1) maximum speed of the tooltip: 1 mm/s, 2) maximum speed of the grippers: 0.1 mm/s, 3) maximum force exerted on the retina: ∼2 mN [79], and 4) maximum grip force exerted on the ball: ∼325 mN [79]. Data from the simulation environment was actively fuzzed to be only as accurate as the sensors would allow.

4.3.3 Problem Formulation

4.3.3.1 Reinforcement Learning Algorithm

In order to evaluate the simulation environment, we used the Deep Q Networks (DQN) algorithm available in the Stable-Baselines3 Python package [57]. We chose this over model-based RL because we did not have a good reference movement trajectory based on real-world data for the full dynamics model.

4.3.3.2 Action States

Using DQN by Stable-Baselines3 with OpenAI Gym meant we had to use a Discrete action space. Three actions moved the simulated Micron an incremental distance closer to the retinal membrane, three actions moved it away from the retina, and three kept it in place. Within each group, one action would increase the Micron gripper position (widening the gripper) by an incremental distance, one action would decrease gripper position, and one would maintain it with no change. This action space of 9 was chosen because a Micron operator would have direct position control of the tool in both dimensions simultaneously with fine feedback. (We are also assuming a perfect actor whose inputs are exactly as planned and does not make mistakes or conduct unwanted behaviors.)

4.3.3.3 Reward Function

The grasping problem we tackled in this project is very different than most grasping done by robotic simulations because we do not directly observe the position of the ball in world coordinates. The retinal membrane is transparent and colorless and the small workspace of retinal surgery makes intraoperative illumination and photography extraordinarily challenging. Our determination of whether the ball is in our grasp and has been lifted can only be informed by the force and position sensors on the gripper. To compensate for this, we referenced a recent paper on robotic grasping [45] that used RL with a hierarchical reward function that creates three different reward functions and then activates them in sequence, adding later stages of reward or deactivating previous stages when certain triggers are met by state observations.

Similar to how the paper did things, we start with an Approach Stage rewarding forward movement to the target. This was defined by providing an approach reward ($r_{approach}$) that increased as the depth observation (measured from the Micron tooltip to the retinal membrane) went down. The reward penalized moving away from the initialized position of the grippers, but it was small to encourage exploration. This stage continued until Micron arrived at the retina membrane. The reward function would then switch to the Grasp Stage. This was checked from both depth and force along the axis of travel, i.e. retinal force, of the observation. If the depth observation was near zero, or if the observed retinal force met a safety threshold of force constant, the stage would advance.

The Grasp Stage rewards closing the grippers around the target. r_{grasp} was structured to penalize any non-zero value for observed position of the grippers, which rewarded closing the grippers. r_{grasp} penalized even more heavily moving the Micron tool along its axis of travel to prevent piercing the retina. The Grasp Stage would end until the observed force against the grippers met a gripper force threshold (which was set to a value that meant it was highly likely the target was being gripped) or the observed position of the grippers was 0.

This would change the state of the reward function to the Lift Stage, which rewarded grip and negative movement. r_{lift} gave a reward directly proportional to observed depth, the opposite of the Approach Stage. However, it also penalized if the observed gripper position increased. This stage would continue until the observed depth was equal to the depth at initialization, at which point the episode would terminate. These stages are shown in Figure 4.6 below.

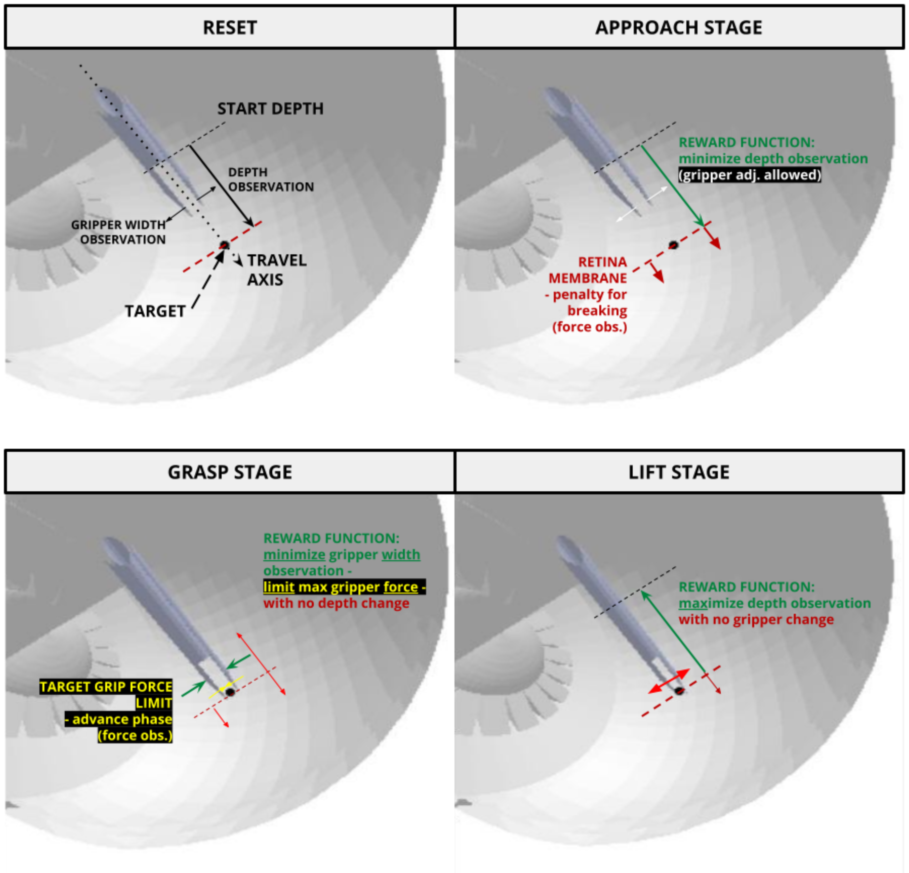

RESET
APPROACH STAGE
START DEPTH
DEPTH OBSERVATION
GRIPPER WIDTH OBSERVATION
TRAVEL AXIS
TARGET
REWARD FUNCTION: minimize depth observation (gripper adj. allowed)
RETINA MEMBRANE - penalty for breaking (force obs.)
GRASP STAGE
LIFT STAGE
REWARD FUNCTION: minimize gripper width observation - limit max gripper force - with no depth change
TARGET GRIP FORCE LIMIT - advance phase (force obs.)
REWARD FUNCTION: maximize depth observation with no gripper change

Figure 4.6: The reset state and three stages of the hierarchical reward function designed for the pick-and-place task in the constrained environment.

4.3.4 Results

After experiencing difficulty in getting the simulated manipulator to change phases within the hierarchical reward function, we also took advice from more experienced robotics engineers. One important suggestion we implemented is a decreased reward coefficient, which works by penalizing any future rewards gained after violating a hard constraint, in order to teach the DQN system that reaching those states would make any successes less rewarding [20]. Unfortunately, after over a week of attempting various ways to do the hierarchical function we had limited success. No matter what we did, we could not get the policy to adapt when the reward function switched stages.

The core issue with the hierarchical reward function was that the policy proved unable to learn when the reward function had switched stages. If we started it on the approach task, it quickly learned to move the tooltip deeper, but it could not figure out when to stop and grasp - even as the grasping reward function penalized deeper motion. While it was moving forward, it simply did not have enough time to explore and adapt before hitting the retina and crossing our safety constraints. Attempting to circumvent this, we started the simulation in the grasping stage and tried to teach it to grasp and lift. However, we encountered the same issue: it learned to grasp, but not when to stop grasping. Even as the reward decreased with further grasping and it violated the hard safely constraint, the policy continued to believe that further grasping was desirable.

Our workaround to show that RL could accomplish this three-staged maneuver was to separate it into three separate policies (one for approach, one for grasp, and one for lift). Reaching a terminal success state for the first sub-policy would complete it and pass on its final state as the initial state for the second sub-policy, and a similar relationship exists between the second and third sub-policies. We chose this approach because we reasoned that a more general model could use action primitives to learn when to use each state's policy. This approach proved to be much more successful, since each policy only had to learn one, unchanging, reward function.

The final approach taken with three sub-policies (one for each stage of the hierarchical reward design) was successful. The simulated results can be seen in the video captured at this link (note that the camera follows the ball). The rewards during training for each sub-policy are displayed on the next page in Figure 4.7. Each sub-policy corresponds to a stage in the hierarchical reward system we designed. Each graph shows some exploration before quick convergence to the best reward and optimal sub-policy, indicating the need for exploration once converged is minimal and the optimal trajectory is quickly found and optimized.

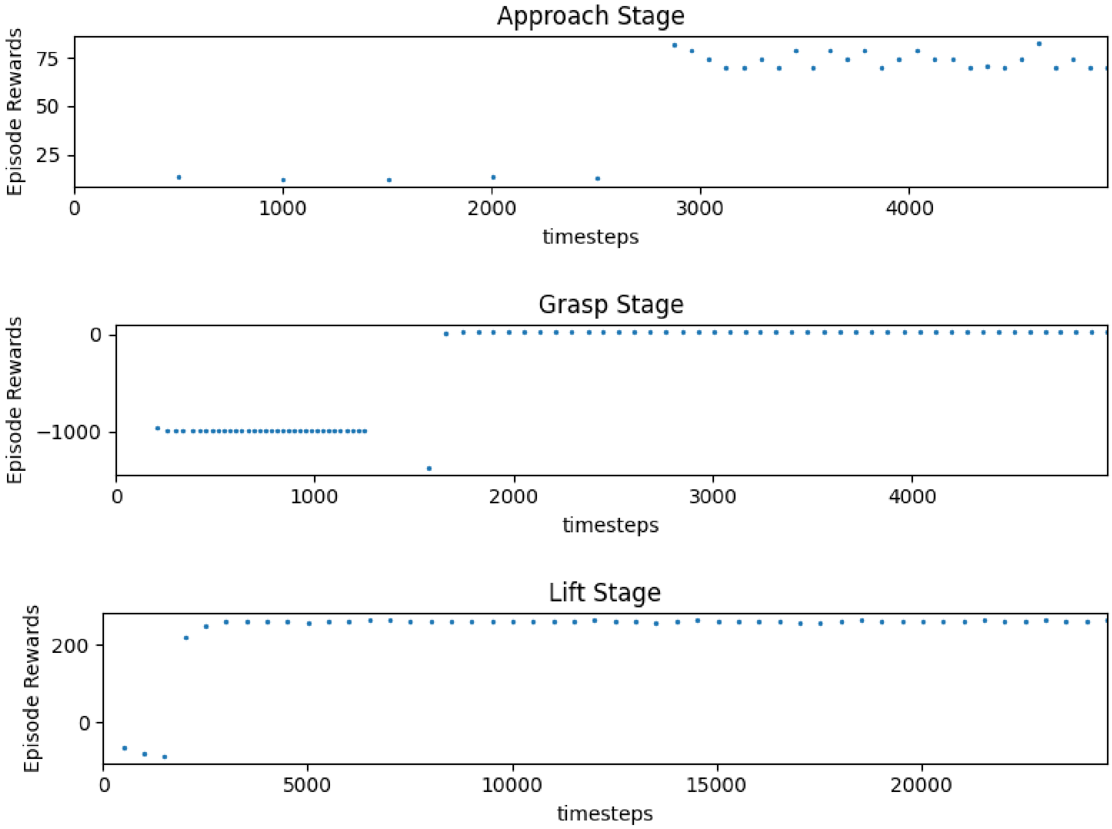


Figure 4.7: The episode returns for the three sub-policies train to accomplish lifting in the simulation of the human eye.

4.3.5 Discussion

This project was a first step towards simulating Micron and it was full of insights. Chief among them was that the PyBullet simulator is not well suited for simulating Micron because it cannot adequately simulate closed kinematic chains and it often experiences difficulty simulating such small motions and interactions. More than anything, this project highlighted the severe sim-to-real gap that would exist in any simulation of Micron. Such a small device, with such small motions and forces, is highly sensitive to the idiosyncrasies of its mechanical components and motors. Significant work would need to be done to capture these idiosyncrasies in a different simulator that can more effectively use them. Provided that such a simulation were created, the membrane grasping sub-task that we automated could be reformulated into a system identification task, where learning from demonstration during surgery is used to identify the force and position thresholds beyond which we should switch to a different stage (approach, grasp, or lift). Lacking any physical data, we were forced to hard code those thresholds into our project. Realistically, though, they are the most important and variable factors for success of the grasp.

Chapter 5

Constructing Micron

The main contribution of this thesis has been the physical construction of a new iteration of Micron. The new iteration is a recreation of the KIST microsurgical robot, adapted to interface with our group's advanced control systems and to use the ASAP optical sensor. The KIST microsurgical robot was chosen as a basis for the new iteration because it replaces the finicky piezoelectric motors of prior iterations with much more reliable brushless DC motors. The motors also enable a much more rigid tooltip that can withstand side load at the remote center of motion (RCM) of up to 5 N [46]. However, the electromagnetic (EM) tracker it uses to track its pose introduces three main limitations. First, it achieves a position root-mean-square error (RMSE) of 10 μm, but only when the electromagnetic source is within 150 mm of the tracker [46]. At a distance of 250 mm, the position RMSE increases to 20 μm. In a microsurgical procedure, that much error can be severely limiting. The ASAP optical sensor has a position resolution of just 4 μm in its workspace [52, 94]. Secondly, EM trackers are highly sensitive to ambient EM fields and cannot be used with patients who have medical devices such as pacemakers [77]. Finally, the EM tracker only allows tracking of the pose of the handle [46]. This limitation forces the control system to use forward kinematics to estimate the pose of the end-effector from the link-lengths of its motors, which introduces imprecision wherever the ideal kinematic dimensions of the manipulator do not perfectly match its tiny, complex parts. Additionally, although the motors are equipped with encoders, they are relative, and cannot accurately tell the absolute position of each link without being homed. Since no homing method can be easily implemented on this manipulator, the encoders cannot provide the absolute position of each link. ASAP optical markers can be mounted on both the base and top platform to provide a measurement of the full 6-DoF poses of both the handle and the end-effector. All of these factors combined made integrating with ASAP particularly appealing.

This chapter covers the construction of Micron, beginning with gathering information about the KIST microsurgical robot from the ANSUR Laboratory and ending with a test that analyzes feedback from the ASAP sensor as the manipulator moves all of its motors. The process has been long, limited at times by severe hardware delays and a lack of funds, but it represents the most important contribution of this thesis.

5.1 Manipulator Parts

5.1.1 Fabrication

The mechanical construction of Micron necessarily began with the CAD files and parts drawings of the KIST microsurgical robot provided to us by Dr. Sungwook Yang. With those as a guide, we created a new SolidWorks assembly of the manipulator to better model it. The assembly is shown in Figure 5.1. There are 82 parts, not counting the motors or standard screws. The motors are all inserted into the base and held in place by motor holders. Their heads connect to motor couplers, which in turn connect to custom lead screws. These lead screws turn the motor heads' rotations into axial translations of the connected universal joints. The universal joints pivot along the tilting axes and connect to the ball joints. The ball joints allow 3-DoF rotations and connect up to the top platform, on which an end-effector can be mounted.

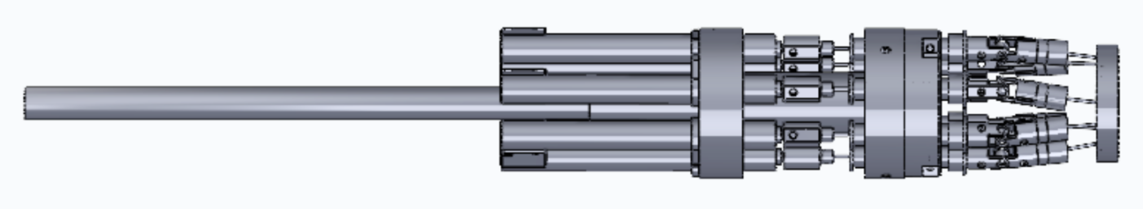


Figure 5.1: SolidWorks assembly of the manipulator.

We used our assembly to create new parts drawings of the parts. Very few tolerances existed on the original parts drawings. Such small parts that must move together dynamically require many tolerances, especially where they must fit together. We defined tolerances for 18 distinct shaft and hole fits according to the guidelines in ISO 286-2:2010.

The complexity of these small parts made it impossible to get them fabricated within our university, forcing us to look for an outside fabricator. In the process of defining our tolerances, we found that many of them were tighter than the guaranteed minimum tolerance of even the highest precision fabricator. We discussed the issue with all quoted fabricators and were advised on the tightest tolerance each was willing to quote. Two local fabricators, Conturo and Astley Precision, were willing to quote tolerances as tight as 12.7 μm, but their prices were beyond our budget. SuNPe Manufacturing, based in China, was willing to quote tolerances as tight as 15 μm. Their prices were significantly cheaper and within our budget. We relaxed tolerances for 8 of the fits and obtained assurances that the remaining 10 fits would be tested on video for us to review before the parts were shipped. We also switched all the fits to use hand assembly, rather than a press fit, so that they could be more easily tested for successful assembly.

With the decision made that the parts would be purchased from SuNPe, we attempted to order the parts. Unfortunately, processing times within our university delayed the purchase by several months. The parts arrived in January of 2024. They are shown in Figure 5.2. All the parts were either as ordered or were quickly refabricated to be up to the tolerances in the drawings.

The only manipulator parts that were not ordered from SuNPe were the screws, ball bearings, o-rings, and nitinol wire. Varying lengths of M1 screws are used in the manipulator, all of which were ordered from McMaster-Carr and NBK. Ball bearings for the universal joints were also ordered from McMaster-Carr. O-rings provide a buffer between the base plate and the bottom of the universal joints so that they don't slam together. Superelastic nitinol wire is used to connect the motor couplers to the motor screws, as well as to connect the ball joints to the top platform. Wire that was slightly thicker than that used in the KIST microsurgical robot was used because we had it laying around in the lab. The appropriate holes in other parts were ordered wider to accommodate the wire. The lengths of the nitinol wire were

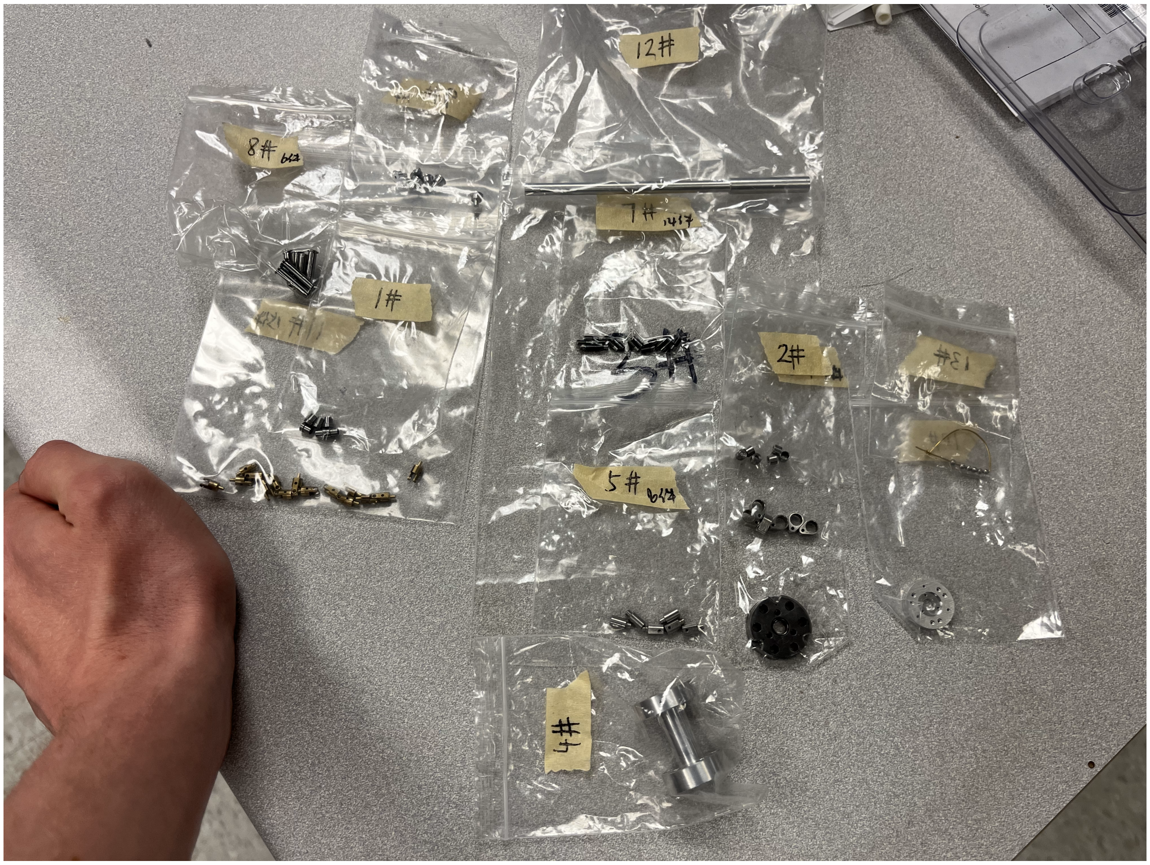


Figure 5.2: The manipulator parts as they arrived in the lab.

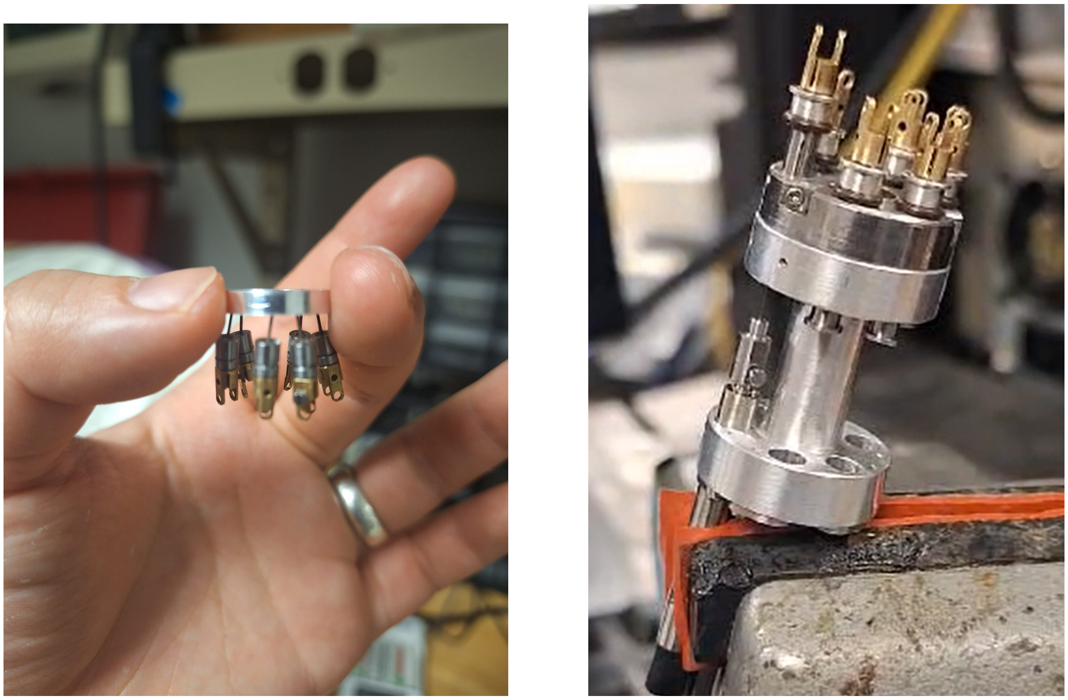
cut in the lab. The wires that connect the motor couplers to the motor screws were cut to be 17 mm. The wires that connect the top platform to the ball joints were cut to be 8.5 mm. Additional modifications had to be made to six 3 mm long M1 screws that were used in the lead screw. These screws screw horizontally into the base plate and interface with a slit in the lead screws' slide shafts to restrict the shafts from rotating. The screws were precision machined to be slightly shorter and pointier so that they fit into that slit with minimal friction.

5.1.2 Assembly

Connections between the 82 parts of the manipulator are made either with screws or epoxy. Specifically, Epo-Tek 301 two-component optical grade epoxy was used. This epoxy has a minimum bond line thickness of just 3 μm and can operate at temperatures of up to 300 C, making it well-suited for a manipulator of this size. The two components of the epoxy are mixed with a 20:5 ratio using eyedroppers and coated onto the connection before the mechanical parts are fit together. The parts are then left to cure. Most parts were cured on a soldering hot plate that was placed inside an insulated reusable wine bag for 2 hours at 65 C. If heating the mechanical parts to this temperature was inadvisable, the epoxy was cured over 24 hours at room temperature.

The order in which the connections were epoxied was critical. The first connection epoxied was between the base body and the wire guide pipe, which fits into its center hole to provide access along the manipulator for any end-effector wires or tubes. It was epoxied first to test our setup on an easy connection. With the setup proven, the other connections were epoxied in accordance with their dependencies. The ball joints were particularly challenging in this regard. The ball joint includes a socket, a ball bearing, and a housing. The ball bearing fits into the socket and the housing fits over it to keep the ball from escaping. The ball bearing has a hole in it to connect to the 8.5 mm nitinol, which in turn connects into the top platform. The connection between the nitinol and ball bearing was epoxied first. The upper half of the universal joint was also epoxied to the ball joint's socket. Then, the ball bearing was placed into the socket and the housing was epoxied to the socket. This last connection was made at room temperature to avoid melting previously cured epoxy and accidentally fixing the ball in place. The absolute last connection epoxied was the connection between the top platform and the nitinol wires. Again, this was cured at room temperature. These epoxied parts are shown in Figure 5.3a. The rest of the epoxied parts are shown in Figure 5.3b.

Each of the motors had to be epoxied to a teardrop-shaped motor holder that fits and screws into the base body. Each of the motors has a set of sensors in a black box on its back end which must face inward towards the central axis of the manipulator. This makes the position and rotation of the motor holder along the motor critical. In order to epoxy these parts in the correct pose, a stand was printed, shown in Figure 5.4. Epoxy was placed on each motor, their holders were positioned on them, and they were placed in this stand to cure at room temperature for a day to avoid heating the motors.



(a)

(b)

Figure 5.3: The manipulator parts epoxied together. (a) The top parts. (b) The bottom parts.

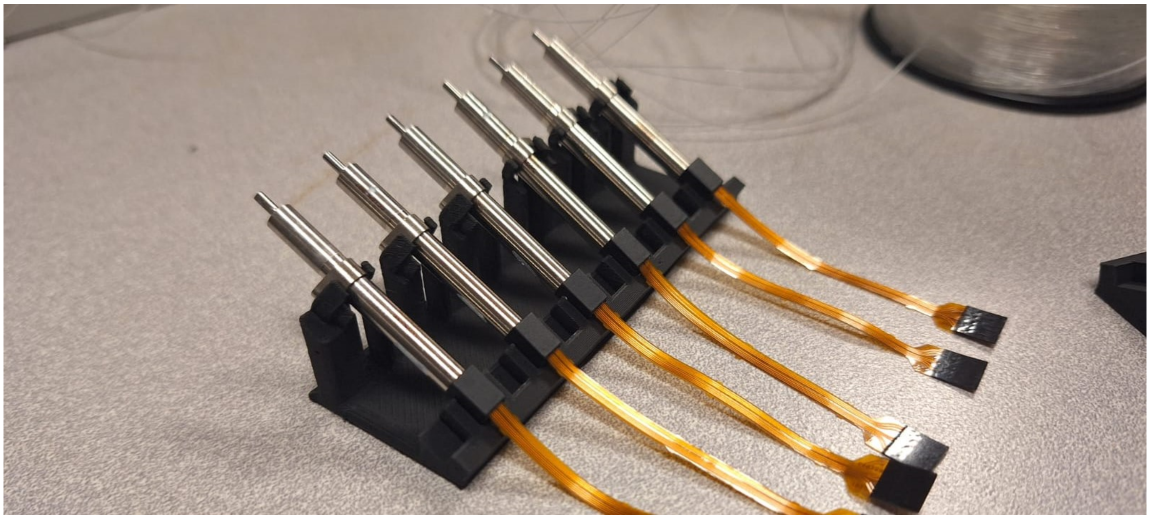


Figure 5.4: The stand in which the motors were epoxied to their holders.

The rest of the parts were more straightforward to epoxy; only a few more considerations were made. When the bottom half of the universal joint was epoxied to the lead screw, it was epoxied at an angle that allows easy access to its tightening screw. When the lead screw's slide shaft was epoxied to the adapter between it and the universal joint, care was taken to ensure epoxy did not seep into the threads. The final assembled parts are shown in Figure 5.5.

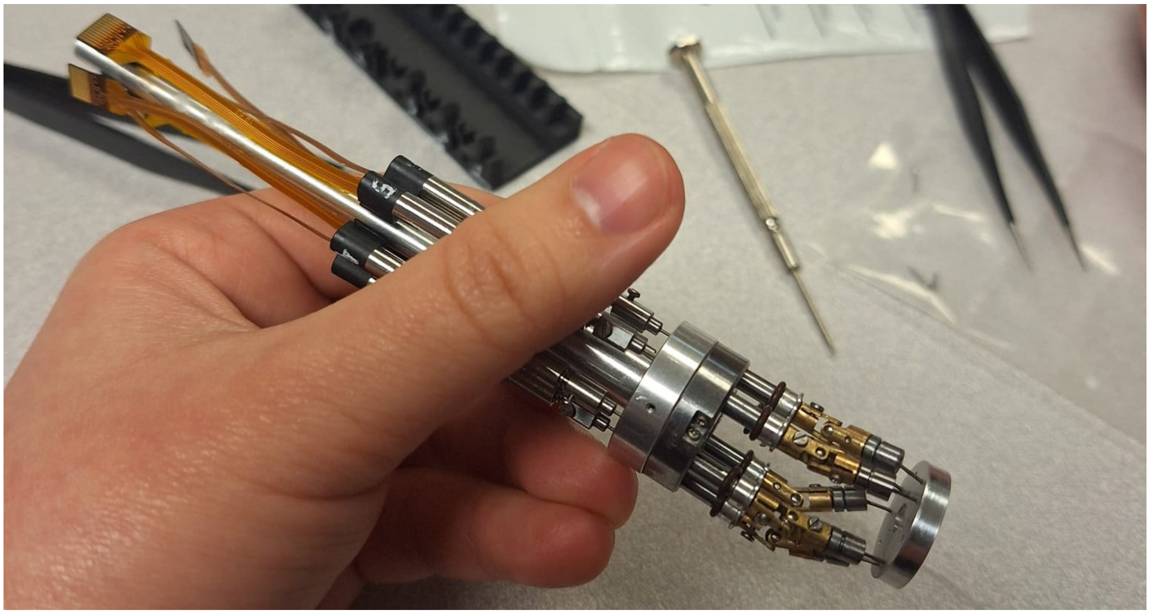


Figure 5.5: The assembled manipulator parts.

Several errors were made and had to be fixed during the assembly of the manipulator. These include accidentally fixing a ball joint in place and fixing motor holders to the wrong position along their motors. A variety of methods were tried in order to remove the epoxy and disconnect the parts. First, repeatedly boiling the ball joint in an old pot of water melted the epoxy and helped disconnect it from the top half of the universal joint. However, it proved ineffective at getting the ball joint's housing free from its socket. Second, bathing the ball joint in toluene, a strong solvent that is noncorrosive to most metals, was attempted, but proved ineffective. The ball joint was particularly challenging to disconnect because its connected surface area was greater than a shaft and hole fit. In the end, a new ball joint was ordered. The final method of disconnecting epoxied parts was heating them to 400 C on the soldering hot plate. This was used to successfully salvage a motor holder from a broken motor.

At that temperature, the epoxy quickly oozed out of the crevice between the parts as ash and tweezers were used to separate the parts.

5.2 Electrical Parts

In preparation for wiring up the full electrical setup, several custom PCBs had to be designed and assembled, as discussed in the below subsections.

5.2.1 Custom Motor Electronics

The design of the electronics of the manipulator started with the motors. Seven EC4-431182 Maxon motors were ordered from Maxon Precision Drive Systems, six motors for the manipulator and one as a spare. Six of Maxon's EPOS4 Compact 24/1.5 EtherCAT drives were also ordered. These drives communicate with a master controller over an EtherCAT daisy-chain and each connect to a single motor. They allow control of the motors in a variety of modes using current, velocity, and position PID controllers. The motor and the EPOS4 drive are shown in Figure 5.6.

(a)

(b)

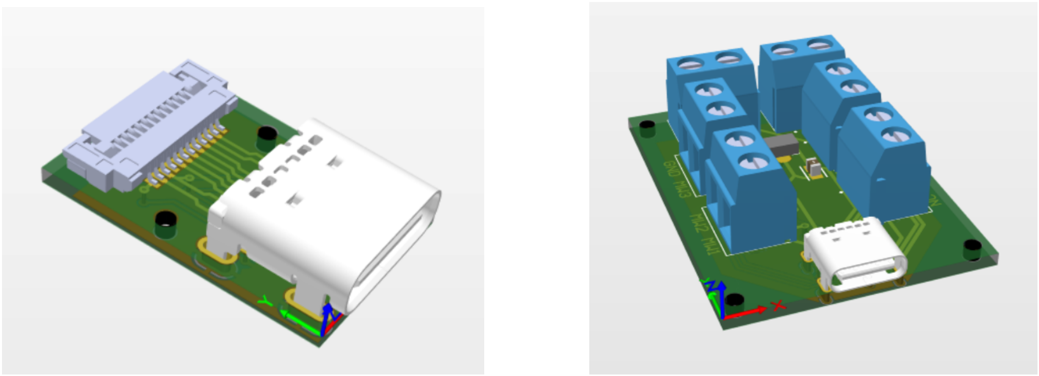
Figure 5.6: The maxon motor (a) and maxon EPOS4 motor driver (b) used in the manipulator.

The drivers are bulky and must be held separate from the manipulator inside a desktop enclosure. However, the motor that connects to them has a flat flexible connector (FFC) that is only 50 mm in length. Therefore, some means of lengthening the connection needed to be created.

Each motor has 11 distinct signals it must connect to its driver. They power the motor (all three windings), they provide a common ground, they provide power to the sensors, they return data from the digital incremental encoder (A, B, and I), and they return data from the Hall sensor (H1, H2, and H3). These signals must be hooked up for all six motors, meaning a total of 66 independent signals must be routed from the motors to their drivers.

The cheapest method of delivering these 66 signals is to use a data-carrying USB-C cable for each motor. Each cable has 16 independent wires. The motor windings and ground signal must carry increased current, so they were allocated thicker wires. Ground and one of the winding signals were allocated 24 gauge wires. The other windings were allocated three 32 gauge wires each. All other signals were allocated a single 32 gauge wire.

Two PCBs were designed to convert the USB-C connectors on either end of the cable into the motor's FFC and the motor driver's terminals, respectively. They are each shown in Figure 5.7. The Motor Adapter PCB is just 16.5 x 11 mm. It can be mounted with two M1 screws. The Driver Adapter PCB has screw hole terminals to connect to the EPOS4 driver. It can sit freely next to the driver. The data-carrying USB-C cables with the smallest footprint possible were purchased. They are 2 meters in length.



(a)

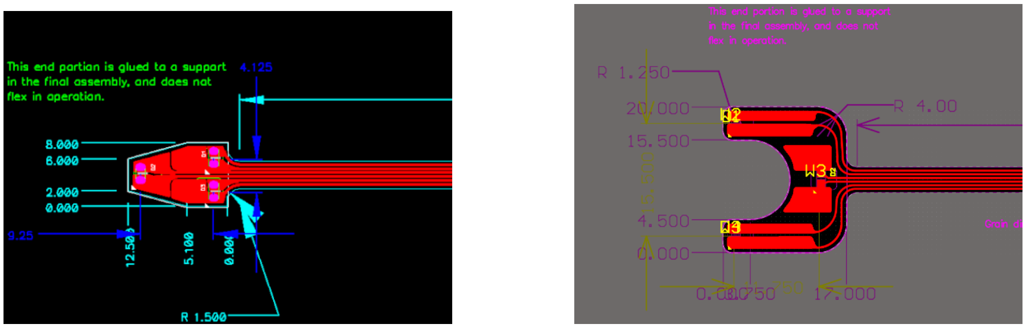
(b)

Figure 5.7: The motor adapter PCB (a) and driver adapter PCB (b).

Soldering these boards turned out to be a non-trivial and time-consuming task. Low-temperature Bi57 ChipQuik solder paste was used, along with stencils and a hot plate. When all the original quantity of ordered boards was used up, new boards with an immersion gold finish had to be ordered, which greatly improved their solderability.

5.2.2 Custom ASAP Electronics

Two PCBs were created to connect up the ASAP electronics. In order to find the poses of the base platform and the top platform, the ASAP optical sensor must be able to see three infrared LEDs mounted to each platform. The three LEDs mounted to the top platform are located on a triangular flexible PCB, which delivers power to them. The three LEDs mounted to the base platform are located on a fork-shaped flexible PCB. These two PCBs are shown in Figure 5.8.



(a) (b)

Figure 5.8: ASAP's triangle-shaped flexible PCB (a) and fork-shaped flexible PCB (b).

ASAP's flexible PCBs have been used for earlier iterations of Micron [93] and had already been designed and created. However, only the fork-shaped PCB was still around in the lab. The triangle-shaped PCB needed to be reordered and assembled. Old KiCad files for the triangle shape were found and imported into Altium Designer, where the dimensions of the PCB were recreated as closely as possible. New copies were ordered and assembled with the same LEDs.

Each flexible PCB provides a common voltage source and independent drains for its LEDs, which each pulse at different frequencies. Therefore, seven independent signals had to be routed back to ASAP's desktop enclosure. Similar to the motor signals, this was done with a USB-C cable. In this case, one end of the USB-C cable was adapted to use a 10-pin Hirose HR10A connector, which plugs directly into

ASAP's desktop enclosure. The ASAP Adapter PCB that mounts on the manipulator is shown in Figure 5.9. It takes in the FFCs from the ASAP markers and converts them to a USB-C receptacle. This PCB was ordered and assembled.

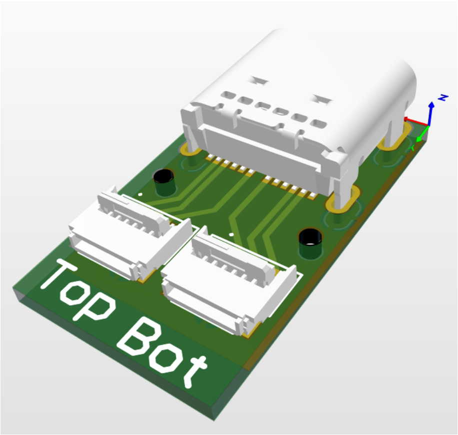


Figure 5.9: The ASAP adapter PCB.

5.3 Auxiliary Parts

Several auxiliary parts were designed to be placed on the manipulator. They include a handle, PCB mount, mounts for ASAP's optical markers on both the base and top platform, and a desktop enclosure. An updated CAD assembly of the overall assembly can be seen in Figure 5.10.

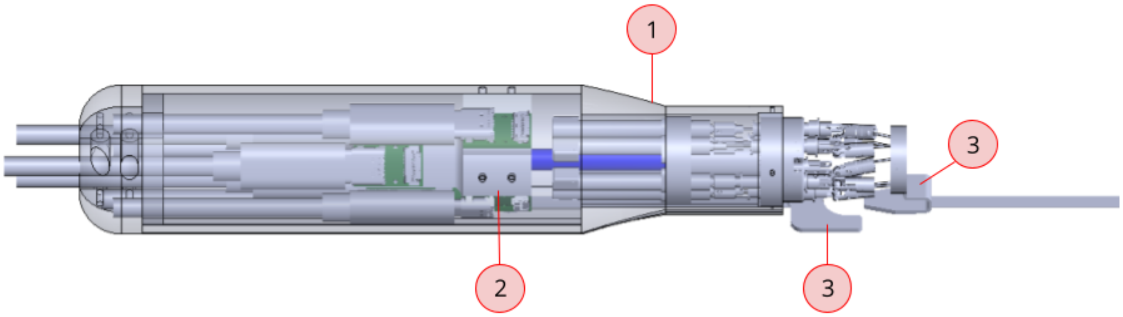


Figure 5.10: CAD of overall assembly with the following new components: 1) Handle, 2) PCB Mount, 3) Mounts for ASAP Optical Markers.

5.3.1 PCB Mount

There are seven PCBs that must fit into the overall design: 1 ASAP PCB and 6 PCBs that each connect to a motor. All of the PCBs connect to flat flexible cables on one end and a USB-C cable on the other. All of the PCBs need to fit in a way that keeps the wires separate and in a compact design to reduce the amount of space taken up in the overall design. The holes in the mount are M1 screw holes, which are necessary to constrain the mount within the overall assembly. Figure 5.11 shows a CAD model of the PCB mount with no mounted PCBs on the left, and with mounted PCBs on the right. The mount was 3D-printed.

(a)

(b)

Figure 5.11: PCB mount CAD drawings: a) PCB mount without PCBs, b) PCB mount with PCBs.

The many screw holes in this part were found to break apart easily when tested, so metal screw threads were melted into each hole to interface with the screws. As shown in Figure 5.10, the mount attaches to the wire guide pipe at the proximal end of the manipulator. Connecting a 3D-printed mount to the manipulator with epoxy seemed inadvisable, so mount is instead epoxied to a metal tube that slides tightly over the wire guide pipe. This constrains the mount in all degrees of freedom except rotation around the pipe's axis. To constrain that final DoF, it connects to the handle with four screws. The mount is shown attached to the manipulator in Figure 5.12.

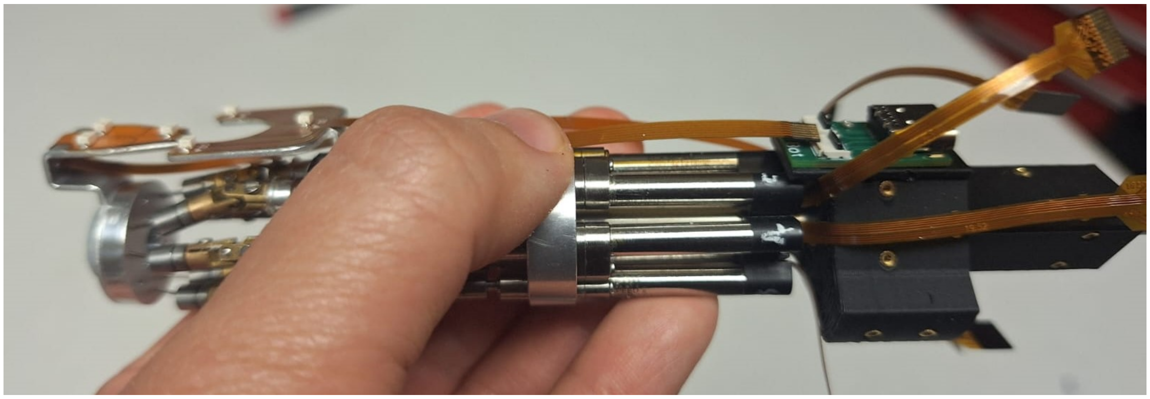


Figure 5.12: PCB and ASAP Mounts connected to the manipulator.

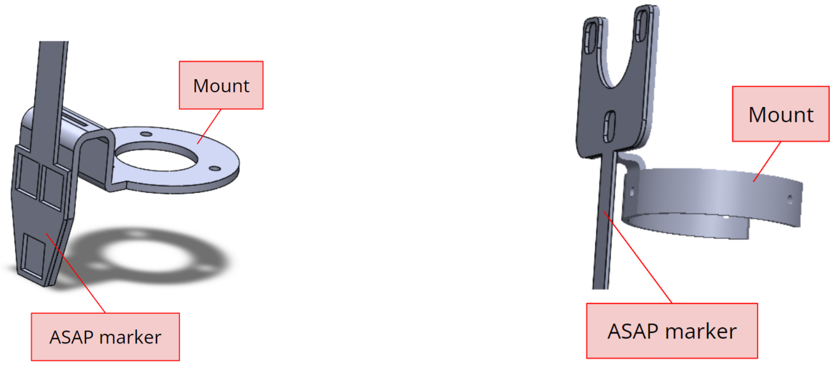
5.3.2 Mounts for the ASAP Optical Markers

ASAP optical markers contain infrared LEDs that are monitored by the ASAP camera. When looking at the overall assembly in Figure 5.10, they are located on the bottom right.

There were three main design considerations for the ASAP optical marker mounts. First, the ASAP mount for the top platform needs to be lightweight and have a small-footprint so that it does not interfere with the motion of the tooltip. Second, the top mount must have a slit for the flexible-flat cable (FFC) to feed through and return to a PCB at the base of the manipulator. Third, the two mounts need to maintain a distance between the markers comparable to the distance in prior iterations of Micron because that distance was empirically optimized to reduce interference between the LEDs while still providing an adequate workspace.

In the CAD design, we visualized how the ASAP markers and mounts would attach, as can be seen in Figure 5.13. The triangle mount attaches on top of the top platform with three screws. It bends upward before bending down in order to increase the distance between the two markers. It also includes the slit through which its FFC fits. The fork mount attaches to three screw holes on the side of the base body. There was not enough space in its bend for a slit so the triangle FFC is routed around it.

Each of these mounts was ordered from Xometry to be laser-cut from 0.8 mm thick Aluminum and bent into shape. The mounts are shown attached on the parts



(a)

(b)

Figure 5.13: ASAP optical marker CAD drawings for the a) triangle flex marker and mount and b) fork flex marker and mount.

in Figure 5.12.

5.3.3 Handle

The handle is necessary to enclose all the mechanical and electrical components of the overall assembly, with an eye to ergonomics to allow the surgeon to manipulate the tool. The handle was designed in SolidWorks, as can be seen in Figure 5.14. The bottom part of the handle is designed to provide strain relief to the PCBs and USB cords, ensuring the USB-C cables do not tear the USB-C receptacles off of the PCBs. This section is necessarily large to accommodate the seven USB-C cables that must connect to the manipulator. As soon as possible, the handle narrows to provide a better grip. For now, the handle does not cover the universal and ball joints of the manipulator so that it does not interfere with the movement of the top platform and its connected ASAP mount. A closure on the proximal end of the manipulator compresses the USB-C cables as they exit the handle to provide strain relief to the attached motor driver PCBs.

The handle was printed using a resin printer with heat-resistant resin. It attaches to the base body of the manipulator using the same three screw holes as the fork-shaped ASAP mount. It also screws into the PCB mount to provide additional support against jostle independent of the manipulator. It is imperative that jostle of the manipulator inside the handle be kept to a minimum, especially during calibration

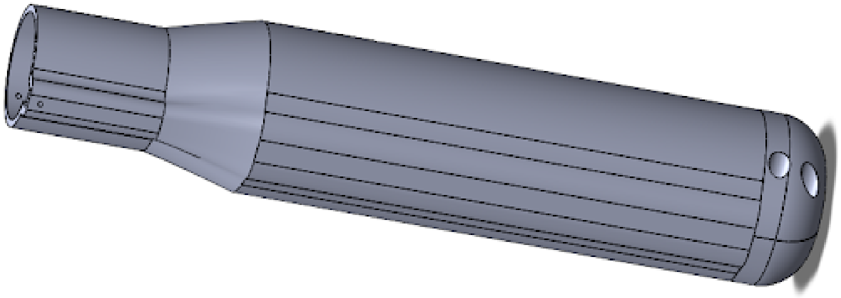


Figure 5.14: CAD drawing of the handle.

of the ASAP sensor, when the manipulator must be able to be mounted and kept in place. A picture of the handle attached to the manipulator is shown in Figure 5.15.

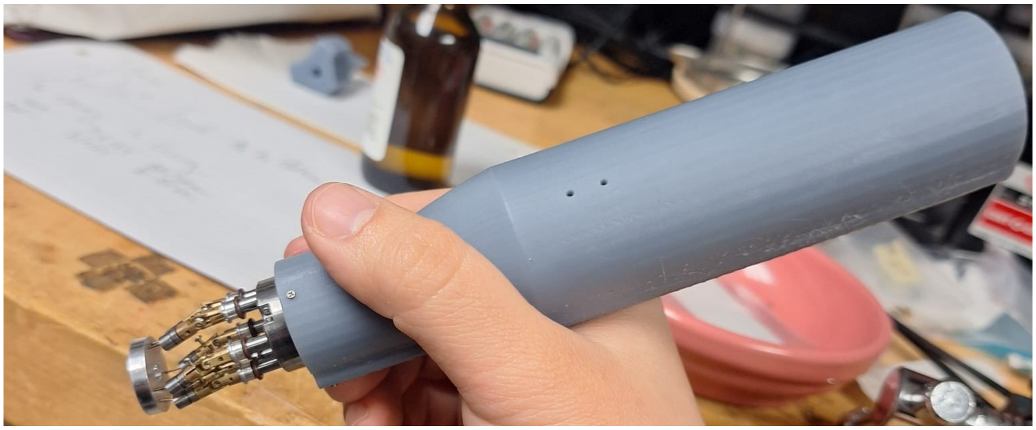


Figure 5.15: The printed handle attached to the manipulator.

5.3.4 Desktop Enclosure

Finally, a desktop enclosure was designed to house all the electrical components of Micron, with the following main components: a power switch, a 12V 30A power supply, a Jetson Nano computer, a fan, the EPOS4 motor drivers, the driver adapter PCBs, the USB-C connections to the motors, a power supply connection for the Jetson Nano, an ethernet connection for the Jetson Nano, and USB connections for the Jetson Nano. The enclosure is shown in Figure 5.16.

In the close-up pictures of Figure 5.17, it can be seen that the driver PCBs are stacked using standoffs, and there are USB-C connections going outside of the

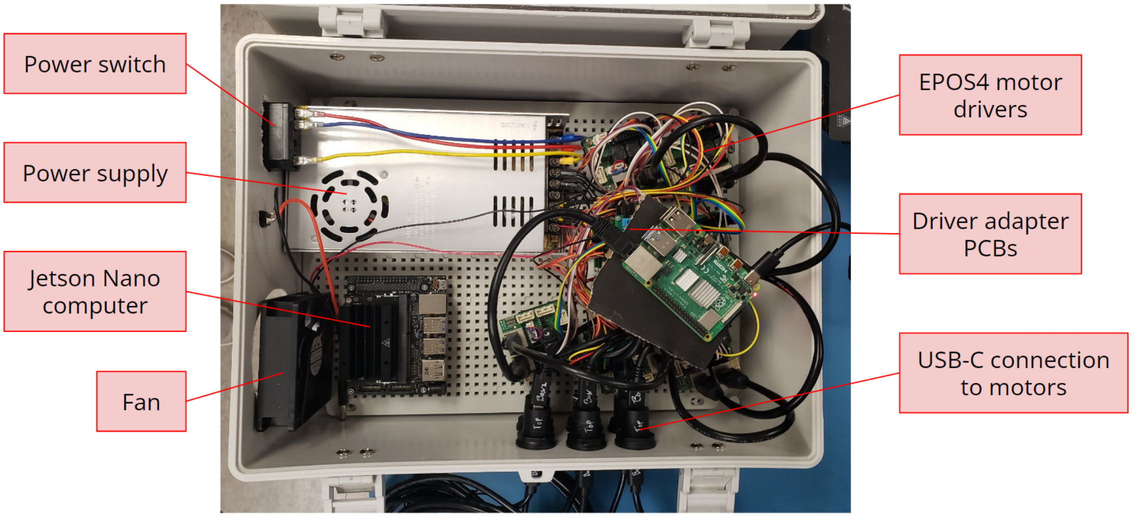
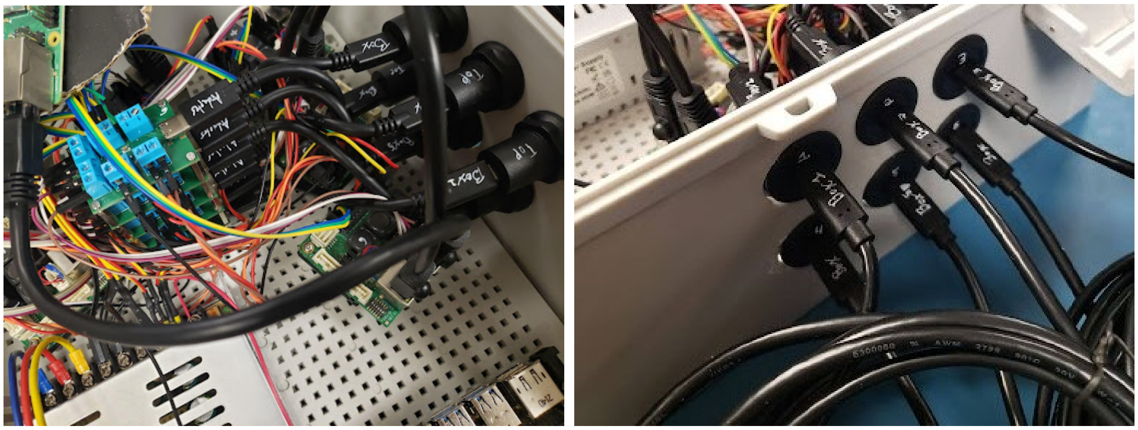


Figure 5.16: Top view of desktop enclosure with labeled main components.

enclosure, which are attached to the motor adapter PCBs mentioned previously. All cables running through the walls of the enclosure are run through connectors mounted in the walls to provide strain relief.



(a)

(b)

Figure 5.17: Close-up pictures of desktop assembly highlighting a) the tower of driver PCBs and b) the USB-C connections to outside of enclosure.

The design of this enclosure was very hasty and the wiring is problematic. The driver adapter PCBs were designed with the idea that they would each be placed in separate boxes with their EPOS4 driver. However, the limited space made that impossible. The wires running between each driver and its adapter PCB are bundled with zip ties as much as possible, but the problem remains. The driver adapter PCBs should be redesigned to connect the USB-C cables to some other connector receptacle, rather than screw hole terminals.

Additionally, the USB-C cables are incredibly rigid, which made fitting them into the small enclosure very difficult. At one point, the strain caused by the small cables inside the enclosure caused a USB-C receptacle to snap off a driver adapter PCB. The best fix to this issue would be to use a floppier cable. For example, the KIST microsurgical device uses RHD ultra thin SPI interface cables from Intan Technologies. Though expensive, these cables are smaller and much more floppy. Floppier cables would also benefit the manipulator itself. Currently, the manipulator is hard to hold because the rigidity of the USB-C cables imposes significant strain on its proximal end. This strain could be alleviated with floppy cables.

5.4 Firmware

The control systems of the manipulator, as well as the ASAP sensor, are run on a National Instruments real-time computer with LabVIEW. This computer, termed hereafter as the NI box, existed from prior iterations of Micron. However, it cannot communicate with the EPOS4 drivers to send commands to the motors and receive status information. The drivers communicate over an EtherCAT daisy chain and require a precise 1 kHz command, which the NI box is slightly too slow to provide. To remedy this, a computer was placed in between the NI box and the EPOS4s. Its purpose is to communicate with the EPOS4s over EtherCAT, buffer commands from the NI box, and send them out at an exact 1 kHz rate. It also sends status information for each motor back to the NI box. It is connected to the NI box over a LAN Ethernet cable, sending and receiving UDP messages. A diagram of these connections is provided in Figure 5.18.

A Raspberry Pi 4 Model B was initially chosen for this computer due to its low cost and the high bandwidth of its Ethernet port. Its Raspbian OS was patched with PREEMPT_RT to make it in real time at 1 kHz. However, tests of the latencies in its execution loop showed that it could not achieve the necessary speed. As a replacement, an NVIDIA Jetson Nano was chosen. There were several reasons for this

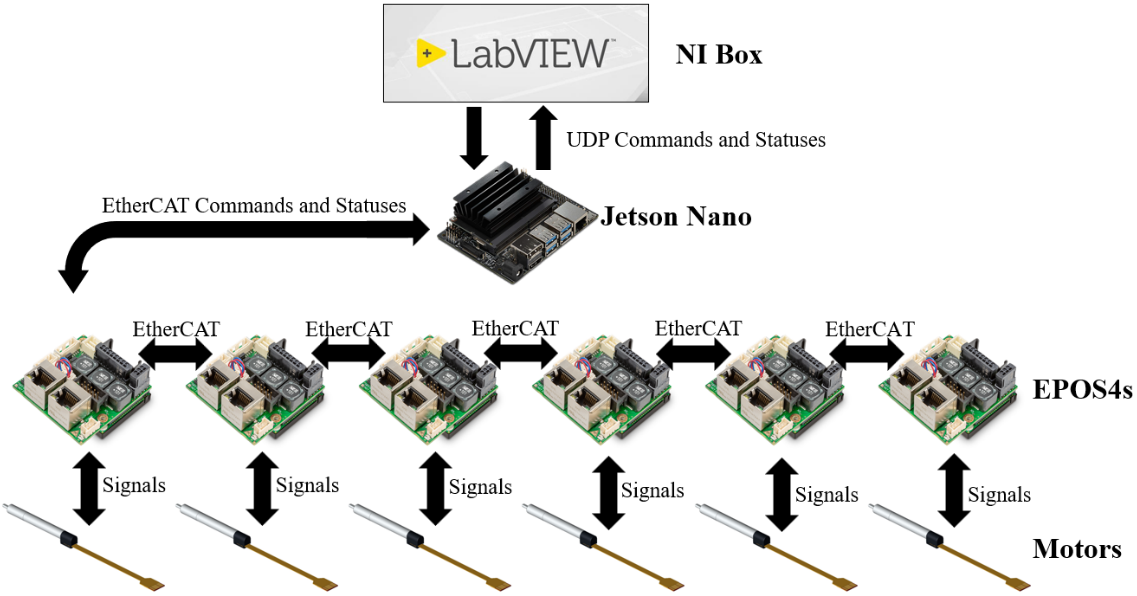


Figure 5.18: Diagram of the communication from the NI box to the motors.

choice. First, it was similarly low cost. Second, NVIDIA has worked with Concurrent Real-Time to produce a version of its real-time RedHawk Linux that operates on the Nano. Concurrent Real-Time generously gave us use of their software for this project for free. Third, its Ethernet port had the same 1 GB/s bandwidth as the Raspberry Pi. Fourth, the significant computing abilities of the Nano enables the development of additional control methods using machine learning, as in [95]. It would also enable implementing computer vision algorithms like [87] that have already been used in prior versions of Micron directly on the Nano.

5.4.1 EPOS4 Device Driver

To implement EtherCAT communication with the EPOS4s, we used LIRMM's open-source ethercat-cpp C++ package for low-level functions [63]. This package, built into their larger PID framework, includes code dedicated to controlling the same EPOS4 drives used in our project. The original code we worked from controlled a single motor to a single target position. This code was adapted to control all six motors to separate target values that are read from shared memory. The code also receives status bytes and position, velocity, and torque readings back from the EPOS4s, which it writes to shared memory. Finally, the code has the option to log all its commands and statuses.

The EPOS4 device driver is run on one of the Nano's 4 cores by a scheduling process. Specifically, this process invokes RedHawk Linux's Frequency-Based Scheduler to run the driver at 1 kHz. Every millisecond, the driver reads the shared memory, sends the commands to the EPOS4s, reads their statuses, and writes those statuses to the shared memory. To verify the real-time function of the code, a GPIO pin on the Nano was toggled every cycle. The pin was monitored with a logic analyzer, which showed max scheduler error of just ±15 ns, more than sufficient for the application.

5.4.2 NI Box Device Driver

The `taskset` command is used to run the NI box device driver on a separate process. It is not scheduled like the EPOS4 Device Driver. Instead, it executes a blocking read on a UDP port for a command from the NI box, reads the newest command as soon as it arrives, writes it to shared memory, reads the status from shared memory, and writes that status to a separate UDP port to send it back to the NI box. As commands and statuses are passed through this driver, their units are converted. Again, the code has the option to log all its commands and statuses.

5.4.3 Controlling the Motors with the Jetson Nano

Tests have been run to control all six motors directly from a command line interface (CLI) on the Nano. Since the Nano and EPOS4s do not provide motion profiling, it is inadvisable to operate the motors in cyclic position mode. Instead, they are operated in cyclic velocity mode, with velocities being commanded to the motors every millisecond. The CLI is useful for situations where we want the motors to move into a specific pose and want to use our own eyes as feedback.

5.5 Calibration of the ASAP Optical Sensor

With every new manipulator, the ASAP optical sensor must be calibrated so that it learns the pose of the infrared LED markers when the manipulator is in a "null+z" pose. The null pose of the manipulator is its position when the top platform is located directly above the base platform, parallel to the base, and with no yaw rotation about the central axis of the manipulator. In the null pose, the top platform is generally regarded to be at the center of its axial translational workspace. The null+z pose simply adds a z-offset to the null pose. A fixture, shown attached in Figure 5.19, was resin printed to hold the manipulator in the null+z pose. The calibration fixture attaches to the base platform with screws and fits snugly over a part of the top platform. The fit over the top platform includes a flat edge to constrain any yaw rotation.

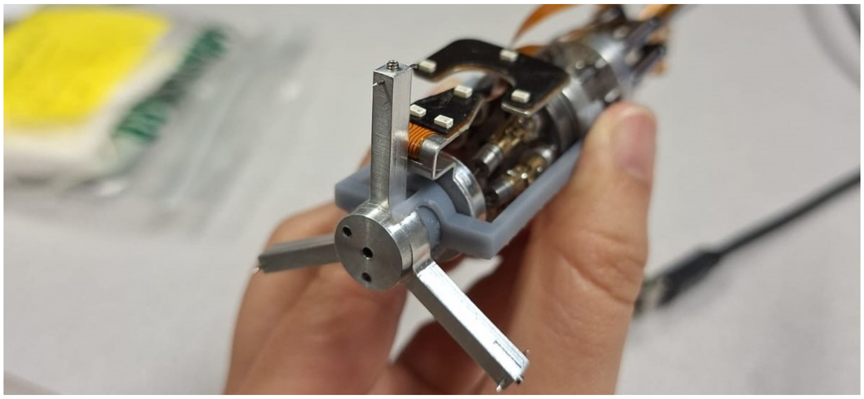


Figure 5.19: Calibration fixture and metal triad attached to the manipulator to keep it in the null+z pose.

A metal triad is fit tightly on top of the calibration fixture, with its yaw rotation constrained by the fixture. This triad includes a sharp needle point at each of its ends. The calibration procedure cycles counter-clockwise through each of these needle points. Micron is clamped in place within ASAP's workspace, as shown in Figure 5.20. A separate needle is clamped into place touching the tip of one of the triad's needles. The pose of the ASAP markers is recorded. Micron is moved to a separate pose, but is kept touching the same unmoved needle. After the pose of the ASAP markers has been recorded in 6 different manipulator poses, the needle is moved to touch a new triad needle point and 6 more poses are recorded. The same is done for the last needle point. Using the hard-coded kinematics of the manipulator, the NI box optimizes for the homogeneous transformation between the markers and the base and top platform.

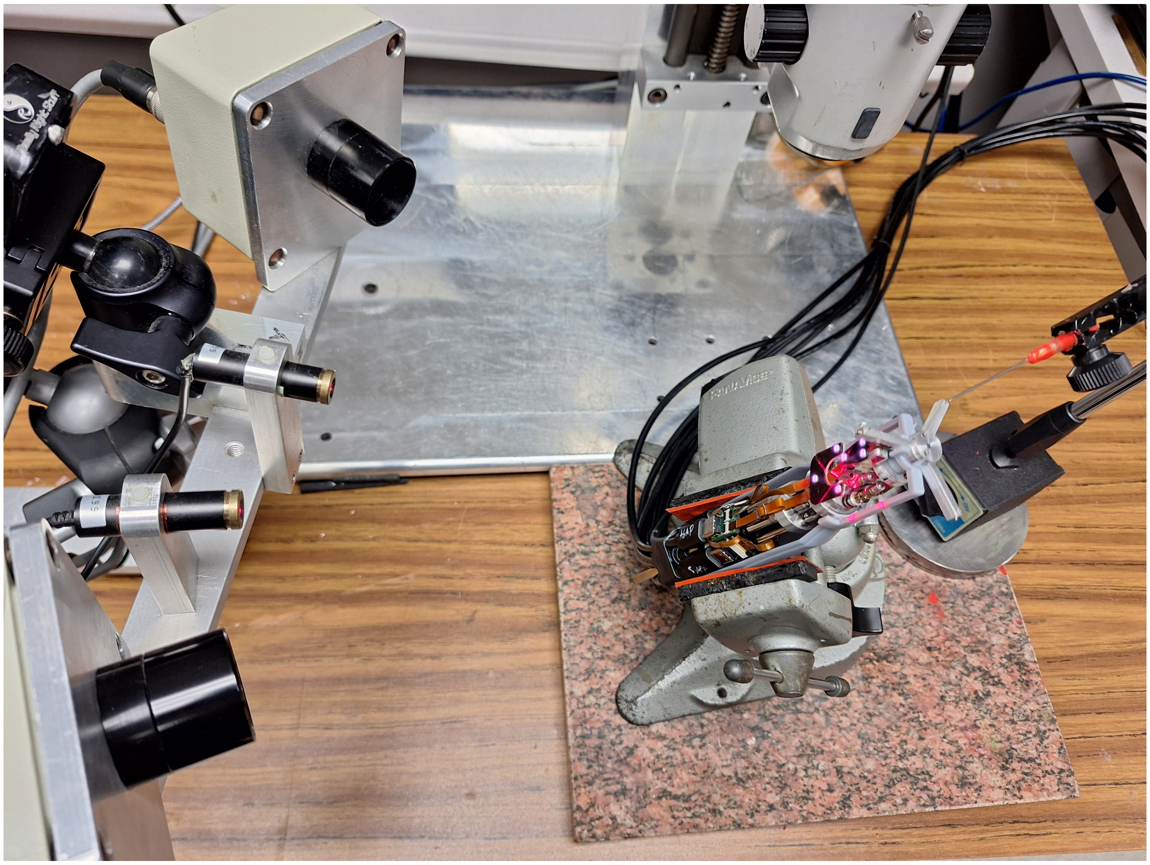


Figure 5.20: Calibration setup. Micron is fixed in the null+z pose and clamped into ASAP's workspace.

5.6 Moving the Motors and Analyzing Feedback from the ASAP Optical Sensor

Once the manipulator was calibrated, its inverse kinematics were implemented on the NI box. The control software on the NI box takes in measurements of the 6-DoF poses of the handle and end-effector from ASAP, defines a goal pose of the end-effector, and uses inverse kinematics to determine the changes required in the positions of the prismatic joints. The motors are commanded with velocity commands from the NI box. Velocity control allows more of the controller to be built and edited in the NI box and does not require a homing method for the motors, which would be difficult to implement on such a small device.

With the inverse kinematics implemented, a test was run to verify their operation. The NI box measures the poses of the handle and end-effector with ASAP and runs the inverse kinematics to estimate the linear positions of each of the motors. Each of the motors was moved to up and down consecutively and the NI box's estimation of how the motors had moved was compared to the motors' encoder measurements. The results are shown in Figure 5.21. Unfortunately, this test showed early on that these estimates are not currently reliable.

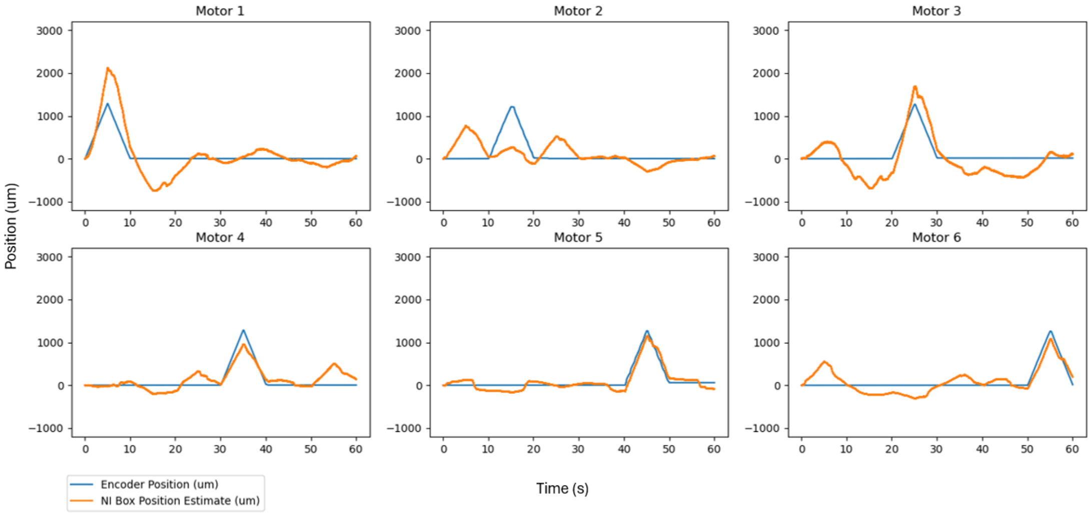


Figure 5.21: Plots of the estimated position of each motor while each of the motors is commanded to moved in turn.

The NI box's estimates of the motor positions roughly follow the encoder readings for all motors except motor 2. However, even when the trend is visible, there is still significant error. The implementation of the inverse kinematics and the calibration software was thoroughly debugged prior to these results. The kinematic data that was entered to define the kinematic structure of the manipulator was also compared against measurements taken on the physical manipulator, but no significant deviations were discovered. There is one likely culprit: a distinct wobbly behavior in the manipulator.

During ASAP calibration, it was noted that the assembled manipulator was far more wobbly than expected. Specifically, one can move the top platform without moving any of the prismatic joints. Ideally, the positions of the six prismatic joints should constrain the top platform to one pose. As soon as we noticed this issue, we reached out to the ANSUR Laboratory at KIST to see if they had seen the same problem. They sent back a video showing essentially no wobble. The video also showed that the nitinol wires they used to connect their top platform to their ball joints is shorter than ours. An error in the dimensions given to us for this wire led us to cut these wires too long. Since the wires exit the top platform at an outward angle, longer wires increase the effective kinematic diameter of the top platform, which forces the struts to lean at a much more shallow angle, so that they are almost vertical. A kinematic singularity exists in a 6-PUS manipulator when its legs are vertical [46]. Proximity to this singularity allows the top platform to wobble independent of its prismatic joints. Since the top platform can wobble when its prismatic joints are fixed, the inverse kinematics are no longer a function with a single solution. Since they are implemented in a way that assumes only a single solution exists, their output cannot be trusted with the current mechanical structure of the manipulator.

Chapter 6

Conclusion

6.1 Contributions

This thesis demonstrated the construction of a new iteration of Micron with more reliable and stiffer motors. The design is based off of the KIST microsurgical robot and then integrated with existing Micron control software and the ASAP optical sensor, for more accurate and informational pose measurements. Hardware delays and budget constraints often characterized this project. During those delays, the applicability of a parallel continuum manipulator's design to Micron was investigated. The kinematics and dynamics of a representative manipulator was simulated. The stiffness of various possible designs for Micron was evaluated and ultimately the concept was proven infeasible.

During further delays, the first steps toward a full kinematics and dynamics simulation of Micron were taken. Two simulations, one in Gazebo and the other in PyBullet, were created. Although the lack of a physical Micron made it impossible to accurately simulate the manipulator's dynamics, its kinematic performance was evaluated in both simulations, with Gazebo displaying the best performance at Micron's small scale. A heavily simplified version of Micron in PyBullet was automated to pick up a small object on a model retina, the first step toward automating a surgical sub-task.

The hardware arrived just in time to enable the physical construction of the next iteration of Micron. The 82 parts of the manipulator were assembled carefully with epoxy and screws. Auxiliary mechanical parts were designed and constructed to hold PCBs and house the electronics. Custom PCBs were designed, assembled, and tested. Firmware was written to connect the motors to existing control software and the ASAP optical sensor in the lab. ASAP was calibrated for the new manipulator. The manipulator was moved and its pose was measured with ASAP.

6.2 Future Work

In Micron's current design, no more work regarding the application of parallel continuum manipulators to Micron is required. Our analysis showed that their compliance makes them uniquely ill-suited for the current design of Micron. Whenever the parallel continuum manipulators we examined were compliant enough to be actuated by the motors to a sufficiently large workspace, they were too compliant to resist side load at the remote center of motion (RCM). This broke two design rules of Micron: that it should have a sufficiently large workspace and that it should not be affected by side load applied at the sclerotomy. Should these design requirements of Micron ever change, parallel continuum manipulators would merit a second look. Specifically, compliance to side load would make the sclerotomy safer by reducing the chances of it tearing. Additionally, studies performed with prior 3-DoF iterations of Micron observed that ophthalmic surgeons often brace their tools against the side of the sclerotomy during a pars plana vitrectomy. This bracing method steadies the tooltip, making its position error steady-state and easy for the surgeon to unconsciously compensate for. Whether such a method would be effective for a 6-DoF Micron remains to be seen.

Significant work remains for the simulation of Micron. Although the kinematics of the Gazebo simulation were more accurate than the PyBullet simulations, larger error still remained. A variety of reasons could account for this, from errors in the CAD model of the manipulator to errors that we see today in the physical design of the manipulator such as the lengths of the nitinol wire that connects to the top platform. Work has already been done to correct errors in the CAD model. Future work will characterize the correct lengths of this wire on the physical manipulator and correct the CAD model. Once this is done, the kinematic simulation could be reevaluated. The lack of a physical prototype of the manipulator during simulation left the dynamics of its joints a mystery. These dynamics could be characterized and incorporated into the simulation. Lastly, our attempt to automate a surgical sub-task had to be so heavily simplified that it strayed far from a realistic simulation of the manipulator. However, the idea remains and could be implemented in a different simulator. The fact that Gazebo does not natively support deformable objects should push future work toward a new simulator that is used specifically for robotic surgery, such as SOFA or CHAI3D. The project also suffered from a lack of information about the retinal interaction forces observed during epiretinal membrane peeling. Future work should begin by characterizing those forces with retinal force sensors inside real eyes.

Work also remains for the construction of Micron. The wobbliness we've observed with the top platform must be fixed. There are three approaches that should be taken to remedy this. First, the nitinol wire that connects the top platform to the ball joints is too long, increasing the effective kinematic diameter of the top platform and making it less stable. These wires should be removed and shortened. Before they are epoxied in place, they should be super-glued and the manipulator's stiffness tested to verify that the new lengths are correct. Second, the universal joints should be tightened with new screws that better resist loosening. Screws with nylon on their tips would provide the friction to better stay in place. Third, preload springs should be added to the manipulator to reduce backlash and wobble. Three springs could be inserted between the base and top platform to pull them into place and stabilize the top.

The seven USB-C cables that connect the manipulator to its enclosure would merit a redesign. These cables were used because of their low cost. However, they have proven far too rigid to use in a handheld manipulator. Several times, the strain they exert has ripped USB-C receptacles off of our PCBs. Additionally, the strain all seven cables exert at the proximal end of the manipulator makes it difficult to hold and manipulate. In the short-term, this can be alleviated by holding the cables up above the manipulator. In the long-term, these cables should be replaced with cables that are smaller and floppier, like the RHD ultra thin SPI interface cables from Intan Technologies. Smaller cables would also allow for a smaller handle. New PCBs would have to be designed for new cables, but this would again be an opportunity. The wiring in the desktop enclosure is messy because screw hole terminals are used to connect the EPOS4 drivers to the driver adapter PCBs. The EPOS4s have connector receptacles on them that should be used in conjunction with connector receptacles on new PCBs to significantly simplify the wiring.

Once the manipulator's wobbly behavior is fixed, a number of procedures must be followed to make it ready for use in experiments. First, ASAP will need to be recalibrated and its updated kinematic data will need to be entered. Even with these updates, the complexity of the small joints allows a significant accumulation of kinematic error. This error could be reduced with a calibration procedure that optimizes the kinematics to fit a collection of poses taken with the actual manipulator. Then, the dynamics of the manipulator can be characterized. The manipulator is controlled with velocity commands to avoid the need to home it, but a recentering algorithm is needed to move the top platform into the null+z pose directly over and parallel to the base. Once this all implemented, work can proceed to validate the workspace of the manipulator and its resistance to side load. Work can also be done to characterize the frequency response of the manipulator. Experiments involving the manipulator working inside some model of the human eye are the eventual target.

Bibliography

[1] URL https://www.intuitive.com/en-us/products-and-services/da-vinci. 2.2

[2] URL https://www.preceyes.nl/. (document), 2.2, 2.2

[3] Luna innovations. os1100 & os1200 fiber bragg gratings. https://lunainc.com/product/os1100-os1200. 4.3.2.5

[4] Aug 2023. URL https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment. 2.1

[5] Wei Tech Ang, PK Pradeep, and CN Riviere. Active tremor compensation in microsurgery. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 2738–2741. IEEE, 2004. (document), 2.4, 2.3

[6] Brian C Becker, Sandrine Voros, Louis A Lobes, James T Handa, Gregory D Hager, and Cameron N Riviere. Retinal vessel cannulation with an image-guided handheld robot. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 5420–5423. IEEE, 2010. 2.1

[7] Brian C Becker, Robert A MacLachlan, Louis A Lobes, Gregory D Hager, and Cameron N Riviere. Vision-based control of a handheld surgical micromanipulator with virtual fixtures. *IEEE Transactions on Robotics*, 29(3):674–683, 2013. 1.1

[8] Nilufer Berker and Cosar Batman. Surgical treatment of central retinal vein occlusion. *Acta ophthalmologica*, 86(3):245–252, 2008. 2.1

[9] Caroline B Black, John Till, and D Caleb Rucker. Parallel continuum robots: Modeling, analysis, and actuation-based force sensing. *IEEE Transactions on Robotics*, 34(1):29–47, 2017. 3.4

[10] Caroline Bryson Black. Modeling, analysis, force sensing and control of continuum robots for minimally invasive surgery. 2017. 1.2, 3.1

[11] Mark S Blumenkranz, Dimitri Yellachich, Dan E Andersen, Michael W Wiltberger, David Mordaunt, George R Marcellino, and Daniel Palanker. Semiautomated patterned scanning laser for retinal photocoagulation. *Retina*, 26(3):

370–376, 2006. 2.1

[12] Johannes Bodner, Florian Augustin, Heinz Wykypiel, John Fish, Gilbert Muehlmann, Gerold Wetscher, and Thomas Schmid. The da vinci robotic system for general surgical applications: a critical interim appraisal. *Swiss medical weekly*, 135(4546):674–674, 2005. 2.2

[13] Jean-Louis Bourges, Jean-Pierre Hubschman, Jason Wilson, Stephen Prince, Tsu-Chin Tsao, and Steven Schwartz. Assessment of a hexapod surgical system for robotic micro-macro manipulations in ocular surgery. *Ophthalmic research*, 46(1):25–30, 2011. 2.2

[14] Dan H Bourla, Jean Pierre Hubschman, Martin Culjat, Angelo Tsirbas, Anurag Gupta, and Steven D Schwartz. Feasibility study of intraocular robotic surgery with the da vinci surgical system. *Retina*, 28(1):154–158, 2008. 2.2

[15] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016. 4.3.2.1

[16] H Logan Brooks Jr. Macular hole surgery with and without internal limiting membrane peeling. *Ophthalmology*, 107(10):1939–1948, 2000. 1.1

[17] Audun Brunes and Trond Heir. Visual impairment and employment in norway. *BMC public health*, 22(1):648, 2022. 2.1

[18] Caroline E Bryson and D Caleb Rucker. Toward parallel continuum manipulators. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 778–785. IEEE, 2014. 3, 3.2.1, 3.2.2.1

[19] Leon A Bynoe, Robert K Hutchins, Howard S Lazarus, and Mark A Friedberg. Retinal endovascular surgery for central retinal vein occlusion: initial experience of four surgeons. *Retina*, 25(5):625–632, 2005. 2.1

[20] Elliot Chane-Sane, Pierre-Alexandre Leziart, Thomas Flayols, Olivier Stasse, Philippe Souères, and Nicolas Mansard. Cat: Constraints as terminations for legged locomotion reinforcement learning, 2024. 4.3.4

[21] Steve Charles. Techniques and tools for dissection of epiretinal membranes. *Graefe's archive for clinical and experimental ophthalmology*, 241(5):347–352, 2003. 2.1

[22] David Y Choi and Cameron N Riviere. Flexure-based manipulator for active handheld microsurgical instrument. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 2325–2328. IEEE, 2006. 1.2, 2.3

[23] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021. 4.2, 4.3.2.1

[24] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016. 4.3.2.1

[25] Marc D de Smet, Thijs CM Meenink, Tom Janssens, Valerie Vanheukelom, Gerrit JL Naus, Maarten J Beelen, Caroline Meers, Bart Jonckx, and Jean-Marie Stassen. Robotic assisted cannulation of occluded retinal veins. *PloS one*, 11(9): e0162037, 2016. 2.1, 2.2

[26] Simon DiMaio, Mike Hanuschik, and Usha Kreaden. The da vinci surgical system. *Surgical robotics: systems applications and visions*, pages 199–217, 2011. 2.2

[27] TL Edwards, K Xue, HCM Meenink, MJ Beelen, GJL Naus, MP Simunovic, M Latasiewicz, AD Farmery, MD De Smet, and RE MacLaren. First-in-human study of the safety and viability of intraocular robotic surgery. *Nature biomedical engineering*, 2(9):649–656, 2018. 2.2

[28] Mojtaba Esfandiari, Ji Woong Kim, Botao Zhao, Golchehr Amirkhani, Muhammad Hadi, Peter Gehlbach, Russell H Taylor, and Iulian Iordachita. Cooperative vs. teleoperation control of the steady hand eye robot with adaptive sclera force control: A comparative study. *arXiv preprint arXiv:2312.01631*, 2023. 2.2

[29] Howard F Fine, Wei Wei, Roger E Goldman, and Nabil Simaan. Robot-assisted ophthalmic surgery. *Canadian Journal of Ophthalmology*, 45(6):581–584, 2010. 3

[30] Pierre-Antoine Forest. World's first clinical stage bi-manual ophthalmology robot successfully completes retinal surgery case, Mar 2024. URL `https://www.medtechdive.com/press-release/20240307-worlds-first-clinical-stage-bi-manual-ophthalmology-robot-successfully-` 2.2

[31] Robert N Frank. Retinal laser photocoagulation: Benefits and risks. *Vision Research*, 20(12):1073–1081, 1980. 2.1

[32] Andy Gijbels, Niels Wouters, Peter Stalmans, Hendrik Van Brussel, Dominiek Reynaerts, and Emmanuel Vander Poorten. Design and realisation of a novel robotic manipulator for retinal surgery. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3598–3603. IEEE, 2013. 2.2

[33] Kurt Gillis. Medicare physician payment schedule services for 2001-a summary of claims data. In *Physician Marketplace Report*, page 22. American Medical Association, Chicago, Illinois, 2003. 2.1, 4.3.1

[34] Aida Kafai Golahmadi, Danyal Z Khan, George P Mylonas, and Hani J Marcus. Tool-tissue forces in surgery: A systematic review. *Annals of Medicine and Surgery*, 65:102268, 2021. 2

[35] Puneet K Gupta, Pahick S Jensen, and Eugene de Juan. Surgical forces and tactile perception during retinal microsurgery. In *Medical Image Computing and*

Computer-Assisted Intervention–MICCAI'99: Second International Conference, Cambridge, UK, September 19-22, 1999. Proceedings 2, pages 1218–1225. Springer, 1999. 4.3.1

[36] Xu Han and Yang Yang. Customer requirements for a vitreoretinal robot. In *2014 International Conference on Management Science and Management Innovation (MSMI 2014)*, pages 113–118. Atlantis Press, 2014. 1.1

[37] Yuqiao Han, Arpita Routray, Jennifer O Adeghate, Robert A MacLachlan, Joseph N Martel, and Cameron N Riviere. Monocular vision-based retinal membrane peeling with a handheld robot. *Journal of Medical Devices*, 15(3): 031014, 2021. 1.1

[38] Changyan He, Ali Ebrahimi, Marina Roizenblatt, Niravkumar Patel, Yang Yang, Peter L Gehlbach, and Iulian Iordachita. User behavior evaluation in robot-assisted retinal surgery. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 174–179. IEEE, 2018. (document), 2.1

[39] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. `https://github.com/hill-a/stable-baselines`, 2018. 4.3.2.1

[40] Lee F Hotraphinyo and Cameron N Riviere. Precision measurement for microsurgical instrument evaluation. In *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 4, pages 3454–3457. IEEE, 2001. 2.3

[41] Daniel Ingram. ros_sdf. `https://github.com/daniel-s-ingram/ros_sdf`, 2019. 4.2.1

[42] Iulian Iordachita, Zhenglong Sun, Marcin Balicki, Jin U Kang, Soo Jay Phee, James Handa, Peter Gehlbach, and Russell Taylor. A sub-millimetric, 0.25 mn resolution fully integrated fiber-optic force-sensing tool for retinal microsurgery. *International journal of computer assisted radiology and surgery*, 2009. 4.3.2.5

[43] Iulian I Iordachita, Marc D De Smet, Gerrit Naus, Mamoru Mitsuishi, and Cameron N Riviere. Robotic assistance for intraocular microsurgery: Challenges and perspectives. *Proceedings of the IEEE*, 110(7):893–908, 2022. (document), 2.1, 4, 4.3.1

[44] Anirudha D Jagtap and Cameron N Riviere. Applied force during vitreoretinal microsurgery with handheld instruments. In *The 26th annual international conference of the IEEE engineering in medicine and biology society*, volume 1, pages 2771–2773. IEEE, 2004. 2, 3.1

[45] Yunsik Jung, Lingfeng Tao, Michael Bowman, Jiucai Zhang, and Xiaoli Zhang. Physics-guided hierarchical reward mechanism for learning-based robotic grasping, 2023. 4.3.3.3

[46] Eunchan Kim, Ingu Choi, and Sungwook Yang. Design and control of fully handheld microsurgical robot for active tremor cancellation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12289–12295. IEEE, 2021. (document), 1.1, 2.4, 2.5, 2.4, 2.4.1, 3.1, 3.3.1, 3.3.2, 4.2.1, 4.3.2.2, 5, 5.6

[47] Win Tun Latt, U-Xuan Tan, Cheng Yap Shee, Cameron N Riviere, and Wei Tech Ang. Compact sensing design of a handheld active tremor compensation instrument. *IEEE sensors journal*, 9(12):1864–1871, 2009. (document), 2.4, 2.3

[48] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 3.2.2.3

[49] Soo-Chul Lim, Hyung-Kew Lee, and Joonah Park. Grip force measurement of forceps with fibre bragg grating sensors. *Electronics Letters*, 50(10):733–735, 2014. 4.3.2.5

[50] Lynne Loh, Mallika Prem-Senthil, and Paul A Constable. A systematic review of the impact of childhood vision impairment on reading and literacy in education. *Journal of Optometry*, 17(2):100495, 2024. 2.1

[51] Elizabeth A Lundeen, Zeb Burke-Conte, David B Rein, John S Wittenborn, Jinan Saaddine, Aaron Y Lee, and Abraham D Flaxman. Prevalence of diabetic retinopathy in the us in 2021. *JAMA ophthalmology*, 141(8):747–754, 2023. 2.1

[52] Robert A MacLachlan and Cameron N Riviere. High-speed microscale optical tracking using digital frequency-domain multiplexing. *IEEE transactions on instrumentation and measurement*, 58(6):1991–2001, 2008. 1.1, 3.2.4, 5

[53] Robert A MacLachlan, Brian C Becker, Jaime Cuevas Tabarés, Gregg W Podnar, Louis A Lobes, and Cameron N Riviere. Micron: an actively stabilized handheld tool for microsurgery. *IEEE transactions on robotics*, 28(1):195–212, 2011. (document), 1.1, 1.1, 1.2, 2.4, 2.3, 3.4

[54] Robert A. MacLachlan, Brian C. Becker, Jaime Cuevas Tabares, Gregg W. Podnar, Louis A. Lobes, and Cameron N. Riviere. Micron: An actively stabilized handheld tool for microsurgery. *IEEE Transactions on Robotics*, 2012. 4.3.1

[55] Robert E MacLaren, Thomas Edwards, Kanmin Xue, Matthew Simunovic, Thijs C Meenink, Maarten J Beelen, Gerrit Naus, and Marc D De Smet. Results from the first use of a robot to operate inside the human eye. *Investigative Ophthalmology & Visual Science*, 58(8):1185–1185, 2017. 1.1

[56] S Miller, T Soares, YV Weddingen, and J Wendlandt. Modeling flexible bodies

with simscape multibody software. *An Overview of Two Methods for Capturing the Effects of Small Elastic Deformations, Technical Paper, MathWorks*, 2017. 3.2.3

[57] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013. URL https://arxiv.org/abs/1312.5602. 4.3.3.1

[58] Amit P Mulgaonkar, Jean-Pierre Hubschman, Jean-Louis Bourges, Brett L Jordan, Christopher Cham, Jason T Wilson, Tsu-Chin Tsao, and Martin O Culjat. A prototype surgical manipulator for robotic intraocular micro surgery. In *Medicine Meets Virtual Reality 17*, pages 215–217. IOS Press, 2009. 2.2

[59] Ngoc Duy Nguyen, Thanh Nguyen, Saeid Nahavandi, Asim Bhatti, and Glenn Guest. Manipulating soft tissues by deep reinforcement learning for autonomous robotic surgery. In *2019 IEEE International Systems Conference (SysCon)*, pages 1–7. IEEE, 2019. 4

[60] Michael L Nordlund, Daniela MV Marques, Frederico F Marques, Robert J Cionni, and Robert H Osher. Techniques for managing common complications of cataract surgery. *Current opinion in ophthalmology*, 14(1):7–19, 2003. 1.1

[61] Andrew L Orekhov, Caroline B Black, John Till, Scotty Chung, and D Caleb Rucker. Analysis and validation of a teleoperated surgical parallel continuum manipulator. *IEEE Robotics and Automation Letters*, 1(2):828–835, 2016. 3

[62] Benjamin T Ostrander, Daniel Massillon, Leo Meller, Zih-Yun Chiu, Michael Yip, and Ryan K Orosco. The current state of autonomous suturing: a systematic review. *Surgical Endoscopy*, 38(5):2383–2397, 2024. 1.2

[63] Robin Passama and Arnaud Meline. ethercatcpp framework. URL https://ethercatcpp.lirmm.net/ethercatcpp-framework/index.html. 5.4.1

[64] Scott Prahl. Oslo and the eye from ee/phy 448/548. https://omlc.org/classroom/oslotut/eye/index.html, 2018. 4.3.2.3

[65] Cameron N Riviere, Wei Tech Ang, and Pradeep K Khosla. Toward active tremor canceling in handheld microsurgical instruments. *IEEE Transactions on Robotics and Automation*, 19(5):793–800, 2003. 1.1, 2.3

[66] Sophie Rogers, Rachel L McIntosh, Ning Cheung, Lyndell Lim, Jie Jin Wang, Paul Mitchell, Jonathan W Kowalski, Hiep Nguyen, Tien Y Wong, International Eye Disease Consortium, et al. The prevalence of retinal vein occlusion: pooled data from population studies from the united states, europe, asia, and australia. *Ophthalmology*, 117(2):313–319, 2010. 2.1

[67] Marina Roizenblatt, Thomas L Edwards, and Peter L Gehlbach. Robot-assisted vitreoretinal surgery: current perspectives. *Robotic Surgery: Research and*

Reviews, pages 1–11, 2018. 1

[68] Jinan B Saaddine, Amanda A Honeycutt, KM Venkat Narayan, Xinzhi Zhang, Ron Klein, and James P Boyle. Projection of diabetic retinopathy and other major eye diseases among people with diabetes mellitus: United states, 2005-2050. *Archives of ophthalmology*, 126(12):1740–1747, 2008. 2.1

[69] Basil Safwat, Eileen LM Su, Roger Gassert, Chee Leong Teo, and Etienne Burdet. The role of posture, magnification, and grip force on microscopic accuracy. *Annals of biomedical engineering*, 37:997–1006, 2009. 1.1

[70] Roberto Sandoval, Robert A MacLachlan, Michael Y Oh, and Cameron N Riviere. Positioning accuracy of neurosurgeons. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 206–209. IEEE, 2007. 1.1

[71] Khushali Shah, Charles R Frank, and Joshua R Ehrlich. The association between vision impairment and social participation in community-dwelling adults: a systematic review. *Eye*, 34(2):290–298, 2020. 2.1

[72] Amin Abbasi Shahkoo and Ahmad Ali Abin. Autonomous tissue manipulation via surgical robot using deep reinforcement learning and evolutionary algorithm. *IEEE Transactions on Medical Robotics and Bionics*, 5(1):30–41, 2023. 4

[73] Sumit Sharma, Tamer H Mahmoud, and Seenu M Hariprasad. Surgical management of proliferative diabetic retinopathy. *Ophthalmic Surgery, Lasers and Imaging Retina*, 45(3):188–193, 2014. 2.1

[74] Changyeob Shin, Peter Walker Ferguson, Sahba Aghajani Pedram, Ji Ma, Erik P Dutson, and Jacob Rosen. Autonomous tissue manipulation via surgical robot using learning based model predictive control. In *2019 International conference on robotics and automation (ICRA)*, pages 3875–3881. IEEE, 2019. 4

[75] Moshe Shoham, Michael Burman, Eli Zehavi, Leo Joskowicz, Eduard Batkilin, and Yigal Kunicher. Bone-mounted miniature robot for surgical procedures: Concept and clinical applications. *IEEE Transactions on Robotics and Automation*, 19(5):893–901, 2003. 1.1

[76] SPN Singh and CN Riviere. Physiological tremor amplitude during retinal microsurgery. In *Proceedings of the IEEE 28th Annual Northeast Bioengineering Conference (IEEE Cat. No. 02CH37342)*, pages 171–172. IEEE, 2002. 1.1

[77] Angela Sorriento, Maria Bianca Porfido, Stefano Mazzoleni, Giuseppe Calvosa, Miria Tenucci, Gastone Ciuti, and Paolo Dario. Optical and electromagnetic tracking systems for biomedical applications: A critical review on potentialities and limitations. *IEEE reviews in biomedical engineering*, 13:212–232, 2019. 5

[78] Jaimie D Steinmetz, Rupert RA Bourne, Paul Svitil Briant, Seth R Flaxman,

Hugh RB Taylor, Jost B Jonas, Amir Aberhe Abdoli, Woldu Aberhe Abrha, Ahmed Abualhasan, Eman Girum Abu-Gharbieh, et al. Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to vision 2020: the right to sight: an analysis for the global burden of disease study. *The Lancet Global Health*, 9(2):e144–e160, 2021. 2.1

[79] Yuta Taniguchi, Hirotaka Sugiura, Toshiro Yamanaka, Shiro Watanabe, Seiji Omata, Kanako Harada, Mamoru Mitsuishi, Tomoyasu Shiraya, Koichiro Sugimoto, Takashi Ueta, Kiyohito Totsuka, Fumiyuki Araki, Muneyuki Takao, Makoto Aihara, and Fumihito Arai. A force measurement platform for a vitreoretinal surgical simulator using an artificial eye module integrated with a quartz crystal resonator. *Microsystems & Nanoengineering*, 8(74), 2022. 4.3.2.5

[80] Brijen Thananjeyan, Animesh Garg, Sanjay Krishnan, Carolyn Chen, Lauren Miller, and Ken Goldberg. Multilateral surgical pattern cutting in 2d orthotropic gauze with deep reinforcement learning policies for tensioning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2371–2378. IEEE, 2017. 4

[81] Yingzhong Tian, Mingxuan Luan, Xu Gao, Wenbin Wang, and Long Li. Kinematic analysis of continuum robot consisted of driven flexible rods. *Mathematical Problems in Engineering*, 2016, 2016. 3.2.2

[82] John Till and D Caleb Rucker. Elastic stability of cosserat rods and parallel continuum robots. *IEEE Transactions on Robotics*, 33(3):718–733, 2017. 3.3.4

[83] John Till, Caroline E Bryson, Scotty Chung, Andrew Orekhov, and D Caleb Rucker. Efficient computation of multiple coupled cosserat rod models for real-time simulation and control of parallel continuum manipulators. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 5067–5074. IEEE, 2015. (document), 3, 3.1, 3.2.2, 3.2.2.2, 3.2.2.3, 3.2.6

[84] John Till, Vincent Aloi, and Caleb Rucker. Real-time dynamics of soft and continuum robots based on cosserat rod models. *The International Journal of Robotics Research*, 38(6):723–746, 2019. 3.2.6

[85] Angelo Tsirbas, C Mango, and E Dutson. Robotic ocular surgery. *British journal of ophthalmology*, 91(1):18–21, 2007. 2.2

[86] Takashi Ueta, Yoshiharu Yamaguchi, Yoshihiro Shirakawa, Taiga Nakano, Ryuichi Ideta, Yasuo Noda, Akio Morita, Ryo Mochizuki, Naohiko Sugita, Mamoru Mitsuishi, et al. Robot-assisted vitreoretinal surgery: Development of a prototype and feasibility studies in an animal model. *Ophthalmology*, 116(8):1538–1543, 2009. 2.2

[87] Aravind Venugopal, Sara Moccia, Simone Foti, Arpita Routray, Robert A MacLachlan, Alessandro Perin, Leonardo S Mattos, Alexander K Yu, Jody

Leonardo, Elena De Momi, et al. Real-time vessel segmentation and reconstruction for virtual fixtures for an active handheld microneurosurgical instrument. *International Journal of Computer Assisted Radiology and Surgery*, 17(6):1069–1077, 2022. 4, 5.4

[88] Wei Wei, Roger Goldman, Nabil Simaan, Howard Fine, and Stanley Chang. Design and theoretical evaluation of micro-surgical manipulators for orbital manipulation and intraocular dexterity. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 3389–3395. IEEE, 2007. 2.2

[89] Jason R Wilkins, Carmen A Puliafito, Michael R Hee, Jay S Duker, Elias Reichel, Jeffery G Coker, Joel S Schuman, Eric A Swanson, and James G Fujimoto. Characterization of epiretinal membranes using optical coherence tomography. *Ophthalmology*, 103(12):2142–2151, 1996. 2.1

[90] Koen Willekens, Andy Gijbels, Laurent Schoevaerdts, Laure Esteveny, Tom Janssens, Bart Jonckx, Jean HM Feyen, Caroline Meers, Dominiek Reynaerts, Emmanuel Vander Poorten, et al. Robot-assisted retinal vein cannulation in an in vivo porcine retinal vein occlusion model. *Acta ophthalmologica*, 95(3): 270–275, 2017. 2.2

[91] Thomas H Williamson. *Vitreoretinal surgery*. Springer Nature, 2021. 2.1

[92] Jason T Wilson, Matthew J Gerber, Stephen W Prince, Cheng-Wei Chen, Steven D Schwartz, Jean-Pierre Hubschman, and Tsu-Chin Tsao. Intraocular robotic interventional surgical system (iriss): Mechanical design, evaluation, and master–slave manipulation. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 14(1):e1842, 2018. 2.2

[93] Sungwook Yang. *Handheld micromanipulator for robot-assisted microsurgery*. PhD thesis, Carnegie Mellon University, 2015. 2.3, 5.2.2

[94] Sungwook Yang, Robert A MacLachlan, and Cameron N Riviere. Manipulator design and operation of a six-degree-of-freedom handheld tremor-canceling microsurgical instrument. *IEEE/ASME transactions on mechatronics*, 20(2): 761–772, 2014. (document), 1.1, 1.1, 1.2, 2.4, 2.3, 5

[95] Dokyoon Yoon, Eunchan Kim, Ingu Choi, Sung Won Han, and Sungwook Yang. Prediction of voluntary motion using decomposition-and-ensemble framework with deep neural networks. *IEEE Access*, 8:201555–201565, 2020. 5.4

[96] Eric M Young and Katherine J Kuchenbecker. Implementation of a 6-dof parallel continuum manipulator for delivering fingertip tactile cues. *IEEE transactions on haptics*, 12(3):295–306, 2019. (document), 3, 3.1

[97] Haoran Yu, Jin-Hui Shen, Karen M Joos, and Nabil Simaan. Design, calibration and preliminary testing of a robotic telemanipulator for oct guided retinal surgery. In *2013 IEEE International Conference on Robotics and Automation*,

pages 225–231. IEEE, 2013. 2.2

[98] Botao Zhao, Mojtaba Esfandiari, David E Usevitch, Peter Gehlbach, and Iulian Iordachita. Human-robot interaction in retinal surgery: A comparative study of serial and parallel cooperative robots. In *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 2359–2365. IEEE, 2023. (document), 2.3