SCREAM 2.0: Super-elastic Continuum Robot for Endoscopic Articulation and Manipulation

Major Qualifying Project

Submitted By:

Joseph Bartone, Computer Science and Robotics Engineering
Jesse D'Almeida, Robotics Engineering
Andrew Gulotta, Robotics Engineering
Nicholas Pacheco, Robotics Engineering

Project Advisors:

Prof. Loris Fichera
Prof. Gregory S. Fischer
Prof. Haichong Zhang

Worcester Polytechnic Institute

Aug 2019 - May 2020

Abstract

The objective of this project is to investigate robotic platforms to enhance a physician's dexterity in minimally invasive endoscopic surgery. We present the design and construction of a continuum robotic manipulator consisting of two concentric notched tubes capable of providing pan and tilt degrees of freedom. Building the manipulator out of tubes enables the passage of a variety of surgical instruments through the tubes' open lumen. The robot is equipped with a quick-release mechanism that enables hot-swapping of instruments during a procedure. Positional control of the robot is based on its inverse kinematics, which is approximated taking advantage of the geometry of the robot to iteratively and efficiently converge on the desired solution. We report on experimental evidence that verifies the positional accuracy of the robot. Out of the many applications where we believe our robot would be helpful, one where it would help immensely is the endoscopic treatment of laryngeal tumors, which affect 1 in 40 people worldwide at any given time. These tumors are challenging to treat endoscopically due to the lack of articulation in commercially-available surgical instruments. Other potential applications include ear surgery and kidney surgery.

Executive Summary

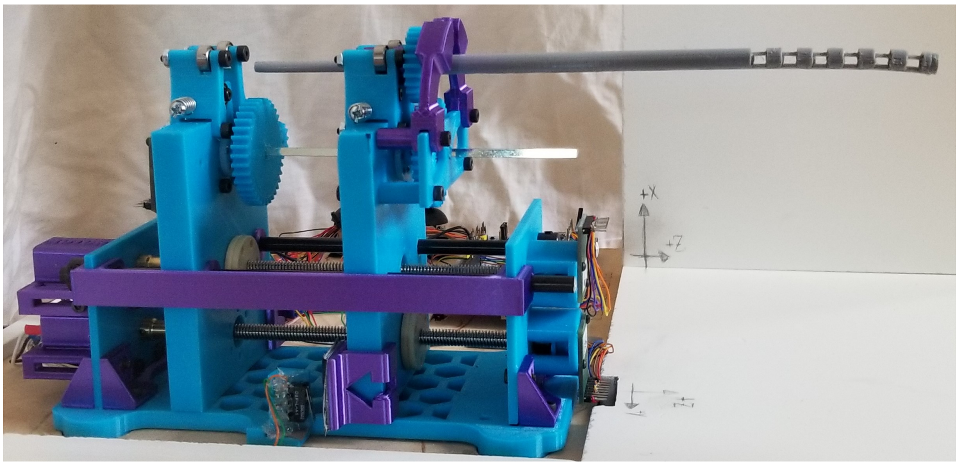
As the cost of surgical care continues to rise, surgeons and engineers are tasked with finding alternative solutions to treat many common medical conditions. Conditions in hard to reach areas have traditionally required complex invasive surgery, however, simpler alternatives are being developed, namely, office-based endoscopic treatment: this option is appealing because it can be simply performed in the clinic with the administration of a topical anesthetic, has low complication rates, and has been shown to enable significant reductions in the cost of care when compared to traditional surgical management [1, 2].

We hypothesize that if physicians had access to steerable endoscopic instruments, then more patients would become eligible to receive treatment in the office and be spared surgery. Motivated by this goal, we have developed a prototype of a novel robotically-actuated steerable manipulator for office-based endoscopic procedures.

In-office endoscopic laser surgery is the main motivating example for this project. We aim to develop a system that can articulate a laser fiber. This was achieved through the development of a continuum end effector with a hollow lumen for passing the laser fiber. An actuation unit controlled by an inverse kinematics algorithm then articulates and steers the tip to the desired location.

The continuum end effector needed to be small enough to fit inside the working channel of an endoscope (2mm diameter) and also enable distal tip bending. Prior research shows that this could be accomplished by making a notched tube wrist made out of a super-elastic nickel-titanium alloy (Nitinol). The notched tube wrist is a hollow tube with notches cut out and a tendon attached to the tip. Pulling this tendon bends the distal end [3, 4]. Further research into this area resulted in our discovery of a different design. The Concentric Agonist-Antagonist Robot (CAAR) design follows the same principles as the notches tube wrist [5]. However, instead of a tendon to pull the tip, this design utilizes a second notched tube of the same notch configuration. This tube fits inside the first outer tube and both are fixed together at the tip. The result is that the backbone of the inner tube acts as the tendon to bend the outer tube. One advantage of this design is that it is capable of bidirectional motion, as pushing the inner tube bends the tip in the opposite direction. We found this design to be advantageous and chose to use it in creating the end effector of our device.

To control the end effector, we iterated over several prototype actuation units. There are three actuator variables to be controlled: translation, rotation, and bending. Research into similar



devices [6, 7] helped to inspire our design. Since our design process was highly iterative, we focused on making parts as modular as possible which assisted in testing quickly and making improvements. The final prototype device with the CAAR tube end effector can be seen in the figure above. The actuation unit centers around two plates, attached to the two tubes that make up the CAAR end effector, that are capable of translating independently on separate lead screws. Translation of the end effector occurs when both plates translate together, and bending occurs when they translate at different rates. Rotation is achieved through gears fixed to the ends of each tube that are driven by motors mounted on the plates.

We also wanted the capability for a user to quickly switch between end effectors. This would allow a user to easily switch to a configuration that best fits the current procedure. We designed a quick-release mechanism, the inspiration for which can be found in [8]. This mechanism consists of a spring that pulls together two sides. These side pieces have bearings to interface with the gear hub of the tube. The springs provide enough force to the bearings to hold the tube down during use, still allowing it to rotate freely, but a user can still pull them apart to release the end effector and switch in a new one.

For the electronic components, we chose to use the STM32 Nucleo 144 embedded board with 12V Micro Metal Gearmotors (Pololu Corporation, Las Vegas, NV) with 298:1 gearboxes. For encoders, we selected hall effect AS5048 High-Resolution Position Sensors, as they had a 14-bit resolution, similar to the resolution of error from [8]. The complete system architecture includes joysticks, the Nucleo board, the actuation unit, and a machine running both MATLAB and our web visualization. Our software includes the embedded code that comprises our robot's Finite State Machine and PID algorithm, our MATLAB inverse kinematics solver, and our Javascript visualization application.

In order to develop an effective controller for the device, we had to derive the forward kinematics of the system. For a continuum system such as this, the process is slightly more complicated than the traditional methodology for rigid link manipulators. We utilized the approach from Webster et al. [9], which breaks up the problem into two parts. The first is a mapping from robot-specific actuator variables to arc parameters (κ, ϕ, s), and the second is a *robot-independent mapping* from arc parameters to the task space. We start by assuming that the tube will bend in a shape that can be described as a sequence of piecewise constant curvature segments, each of which can be described by a set of arc parameters [9] The robot-independent mapping is achieved through an exponential coordinate representation of the rotation and the in-plane translation. The result can be shown by the transformation matrix T where $T \in SE(3)$.

$$T = \begin{bmatrix} \cos(\phi)\cos(\kappa s) & -\sin(\phi) & \cos(\phi)\sin(\kappa s) & \frac{\cos(\phi)(1-\cos(\kappa s))}{\kappa} \\ \sin(\phi)\cos(\kappa s) & \cos(\phi) & \sin(\phi)\sin(\kappa s) & \frac{\sin(\phi)(1-\cos(\kappa s))}{\kappa} \\ -\sin(\kappa s) & 0 & \cos(\kappa s) & \frac{\sin(\kappa s)}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The robot-specific mapping was derived by following [3] and [5]. With this, we map the actuator variables of tube displacement, linear translation, and tube rotation (τ, d, and θ respectively) to the arc parameters of curvature, base rotation, and arc length (κ, ϕ, and s respectively) based on the geometry of the tube.

For solving the inverse kinematics of the system, the general approach is to derive the relationships inverting the forward kinematics. This would consist of the inverse of the robot-independent mapping to go from task space to arc parameters, followed by the inverse of the robot-dependent mapping to go from arc parameters to actuator variables to achieve the specified position. Instead of doing this directly, we propose a numerical solution. This numerical solution is an iterative approach, inspired by how one would manually achieve a target setpoint: by continually bending and translating until the point is reached. Without any base rotation, any tip movement is limited to the x-z plane. A tube translation only moves the tip in the z, and a tube displacement will move it in the x-z. Since translation only affects the z, we can decouple the tip error. The error in the x-direction is attributed to the tube displacement, and the error in the z-direction is attributed to the translation. From there, given the desired position, we can iteratively solve for the tube displacement until the desired x position is reached, and then iteratively solve for the translation until the desired z is reached. When a rotation is required (i.e. the desired point is in three dimensions) the same approach works with a small adjustment. The tube is first rotated by θ about the z-axis so that the desired location lies in the new frame's x-z plane. We then iterate as we did previously to solve for the necessary tube displacement, τ and translation, d.

To test the combination of the hardware and software, we created a test setup in which we could command the device to perform certain tasks and record video of the output from both a top view (y-z plane) and a side view (x-z plane). The video recordings could then be analyzed using Tracker, a video analysis and physics modeling program. Tracker allowed us to set a coordinate frame and track the position of the tip throughout the video. The purpose of our testing was to verify and validate the forward and inverse kinematics, so the results from the Tracker analysis were compared against the expected results from our MATLAB kinematics.

In verifying the forward kinematics, we found the physical system to be significantly off from the calculated results, with an RMS error of 9.72mm. To ensure that the error was not coming from incorrect modeling of our end effector, a separate test setup was devised which utilized a calibrated lead screw and dial to manually bend the end effector. Results from this test were much closer to what was expected, with an RMS error of 1.81mm. This remaining error was determined to be due to each notch not bending evenly, which is assumed in the kinematics. After ruling out the end effector as the main source of error, additional tests were performed on the actuation unit, which revealed that looseness and bending in the front actuation plate due to low tolerances in the 3D printed parts were causing most of the error. We believe that tighter tolerances and metal components would allow for a more rigid system and would eliminate most of this error.

We believe that our robot serves as a sound proof of concept for a tendon-less notched tube robotic manipulator as a surgical platform for office-based endoscopic procedures. Developing a final usable product will require further development and research. More work can be done on improving the kinematics and its ability to work with uneven bending across the end effector. The end effector itself still needs to be created from Nitinol and changes can be made to enable even bending. The actuation unit must be made more rigid and potentially more compact and control of the robot can be improved by adding a handheld controller. With these improvements implemented, we believe that this tool could become invaluable in a physician's in-office procedure workflow.

With the closure of the WPI campus due to the COVID-19 pandemic, we were limited in our manufacturing capabilities for the final prototype. We had originally planned to construct a final robot prototype out of machined metal parts. As a result, the final device was almost entirely 3D printed out of polylactic acid (PLA). Further work on the development of a Nitinol end effector to replace our 3D printed version was also planned. Due to the closure of the on-campus machine shop, we had to continue using the 3D printed version.

ACKNOWLEDGEMENTS

This work was supported by Worcester Polytechnic Institute and the Cognitive Medical Technology and Robotics Laboratory (COMET Lab). We would like to thank our graduate student peers in the COMET Lab, Karim Tarabein, Alex Chiluisa, and Mostafa Atalla, for their comments and suggestions that helped us along with our work.

We would like to thank all of our advisors for their guidance and expertise, especially Professor Loris Fichera for his feedback and editorial work on this paper.

Finally, we would also like to thank the WPI staff members, particularly Ian Anderson from the Washburn Shops and Katherine Crighton from the Robotics Engineering Program, for their assistance throughout our project.

Table of Contents

List of Tables

Table **Page**

List of Figures

CHAPTER 1

INTRODUCTION

As the cost of surgical care continues to rise, surgeons and engineers are tasked with finding alternative solutions to treat many common medical conditions. Tumors of the larynx are a relevant example: these tumors affect more than 1 in 40 people worldwide at any time [12], and can lead to a variety of complications, including speech impairment, difficulty in swallowing and, in more severe cases, respiratory obstruction. The management of laryngeal tumors traditionally involves surgical excision under general anesthesia in the operating room [1]. A simpler alternative was recently developed, namely office-based endoscopic laser treatment: this latter option is appealing because it can be simply performed in the clinic with the administration of a topical anesthetic, has low complication rates, and has been shown to enable significant reductions in the cost of care when compared to traditional surgical management [1, 2].

Despite these benefits, office-based endoscopic procedures are still underutilized because of how challenging they are to perform with available instrumentation. In the specific case of laryngeal procedures mentioned above, the laser fibers used to deliver treatment do not possess any distal articulation, making it impossible to treat patients that present with the disease in ”unfavorable” locations, i.e., locations inside the larynx that cannot be reached with a simple linear path. We hypothesize that if physicians had access to steerable endoscopic instruments, then more patients would become eligible to receive treatment in the office and be spared surgery. Motivated by this hypothesis, in this paper we describe the prototype of a novel robotically-actuated steerable manipulator for office-based endoscopic procedures. This robot combines different design solutions observed in the recent medical robotics literature. We demonstrate kinematic control of our robot, based on a numerical inverse kinematics algorithm. In the following, we first review in detail the motivating example for this project and then describe the design, construction, and validation of our robot.

1.1 Motivating Example: In-Office Laryngeal Procedures

Office-based procedures to treat laryngeal lesions are performed endoscopically, as shown below in Figure 1.1 (a): an endoscope is inserted through the patient's nose (or mouth) and passed down into the throat. The endoscope is equipped with a camera to visualize the workspace, as well as a 2 mm diameter working channel through which a laser fiber is passed. The fiber is used to deliver laser pulses on diseased tissue, a procedure that is repeated until the complete thermal destruction of the target tumor is achieved. Correct laser aiming is vital to ensure a successful outcome; because the laser fibers commercially available for these procedures do not incorporate any bending mechanism, doctors can only control the laser aiming by bending the entire endoscope as seen in Figure 1.1 (a). This greatly limits the maneuverability of the laser fiber, and it also makes the procedure disorienting for the operating physician due to the inability to control the laser aiming without also moving the field of vision. Furthermore, if the tumor is found in an area that requires significant bending, such as in the example in Figure 1.1 (b), then the view from the camera could be obstructed, rendering the procedure impossible to carry out. Patients who present tumors in locations like these are considered inoperable in the office and referred for surgical treatment in the operating room.

The clinical scenario described above provides a motivating example for the work we describe in this document: by enabling independent articulation of the laser fiber, we could one day help reduce the number of laryngeal tumors patients who need to be treated surgically in the operating room. Creating steering instruments at the size required to enable passage through the operating channel of an endoscope (< 2 mm) is challenging, as traditional articulation mechanisms can only be miniaturized down to a certain scale. In this project, we explore mechanisms that

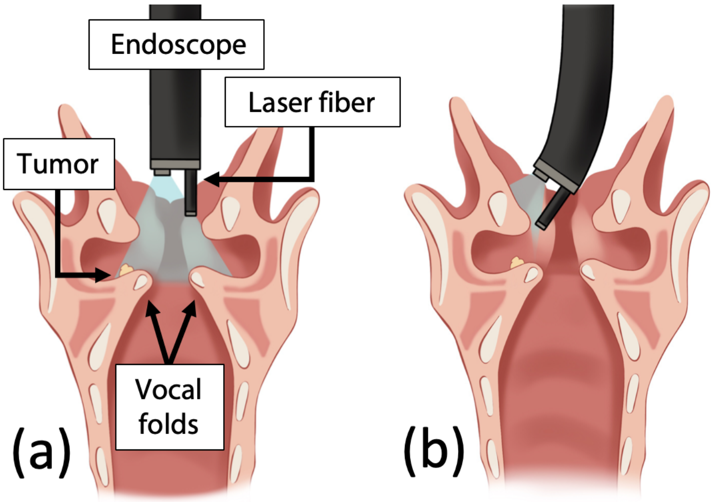


Figure 1.1: *Frontal cut of the larynx: (a) The blue cone represents the field of view of the endoscope; (b) Laser aiming can only be controlled by bending the entire endoscope, which creates limitations in access and field of vision.*

achieve articulation by selectively removing material from the body of a tube, therefore creating a compliant section that can be articulated via the application of a bending moment. We take inspiration from the recent surgical robotics literature, which has seen the introduction of several miniaturized robotic continuum joints.

1.2 Paper Outline

The outline of this paper is as follows:

Design, Modeling, and Construction outlines the functional and nonfunctional requirements that were taken into consideration when designing the device, and details how these requirements were taken into account in the design process. We describe the different design solutions that were evaluated for the construction of the manipulator. In addition, we explain the construction and assembly of the different components of the robot, including end effector fabrication, transmission, circuitry, and software.

Testing and Validation describes experimental work undertaken to verify the kinematic control of the robot. The validation component details the error in position (in mm) across various experimental trials.

Discussion and Future Work offers suggestions for improvements to the robot and describes future inquiry avenues for this project.

CHAPTER 2

DESIGN, MODELING, AND CONSTRUCTION

2.1 Gathering of Requirements

Our team has established a partnership with a physician at a local research hospital that specializes in office-based laryngeal laser procedures. Based on the information we gathered from him through interviews, and from data that was available from the SCREAM 1.0 project [4], we determined that a steerable instrument for endoscopic office procedures would have to have the following essential characteristics:

- **A diameter (Ø 2mm) small enough to transit through the working channel of an endoscope** (see Figure 2.1 for reference).

- **A distal steering mechanism to enable bending independently from the endoscope**.

Furthermore, we established the following non-functional requirements:

- **Motorized actuation system**: Due to the number of independent moving parts required, the system should be motorized to enable intuitive physician control.

- **Use of off-the-shelf components when possible**: Using off the shelf components will help to make the device easily manufacturable and reduce costs.

- **Make parts easy to produce**: Any parts custom parts should be as easy as possible to machine and/or 3D-print.

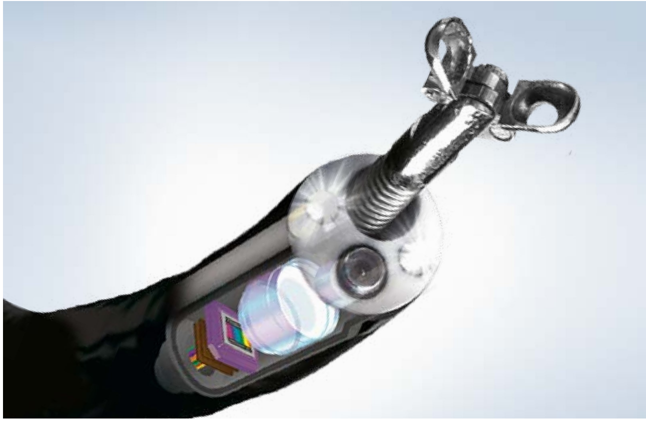


Figure 2.1: *Chip-on-the-Tip endoscope with a working channel used for passing tools such as laser fibers or forceps (shown). Reproduced from [10].*

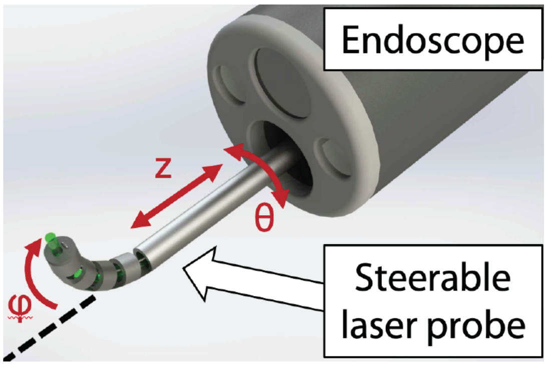


Figure 2.2: *Concept of a notched tube end effector steering a laser fiber through the tool channel of an endoscope.*

2.2 Robot Design

Most robotic devices use traditional rotational or translational joints, however, it can be difficult to construct these joints at the minuscule size needed for use in an endoscopic procedure (recall from the prior section that we are interested in building a manipulator with a diameter < 2 mm). In this project, we explore the use of notched tube cannulas (see Figure 2.2). This type of robotic manipulator can be manufactured at small sizes more easily than traditional joints since manufacturing simply requires the removal of material from a small-diameter tube [3]. Prior research has derived methods to construct these joints out of tubes of nickel-titanium (Nitinol) pulled by a tendon attached at the tip of the tube [3, 4]. Nitinol is a super-elastic alloy, which allows for repeated actuation without fear of plastic deformation and allows the material to recover its original shape once the pull-wire is released. Nitinol is an excellent alloy for use in medical devices and it is available in small diameter tubes that fit our size requirements [2].

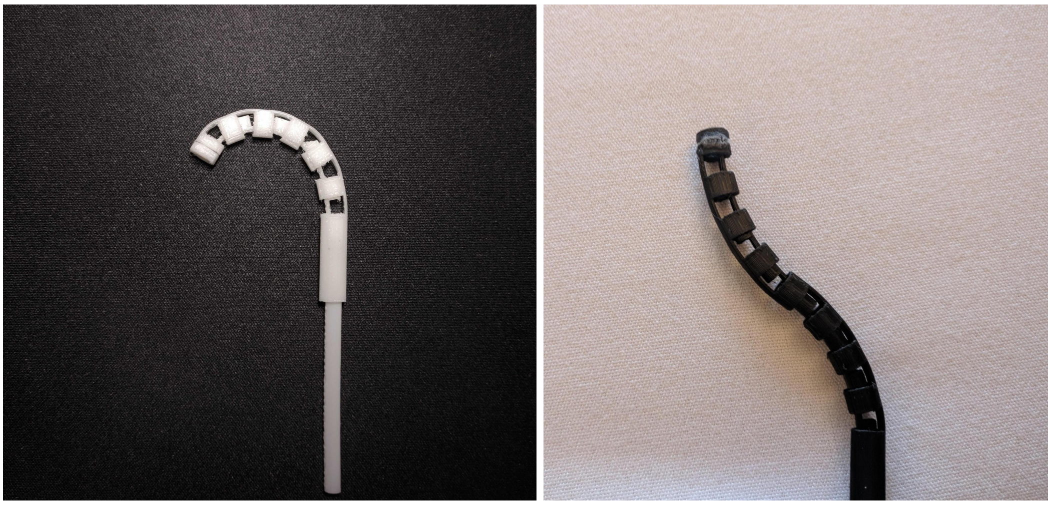


Figure 2.3: *3D printed prototype CAAR tubes bent in the S-curve configuration (left) and standard configuration (right).*

2.2.1 CAAR Manipulator

In exploring the capabilities of notched tube end effectors, we discovered the Concentric Agonist-Antagonist Robot (CAAR) design [5]. The main principle behind a CAAR manipulator is that it contains two concentric notched tubes, oriented with their notches 180 degrees apart, that are connected only at the distal tip. This enables bi-directional bending by pushing/pulling the inner tube. This may be useful in narrow spaces where the physician might normally have to straighten the tube, rotate, and bend again to reach a new location. The CAAR design also allows for different shapes to be achieved by changing the notch pattern. For example, when the notches are flipped 180 degrees for half of the pattern, the result is an S-curve that keeps the distal tip in the same orientation, as seen in Figure 2.3.

2.2.2 Manufacturing of the End Effector

Due to Nitinol's relatively high cost, initial prototyping and design of the end effector were performed with large scale 3D printing. Several rounds of testing were performed before settling on a 3D printing technique that worked well. A simple, single-axis CAAR manipulator with 5mm notches was designed and used for the initial rounds of 3D print testing. This design was printed with several different filament materials to determine which produced the best results for testing.

The first round was printed with Thermoplastic Polyurethane (TPU), which was chosen because of its high flexibility. Unfortunately, the flexibility of the resulting manipulator was too high and the tubes could not maintain their curvature well when experiencing a force in any direction. The next material that was tested was Acrylonitrile Butadiene Styrene (ABS), which is stiffer than TPU but still flexible and durable compared to other plastics. The results of the ABS test were a manipulator with a more ideal stiffness and a cleaner print with less stringing.

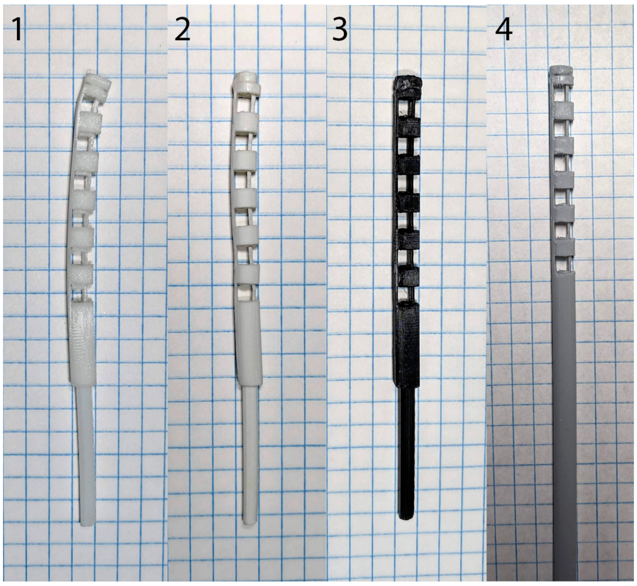


Figure 2.4: *3D printed test end effectors in TPU (1), ABS (2), PLA (3), and PETG (4). The PLA and PETG worked best as they had enough stiffness to spring back to their original shape and did not experience plastic deformation. All examples were printed with a 0.4mm nozzle at a layer height of 0.12mm.*

However, after multiple bending tests, the ABS exhibited high amounts of plastic deformation. The last two materials tested were Polylactic Acid (PLA) and Polyethylene Terephthalate Glycol (PETG). These provided the most promising results. While stiffer than all other materials tested, PLA was still flexible enough at a wall thickness of 1 mm and exhibited no visible plastic deformation. The one downside to PLA is that it is brittle, and too much bending caused the tubes to break. The PETG was as flexible as the ABS manipulator, but experienced less plastic deformation and was much easier to work with as ABS can have difficulties with bed and layer adhesion if the printer does not have a heated enclosure. Examples of prototypes printed in all four materials can be seen in Figure 2.4. After initial testing, all prototype end effector designs were printed out of PLA or PETG. These prints allowed for easy rapid prototyping of notch sizes and configurations, including versions of the s-curve and spiral designs, which worked rather well in 3D printed form.

Upon deciding on a final design for the end effector, initial work began on manufacturing the wrist out of Nitinol. After exploring manufacturing options, three viable methods were determined: laser cutting, wire Electrical Discharge Machining (EDM), and traditional milling. We believed that the laser cutting would have likely produced the best results, however, this would have required the work to be done by a professional laser cutting company as WPI does not possess a laser system capable of cutting Nitinol. After getting a quote it was determined that laser cutting would be too expensive for this project, as this manufacturing step alone would have been already more expensive than the entire project budget ($1,000). This led to laser cutting

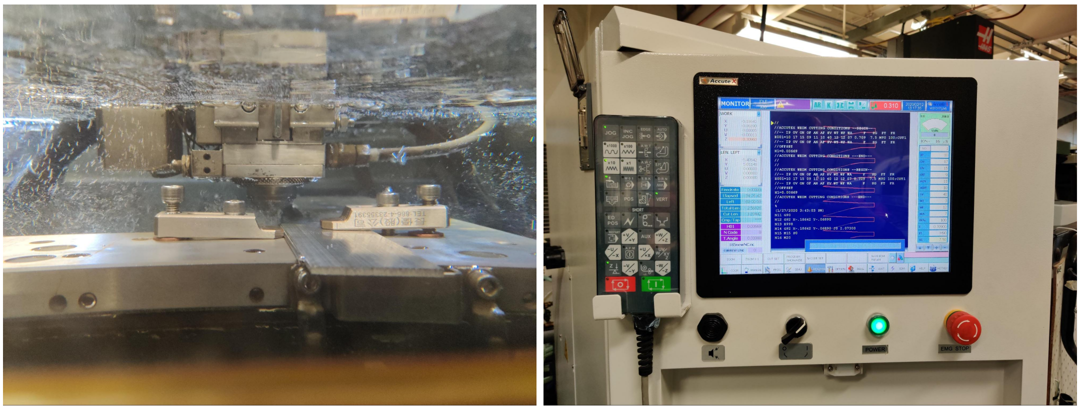


Figure 2.5: *Wire EDM machining setup. Left: Nitinol tube fixture during cutting. Right: G-code display and cutting parameters.*

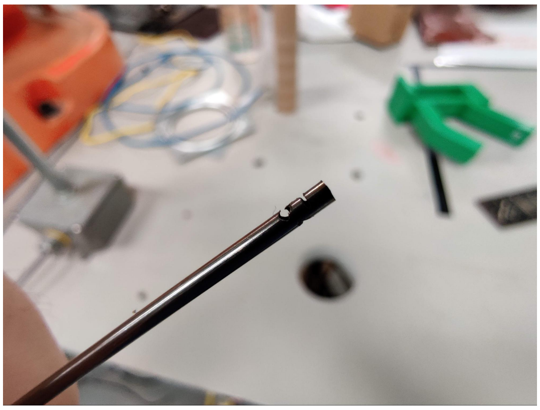


Figure 2.6: *Initial cuts with wire EDM. The first notch is cleanly cut while the second notch was improperly cut due to a failure in the machining fixture.*

being removed from the list of options early on in the manufacturing phase.

Wire EDM was determined to be the next best option, due to the high tolerances listed by the machine manufacturers and the fact that there is a wire EDM machine available for use on WPI's campus. Figure 2.5 shows the setup for the initial Nitinol EDM tests. Overall, EDM machining showed promising results, however, we ran into two main difficulties: fixturing and wire breakage. The first tests explored the use of "super glue fixturing" by using easily available cyanoacrylate glue to secure the Nitinol tube to a bar of aluminum for extra rigidity and ease of mounting in the machine. Unfortunately this method did not prove to be as effective as expected and on multiple occasions the Nitinol tube broke free of the aluminum bar, resulting in failed cuts as seen in Figure 2.6. Further work in exploring options for fixturing for EDM and exploring other machining options is needed to manufacture a proper Nitinol end effector. **Unfortunately for our team, this research into methods to fabricate the Nitinol end effector was cut short by the outbreak of COVID-19 in the United States and the subsequent closure of the WPI campus.**

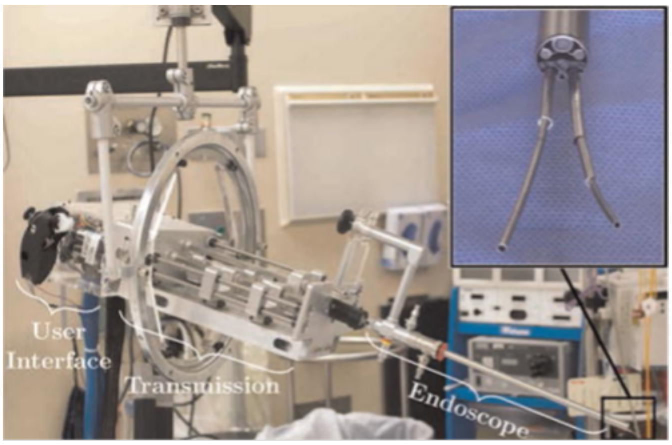


Figure 2.7: *Hand-held robot for transurethral laser surgeries with (top right) a close-up of the endoscope with 2 of the manipulators. Reproduced from [6].*

2.2.3 Design and Construction of the Actuation Unit

To design the actuation unit for our robot, we began by looking at transmissions for similar systems in the surgical robotics literature. In the following, we describe the two designs that inspired our work the most. The first is a hand-held device for transurethral surgery developed by Hendrick et al. [6] (Figure 2.7). This device features two concentric tube manipulators and works in conjunction with an endoscope to perform minimally-invasive procedures on the prostate. Each manipulator is driven by a set of stages. Each stage has its own plate, driven by a lead screw, that each concentric tube is fixed to. The tubes are rotated via gears driven by a square shaft that the plates ride along. The motor to drive the shaft is fixed at the back of the device.

The other device that inspired our work is the skull base surgery developed by Swaney et al. [7]. This robot, shown in Figure 2.8, operates very similarly to the transurethral robot developed by Hendrick et al. [6], however, there are some key differences. The motors that drive the rotation of the tubes are mounted directly to the plates. This means that they move with the plate when it is translated. Also, the tube is driven via a belt from the motors.

In our design, we decided to utilize ideas from each of the systems described above. We used the general layout of the transurethral robot and had the manipulators geared to the motor, as opposed to belted. As in the skull-base surgery robot, we opted for the motors to be mounted to the plates.

2.2.3.1 Hardware Iterations

In this section, we describe the different hardware iterations that we built for the actuation unit of the robot.

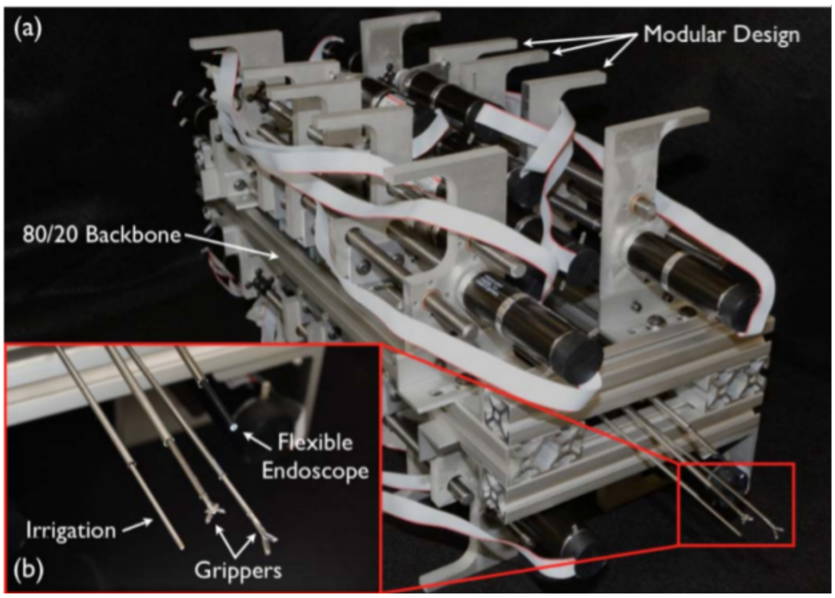


Figure 2.8: *(a) Swaney's quad-manual robot device for skull-base surgery; (b) a close up of the 4 end effectors. Reproduced from [7].*

Version 1: Whisper Version 1 of our robot, nicknamed Whisper, was a proof-of-concept created to verify the general design and gather the base parts. The concept of the robot is shown in Figure 2.9. We initially designed the transmission to actuate three degrees of freedom (DOFs) of each tube independently, for a total of six DOFs. To do this, we created a design that could actuate three DOFs and then repeated it for the second tube. We wanted to make sure everything would function mechanically before investing in any electronics, so knobs were used as placeholders for the motors. The transmission unit is a box containing two plates that the tubes are mounted to. Each plate is attached to a lead screw for translation and a gear train for rotation. As shown in Figure 2.9, these actuating plates are mounted one in front of the other, and the rotation motors are mounted directly to the plates.

The transmission was designed in such a way as to support the easy swapping of different types of end effectors. The idea for a quick-release was inspired by the design described in [8]. An example of the end effector being taken out is shown in Figure 2.10. The quick-release consists of two hinges pulled together by a spring. We inserted a screw through both hinges and the spring. A nut is then secured to the screw, tightening the spring. The hinges clamp onto the hub of the tube's gear, and the force from the springs will keep it from opening. Bearings on the hinges help the tube to rotate freely amongst the pressure from the spring. With this mechanism, a user can easily pull the quick-release apart to take out the end effector and switch in a new configuration that might be better for the current procedure.

Once Whisper was assembled (Figure 2.11), we empirically evaluated its viability and robustness. The two biggest concerns were the quick-release mechanism and the overall stability of the transmission. The spring for the quick-release was mounted about ½" off of the plate. With the

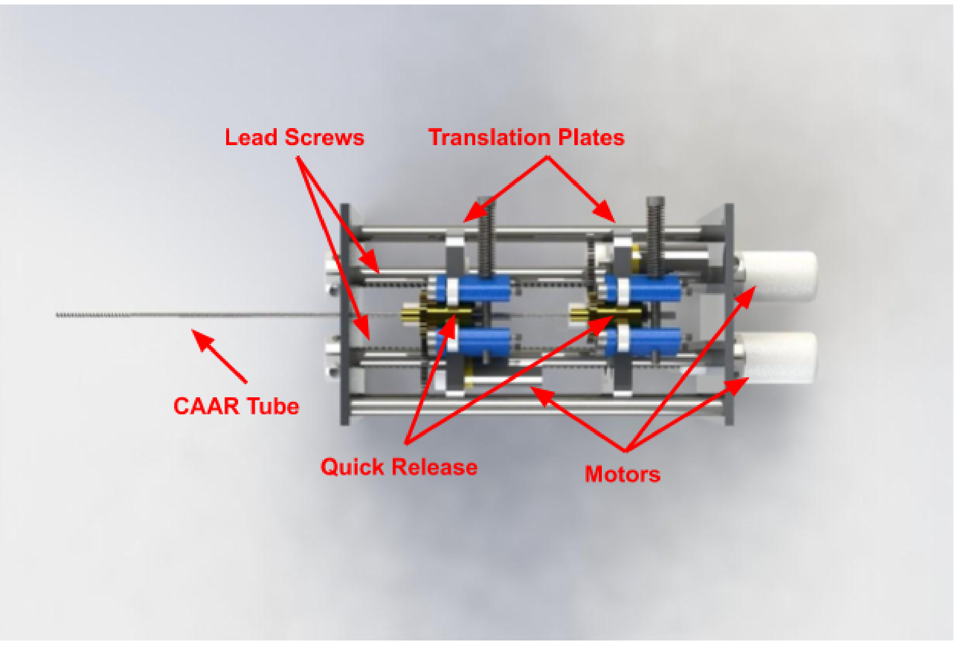


Figure 2.9: *Top view of a Whisper render with labeled lead screws, translation plates, CAAR tube, quick-release, and motors.*

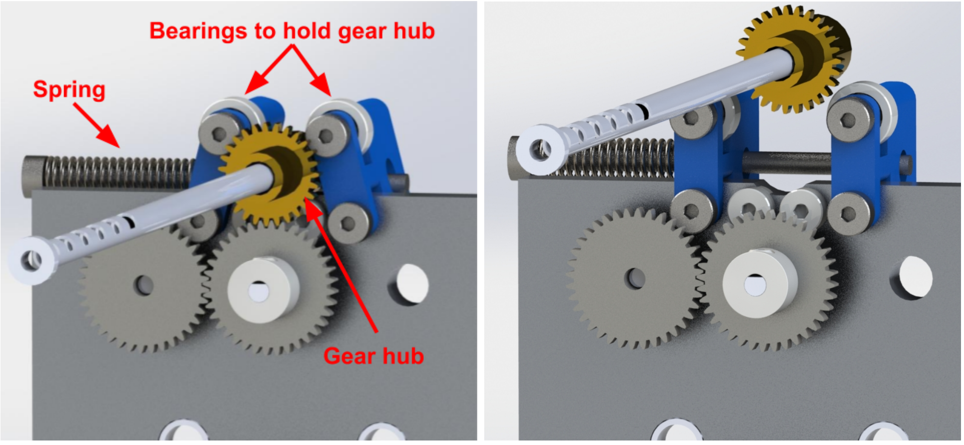


Figure 2.10: *Render of Whisper quick-release with tube in (left) and the tube being taken out (right).*

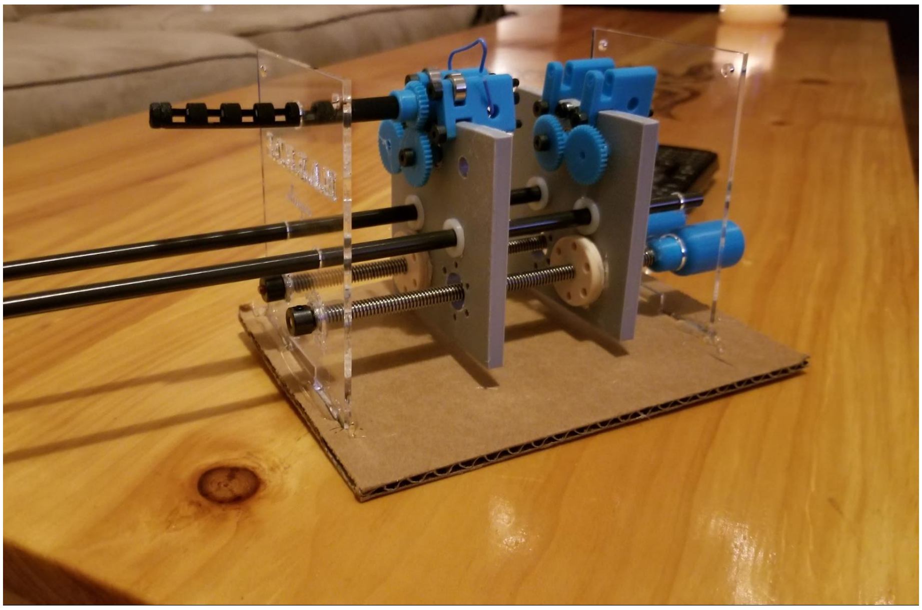


Figure 2.11: *Assembled prototype of Whisper.*

relatively strong force from the spring, instead of clamping down on the tube, it ended up pinching together at the back end. This opened the hinges at the front end reducing its ability to keep the tube in place. We also observed significant slack in the system which caused stability issues. The plates would lean in different directions because the lead screws were mounted off-center. One side of the plate would lag when trying to translate the plate. Several fasteners and brackets were also missing which made other parts unstable. Whisper was tested using a prototype CAAR tube and achieved full manual rotation, translation, and bending.

Version 2: Siren The goal of the second version, nicknamed Siren (Figure 2.12), was to have a testable transmission for initial validation and verification of the kinematics. We had to calculate the necessary torques for the motors and resolution for the encoders because Siren needed to be a fully working prototype. Since the quick-release mechanism did not work in Whisper, it needed to be redesigned.

We began with a list of items that would improve upon Whisper's design. The lead screws in Siren were now stacked vertically in the middle of the plates so that one side would not lag behind the other. A bottom plate was added for the front and backs plates to sit into, with brackets to hold them all together. This made Siren easier to assemble and disassemble for small adjustments. The gears were made thicker with half the pitch and tooth count so that they remained the same size but would mesh better. For the CAAR tube, since the tubes would not rotate independently, a motor driving one of the tubes would be sufficient to rotate the entire end effector. Rotating just the inner tube would cause the outer tube to rotate via the glued end tips. However, since we did not want the glue to undergo any additional stress, both tubes were rotated with the same motor.

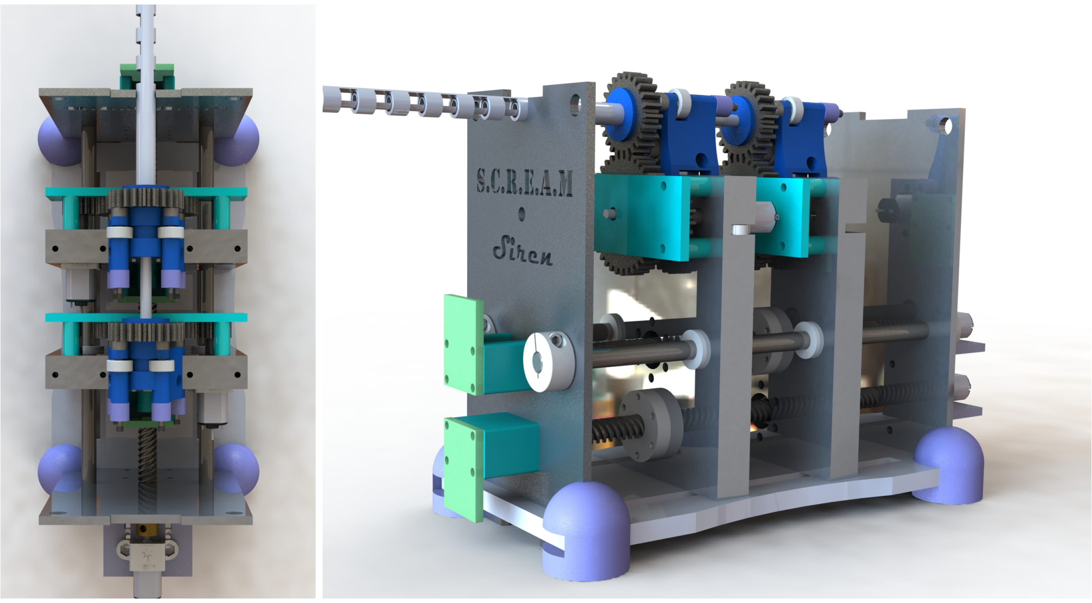


Figure 2.12: *Top view (left) and isometric view (right) of the Siren robot.*

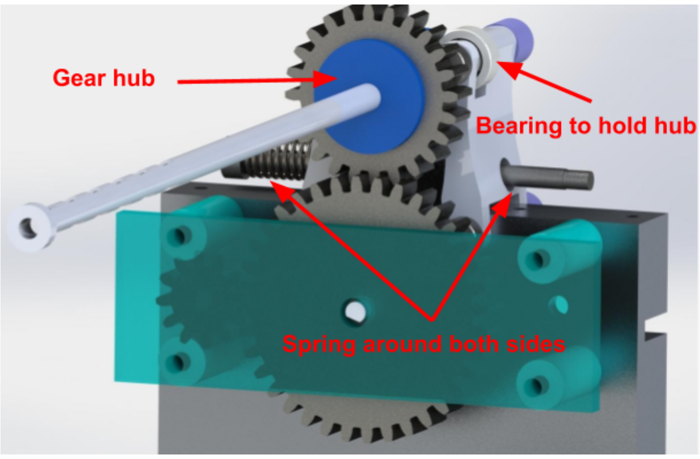


Figure 2.13: *Close up render of quick-release on Siren.*

The shaft of the idler gear from the back plate was extended to the idler gear of the front plate. The front plate would be free to translate along the square shaft.

The new quick-release mechanism, shown in Figure 2.13, was improved by raising a platform for the tube gear to rest on. The actuating plate was also made thicker, which created enough room for the spring and screw to be mounted inline with the plate, and almost directly underneath where the tube would interface with the bearings. The spring then pinched the top together securing the tube.

To determine the requirements for the motors, we proceeded as follows: we first determined the force required to bend the 3D printed CAAR tube, building in a safety factor of 10. We used a spring scale to measure the force required to bend the CAAR tube and found it to be a maximum

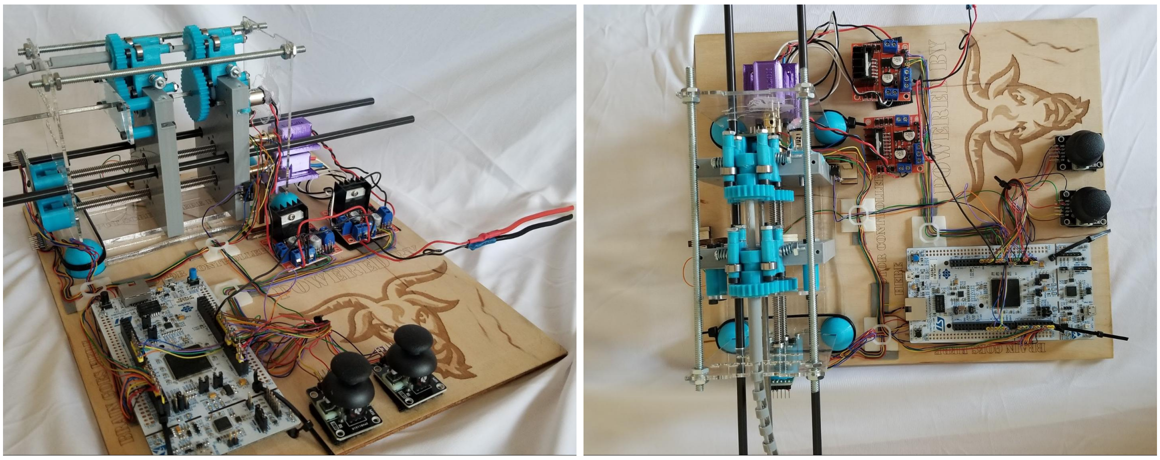


Figure 2.14: *Assembled Siren with board in isometric view (left) and top view (right).*

of about 7N. We then used the following equation, obtained from [13], to calculate the necessary motor torque:

$$T_d = \frac{P * L * e}{2 * \pi} \tag{2.1}$$

where T_d is the torque, P is the load, L is the lead, and e is the efficiency. With the screw lead being $0.2\frac{in}{rev}$ and assuming efficiency equal to 0.5, the driving torque required was determined to be 28 mNm. Based on the lead, we determined that a no-load speed of around 120rpm would be slow enough to control the system accurately. We decided to use 12V Micro Metal HPCB Gearmotors (Pololu Corporation, Las Vegas, NV) with a 298:1 gearbox, as they had a no-load speed of 110rpm and fit the torque requirements established above. We also selected generic L298N Dual H-Bridge motor controllers as they were simple and inexpensive.

Once fully assembled (Figure 2.14), Siren's redesigned components functioned as intended and met the main goal of being a testable, motorized prototype. The redesigned quick-release mechanism worked this time; the tubes fit in, rotated freely without popping out, and meshed with the rest of the gear train. Siren is currently used only for the CAAR configurations, so the front plate does not have a motor for rotation and the middle shaft extends through both plates so that one motor can turn both of the tube gears. We laser cut a board for all of the electronics to mount to which increased the stability of the mechanism and reduced any accidental breaking. Limit switches were added to the sides of the plates with flags to mark the home position as well as the back most limit for each plate.

Version 3: Banshee The third prototype, nicknamed Banshee, was to be the final iteration. Based on our initial plans, this final prototype would have integrated a Nitinol end effector. The plates were to be machined out of metal to make them sturdier. The transmission would have been made longer so that the plates have a longer distance to translate. A box to house the electronics was going to be made attached underneath the transmission.

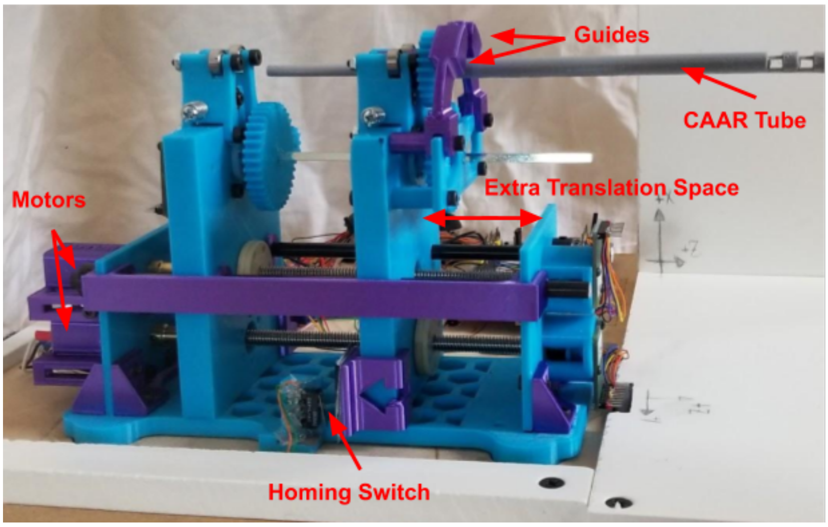


Figure 2.15: *Side view of final version prototype.*

Unfortunately, due to the COVID-19 pandemic, the WPI campus was shut down and we did not have access to the tools and workspaces necessary to build Banshee. Due to these events, we instead decided to improve on Siren. The final version can be seen in Figure 2.15. Fortunately, teammates had access to personal 3D printers, which allowed us to design new parts to test. Although it would have been preferable to make the parts out of a sturdier material, such as metal and acrylic, this was the best we were able to accomplish with the closure of the campus. There were two significant changes made that improved functionality: maximizing space and tube guides. While testing, we frequently ran into errors where the tip would be commanded to go out of bounds of the reachable workspace. Siren initially had a maximum translation of only 25mm, and it was difficult for a user to see where the bounds of reachable workspace were, and this often resulted in the device attempting to go out of bounds. We redesigned and 3D printed new parts and were able to double the translation travel. This was under the constraint of not having access to a machine shop to cut a new lead screw, which would have been a better solution. With this, moving the end effector became a lot easier. However, there was one other challenge; while both of the plates were able to recalibrate using limit switches, we had no way of zeroing the rotation of the tube. At first we tried to line it up through visual inspection, yet during testing this proved to be unreliable. As a result, we designed a guide that would only allow the tube to be placed in only one direction. These guides can be seen from the front view of the device, shown in Figure 2.16. Once it was fully in the quick-release, it could rotate freely and be taken out in any orientation.

2.2.4 Circuit Design and Components Selection

To control the robot, we decided to use the STM32 Nucleo 144 embedded board due to its cost, versatility, and familiarity with the system. This board is also compatible with multiple

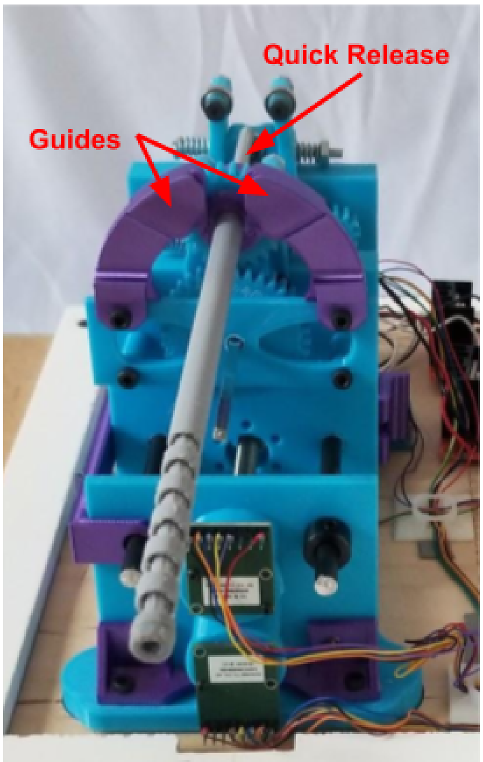


Figure 2.16: *Front view of the final prototype featuring the guides that help to align the tube during placement.*

frameworks, specifically Arduino and Mbed.

We decided on using absolute Hall Effect encoders for the feedback. For a baseline parameter for resolution, we used ± 0.02 degrees for rotation, which was obtained by looking at the results from similar systems. To get this resolution, we used the AS5048 High-Resolution Position Sensor, enabled with SPI. To home the plates, we used two limit switches.

2.3 Kinematic Model

When manipulating a notched tube manipulator, the controller needs to have an understanding of how the tube will move when actuated. It is important, then, to model the forward kinematics and the inverse kinematics of the notched tube. The forward kinematics determines the position of the tube given a specific configuration of actuator variables. The inverse kinematics gives the necessary actuator configuration to achieve the desired tube position.

2.3.1 Forward Kinematics

The forward kinematics allows us to understand the position of the tube, as it is actuated. Forward kinematics for rigid link manipulators are traditionally solved using the Denavit-Hartenberg

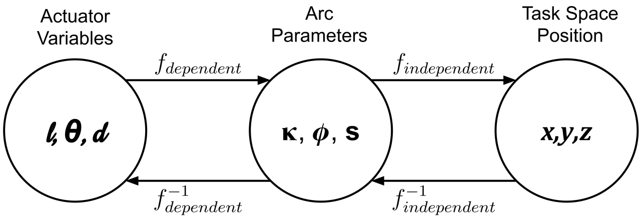


Figure 2.17: *To determine the tip position of the end effector, the actuator variables (tendon displacement, rotation, and advancement) are first mapped to a set of arc parameters (curvature, rotation, and arc length). The arc parameters are then mapped to a tip position in Cartesian Coordinates [9].*

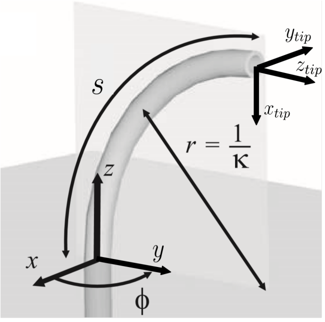


Figure 2.18: *The arc parameters, κ, ϕ, and s, are displayed in respect to a constant curvature segment. The curvature of the arc is represented with κ. The angle to rotate the base frame to keep the base and tip y-axes parallel is ϕ. The arc length of the tube is represented with s. Adapted from [9].*

(D-H) convention and homogeneous transformation matrices. However, these methods cannot be used for flexible continuum (i.e. continuously bending) robots as they do not contain joints and links in the traditional sense. Instead, we can use the approach described in Webster et al. [9], and represented in Figure 2.17.

To model the kinematics of a continuum robot, we start by assuming that the tube will bend in a shape that can be described as a sequence of piecewise constant curvature segments, each of which can be described by a set of arc parameters [9]. The arc parameters, κ, ϕ, and s are respectively the curvature, rotation, and arc length for the constant curvature segment (refer to Figure 2.18). The arc parameters for each notch are determined by the position of the actuator variables. We create a mapping from actuator variables to arc parameters, and a mapping from arc parameters to a point in the task space. The mapping from actuator variables to arc parameters is called the *robot-specific mapping*. The mapping from arc parameters to task space is referred to as *robot-independent mapping*.

For a single curved link, the transformation between the base of the link to the tip can be represented using two 6-dimensional twist vectors. The first twist, ξ_{rot}, represents a rotation about the z-axis of the tube by ϕ (Figure 2.18), i.e.,

$$\xi_{rot} = \begin{bmatrix} v_{rot} \\ w_{rot} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{2.2}$$

This rotation makes the y-axis at the base of the link parallel to the y-axis at the tip of the link (Figure 2.18) [9]. The second twist, ξ_{inp}, represents an in-plane transformation which brings the coordinate frame along the link from base to tip [9], i.e.,

$$\xi_{inp} = \begin{bmatrix} v_{inp} \\ w_{inp} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & \kappa & 0 \end{bmatrix}^T \tag{2.3}$$

To calculate the robot-independent transformation matrix for each curved section, we take the twist coordinates above and map them to 4x4 elements of $\mathfrak{se}(3)$ (i.e., the Lie Algebra of the special Euclidean group SE(3)):

$$\hat{\xi}_{rot} = \begin{bmatrix} \hat{\omega}_{rot} & v_{rot} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.4}$$

$$\hat{\xi}_{inp} = \begin{bmatrix} \hat{\omega}_{inp} & v_{inp} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \kappa & 0 \\ 0 & 0 & 0 & 0 \\ -\kappa & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.5}$$

We can then use the following product of exponentials to calculate the homogeneous transformation matrix for a curved section:

$$T = e^{\hat{\xi}_{rot}\phi} e^{\hat{\xi}_{inp}s} \tag{2.6}$$

which expands to:

$$T = \begin{bmatrix} \cos(\phi)\cos(\kappa s) & -\sin(\phi) & \cos(\phi)\sin(\kappa s) & \frac{\cos(\phi)(1-\cos(\kappa s))}{\kappa} \\ \sin(\phi)\cos(\kappa s) & \cos(\phi) & \sin(\phi)\sin(\kappa s) & \frac{\sin(\phi)(1-\cos(\kappa s))}{\kappa} \\ -\sin(\kappa s) & 0 & \cos(\kappa s) & \frac{\sin(\kappa s)}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

As we shall see in the following, the transformation matrix in Equation 2.7 can be directly applied to model the kinematics of individual notched sections in a tube [3]. Straight sections of length s in the body of a tube can be modeled by using the limit of the same matrix for κ tending to zero, i.e., $\lim_{\kappa \to 0} T$. Creating a transformation matrix for each section of the tube and multiplying them sequentially will produce the transformation from the base of the notched tube to the tip of the notched tube.

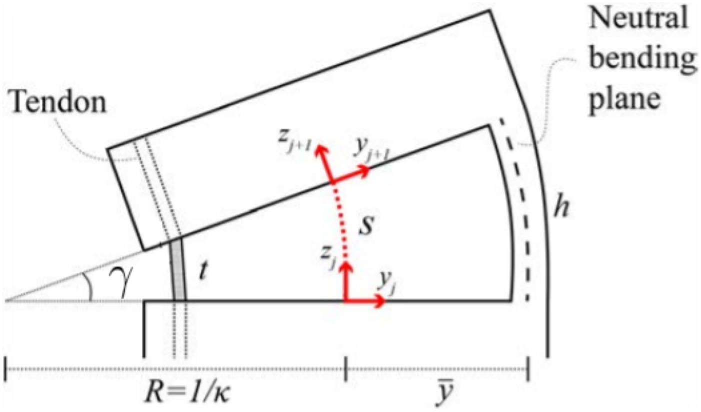


Figure 2.19: *Representation of a single notch and the arc parameters κ and s that describe it. The neutral bending plane maintains the length h, the height of the cut, during bending. The distance of the neutral bending plane from the center of the tube is denoted by $\bar{y}$. Adapted from [3].*

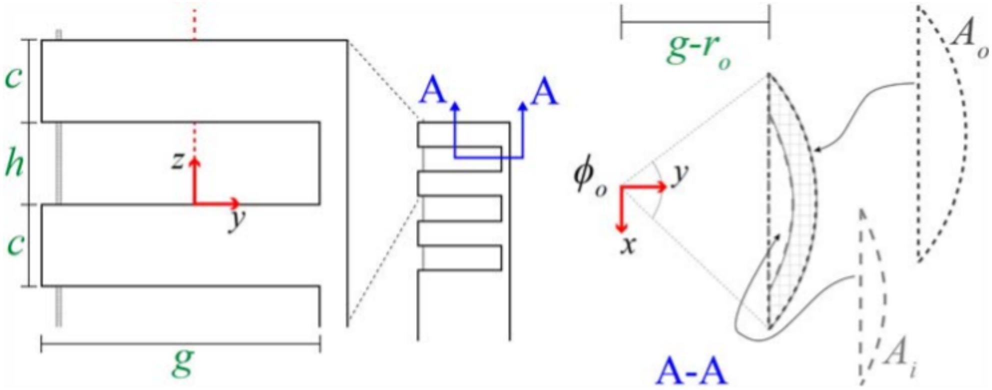


Figure 2.20: *Depiction of variables used in deriving curvature,κ, and arc length s for a multi-notch tendon driven robot. Reproduced from [3].*

2.3.1.1 Robot-Dependent Mapping for Notched-Tube Manipulators

To understand the kinematics of the CAAR manipulator used in our robot, it is convenient to first review the kinematics of a similar yet simpler construct, namely tendon-actuated notched tubes [3]. Let us consider an individual notched section (Figure 2.19). We can find a mapping between tendon displacement and arc parameters. We first find the distance of the neutral bending plane to the center of the tube (Figure 2.19). The neutral bending plane is the area of the tube that does not experience longitudinal strain when bent [3]. The neutral bending plane, $\bar{y}$, can be found with simple geometry:

$$\bar{y} = \frac{\bar{y}_o A_o - \bar{y}_i A_i}{A_o - A_i} \tag{2.8}$$

where A_o and A_i are the areas defined in Figure 2.20. The determine these areas, we use the following relations:

$$A_o = \frac{r_o^2(\phi_o - \sin(\phi_o))}{2} \quad A_i = \frac{r_i^2(\phi_i - \sin(\phi_i))}{2} \tag{2.9}$$

$$\bar{y}_o = \frac{4r_o \sin^3(\frac{1}{2}\phi_o)}{3(\phi_o - \sin(\phi_o))} \quad \bar{y}_i = \frac{4r_i \sin^3(\frac{1}{2}\phi_i)}{3(\phi_i - \sin(\phi_i))} \tag{2.10}$$

$$\phi_o = 2\arccos(\frac{g - r_o}{r_o}) \quad \phi_i = 2\arccos(\frac{g - r_o}{r_i}) \tag{2.11}$$

We can now solve for the curvature, κ, and the arc length, s, using $\bar{y}$, arc geometry, and a small-angle approximation when given a desired tendon displacement:

$$\kappa \approx \frac{\Delta l}{h(r_i + \bar{y}) - \Delta l \bar{y}} \quad s = \frac{h}{1 + \bar{y}\kappa} \tag{2.12}$$

With this approach, we find the curvature and arc-length for each notch individually, assuming the tendon displacement is equally distributed among the notches. For each straight section, we assume the curvature is 0, and the arc length is the length of the tube. Then using the transformation matrices from equation 2.7 we can build the forward kinematics for a notched tube wrist [9].

2.3.1.2 Robot-Dependent Mapping for a CAAR Manipulator

The kinematic modeling for a CAAR manipulator uses an approach similar to the one we have described above for a tendon-actuated notched tube [5]. Since the CAAR is composed of two nested tubes, we need to find the neutral bending plane of both tubes. We shall denote $\bar{y}_1$ as the neutral bending plane of the outer tube and $\bar{y}_2$ as the neutral bending plane of the inner tube. The values of $\bar{y}_1$ and $\bar{y}_2$ can be solved for using the same set of relations derived above (Equations 2.8 - 2.11). Having determined $\bar{y}_1$ and $\bar{y}_2$, we can calculate the total bending angle γ using the displacement between the inner and outer tubes,τ, and the following relation derived in [5]:

$$\gamma = \tau(\bar{y}_1 + \bar{y}_2)^{-1} \tag{2.13}$$

We again assume that all the bending occurs in the notched sections of the tube. We also assume the bending angle for the entire tube can be divided evenly among the notches [3]:

$$\gamma_j = \frac{\gamma}{n} \tag{2.14}$$

The arc length of a single notch, s_j is then gathered from its bending angle, γ_j [5]:

$$s_j = h - \bar{y}_1\gamma_j \tag{2.15}$$

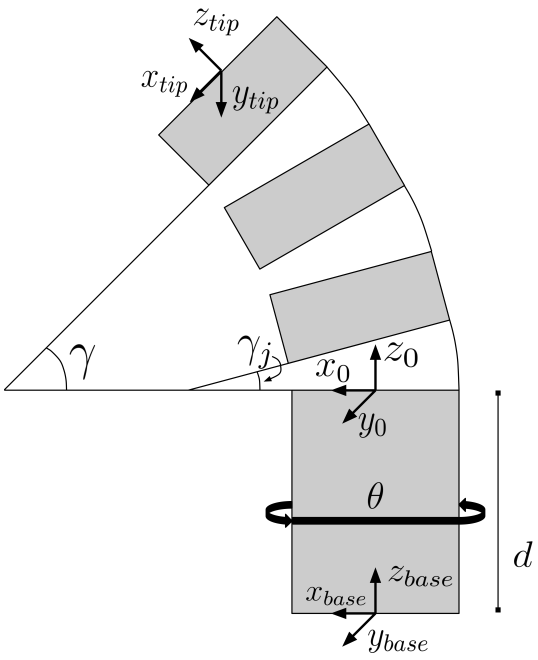


Figure 2.21: *The figure represents a simplified kinematic model of a single notched tube manipulator, which forms the basis of the CAAR tube. The transformation from the base frame to the base of the first notch is based on the rotation about the z-axis of the base frame and a translation along the z-axis of the base frame. Adapted from [11].*

Once we have the bending angle and the arc length for a single notch, we can derive the curvature using the following simple relation:

$$\kappa_j = \frac{\gamma_j}{s_j} \tag{2.16}$$

Having the curvature and arc length for each section of the tube allows us to use the transformation matrix in equation 2.7. With the transformation matrices for each individual section, we can generate a 4x4 homogeneous transformation matrix that gives us the forward kinematics of the tip of the robot. We then add another frame at the base which will have curvature of 0, and arc length of d. This frame will handle the translation of the tube as well as the rotation (Figure 2.21).

2.3.2 Inverse Kinematics

As with the forward kinematics, we could, in principle, solve the inverse kinematics in two steps: the first is a robot-independent mapping finding the relationship between the task space position and the arc parameters to achieve that position; the second is a robot-dependent mapping describing the relationship between the arc parameters and the joint variables [9]. Instead of attempting to directly invert the forward kinematics relations, here we propose a numerical solution that is based on the robot's geometry.

Given a desired pose identified by Cartesian coordinates (x_d, y_d, z_d), we can converge on this location by taking advantage of the geometry of the CAAR manipulator. Assuming no rotation, when the two tubes are displaced, the tip of the manipulator will only move in the robot's local x-z plane, and when the tube is translated the tip will only move in the z-direction. A positive displacement will move the tip in the positive x-direction, and a negative displacement will move the tip in the negative x-direction due to the construction of a CAAR manipulator. Because the translation does not affect the x position of the tip, we can decouple the tip error in the x-direction and z-direction. The error in the x-direction is attributed to the tube displacement, and the error in the z-direction is attributed to the translation. Assuming that the robot is initially in its home configuration, we can iteratively solve for the necessary tube displacement using the following relation:

$$\tau_k = \tau_{k-1} + (x_d - x)G_l \tag{2.17}$$

where τ_k is the tube displacement during the k^{th} iteration and G_l is a positive constant. When $x_d \approx x$, then $\tau_d \approx \tau$, where τ_d is the tube displacement required to reach (x_d, z_d) and τ is the actual tube displacement at the end the loop. Because the translation can only account for error in the z direction, we approximate the translation in a similar way:

$$d_k = d_{k-1} + (z_d - z)G_d \tag{2.18}$$

where d_k is the tube displacement during the k^{th} iteration and G_d is a gain. We can finally solve for the necessary rotation θ by using the arc-tangent function:

$$\theta = \arctan \frac{y_d}{x_d} \tag{2.19}$$

The rotation about the z-axis by θ creates a new frame as seen in Figure 2.22. We can now solve the problem like we did the two dimensional case. The rotation by θ about the z axis puts the desired location in the new frame's x-z plane. The error, $(x_d - x)$, get replaced with a radial error, r_e, in Equation 2.17 where:

$$r_e = \sqrt{(x_d - x)^2 + (y_d - y)^2} \tag{2.20}$$

If we want the tube to be able to bend backwards, we must add π to the necessary rotation, and transforming G_l in Equation 2.17 into a negative constant. To have the robot bend backwards, we need to decrease τ. With a negative gain, a positive error will subtract from the current τ, leading to the desired result.

We tested the inverse kinematics convergence on 3000 setpoints within the robot's workspace (Figure 2.23) in MATLAB, on a machine equipped with an Intel i7-6700HQ CPU (2.6 GHz) and 8 GB of RAM. The algorithm was able to converge on all points within the robot's workspace. To measure the speed we measured the number of iterations and the average time in milliseconds the algorithm took to converge to an accuracy of 1×10^{-3} mm. To measure the accuracy of the system, we measured the accuracy after 50 iterations through the loop. The results are reported in Table 2.1.

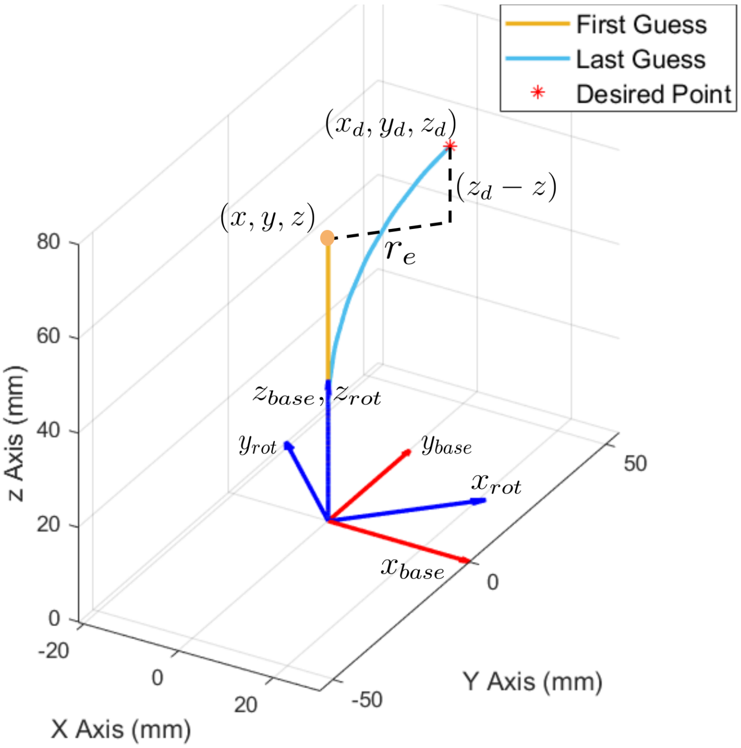


Figure 2.22: *Illustration of the quantities used in the inverse kinematics algorithm. The rotation frame is rotated about z_{base} by θ to bring the problem into two dimensions. The tip error of the manipulator is decoupled and measured as $z_d - z$, and the radial error, r_e.*

Speed Test Results		Accuracy Test Results
Average Time (ms)	Average Number of Iterations	Average Error (mm)
18.4	15.06	9.26e-12

Table 2.1: *Table displaying the speed and accuracy of the inverse kinematics algorithm. The speed test includes both the average time in milliseconds and the average number of iterations*

2.4 Software Design

The control architecture of the robot is shown in Figure 2.24. The architecture includes joysticks, the Nucleo board, the actuation unit, and a machine running both MATLAB and the web visualization. This section details the process by which we decided upon this final architecture and provides brief descriptions of the various components.

2.4.1 Implementation of the Inverse Kinematics

When implementing a control scheme for a robot, it is important to consider the available computational resources. This computation may require non-trivial processing power, including

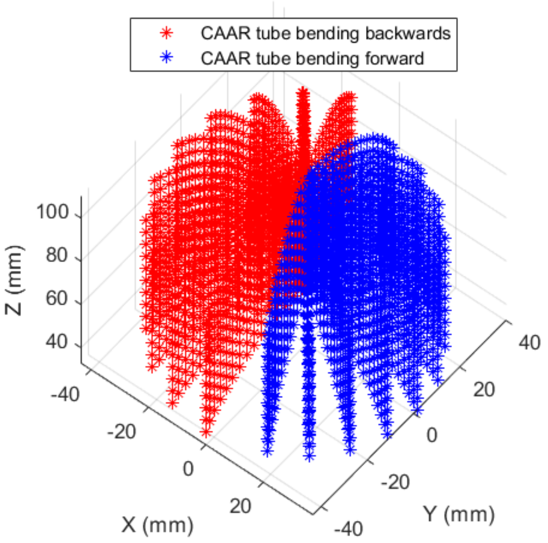


Figure 2.23: *Points reached by the CAAR tube when tested for speed and accuracy metrics. The rotation was kept between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ to test both directions of bending.*

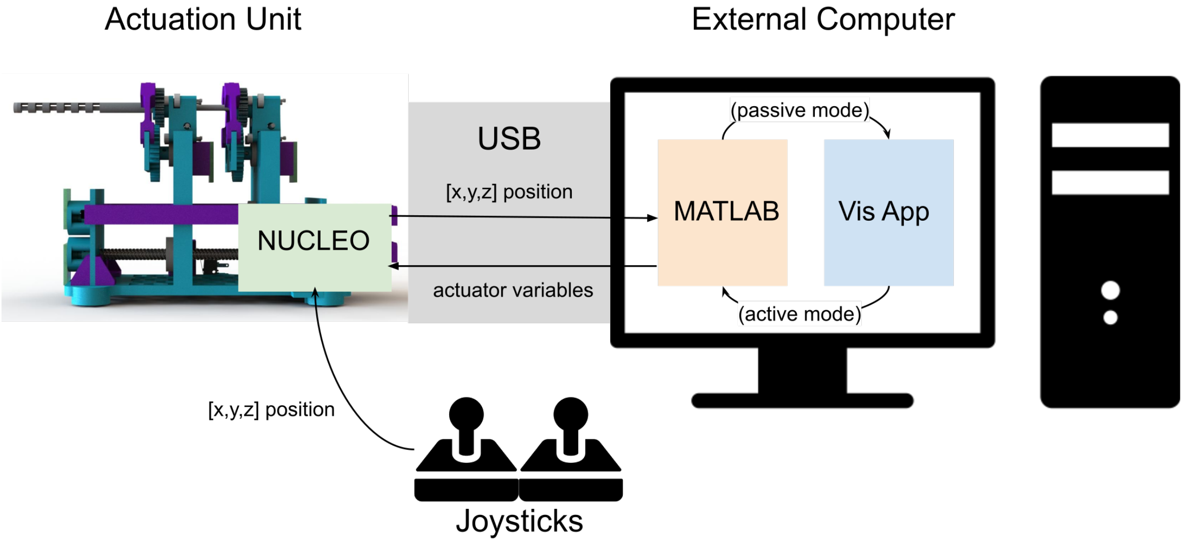


Figure 2.24: *System Architecture Diagram.*

optimized libraries and multi-threading, that oftentimes cannot be easily deployed on embedded systems. In our design process, a decision had to be made as to where the kinematics of the robot would be executed. When making this decision, we considered two options: one was to implement the kinematics in C++ so that it could run on the Nucleo embedded board; the second option was to implement the kinematics in MATLAB and run it on a separate computer connected to the Nucleo board via a serial bus. Each solution has its benefits: implementing the kinematics in MATLAB is arguably easier due to its native syntactic support for matrix operations. However, this solution requires the use of an external computer in addition to the Nucleo board and the use of a communication bus. On the other hand, implementing the kinematics in C++ could have significant benefits in terms of execution time, and avoid the overhead associated with communication between the Nucleo board and an external computer.

To determine the best option for our scenario, an experiment was run to find the baseline "cost" of running inverse kinematics in MATLAB. Using MATLAB's timing functions, 2000 task space locations were passed to an implementation of the inverse kinematics, and each execution was timed. The chosen locations were evenly spaced as to approximate the entire task space. This test was run in MATLAB, on a machine equipped with an Intel i7-6700HQ CPU (2.6 GHz) and 8 GB of RAM. The average execution time was found to be 45ms. We then used MATLAB Coder to translate the implementation of the inverse kinematics in C++. Being the result of an automatic translation process, the resulting code was not straightforward to read and interpret for a human. Running this code on the Nucleo board required 150+ ms at certain task space locations and an average delay of approximately 100ms using the same points as the previous test. The last test we ran involved measuring the communication latency between a Nucleo board and an external computer over a serial bus. We configured the serial bus to operate at a baud rate of 115,200, and measured an average packet transmission time of 50ms. This number is representative of the round trip time of a packet and was measured by using the millis() function on the Nucleo board in order to time the sendPacket() and receivePacket() functions. Based on the results of these experiments, we decided to move forward with the use of an external control computer running the inverse kinematics in MATLAB.

The design of the serial communication is akin to a C++ "client" interfacing with a MATLAB "server". The MATLAB program runs continuously and waits for an incoming packet. When the Nucleo sends a packet of the form "[x;y;z]", MATLAB will parse the message and send back a message of the form "[l;theta;d]". The reverse bracket acts as a terminator character so that the Nucleo knows to stop reading data from the serial bus.

2.4.2 Robot Control Loop

Once it was decided that the device would utilize serial communication to decode XYZ set points, the main control loop was designed as a finite state machine (FSM). The FSM received inputs from a pair of joysticks (Figure 2.25) and uses these inputs to control the position of the robot.

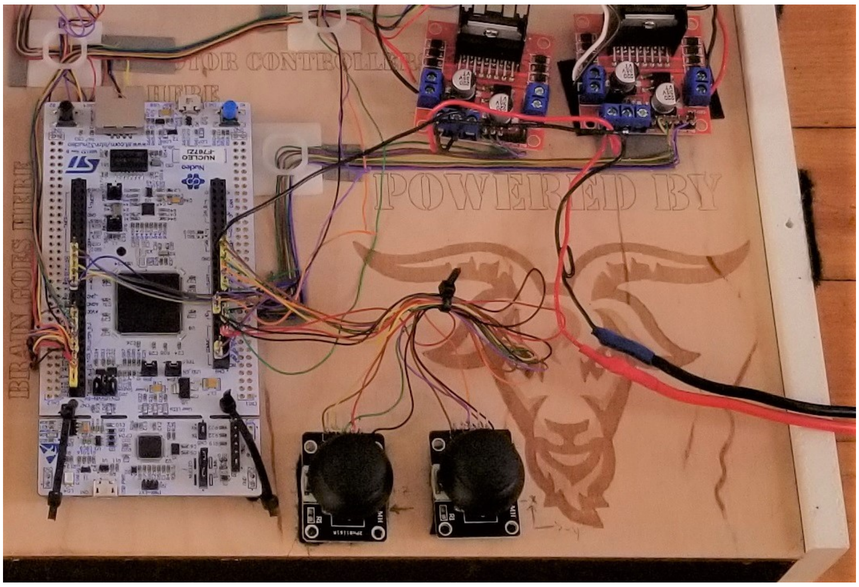


Figure 2.25: *Joysticks attached to the Nucleo board on the final prototype.*

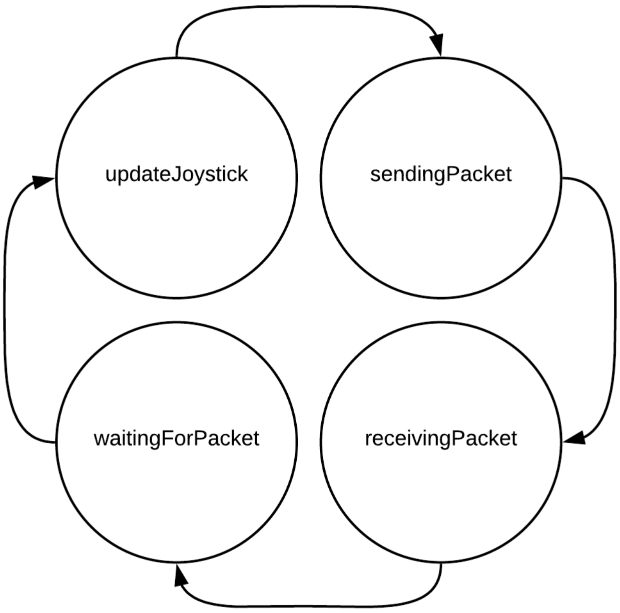


Figure 2.26: *Main Loop State Machine.*

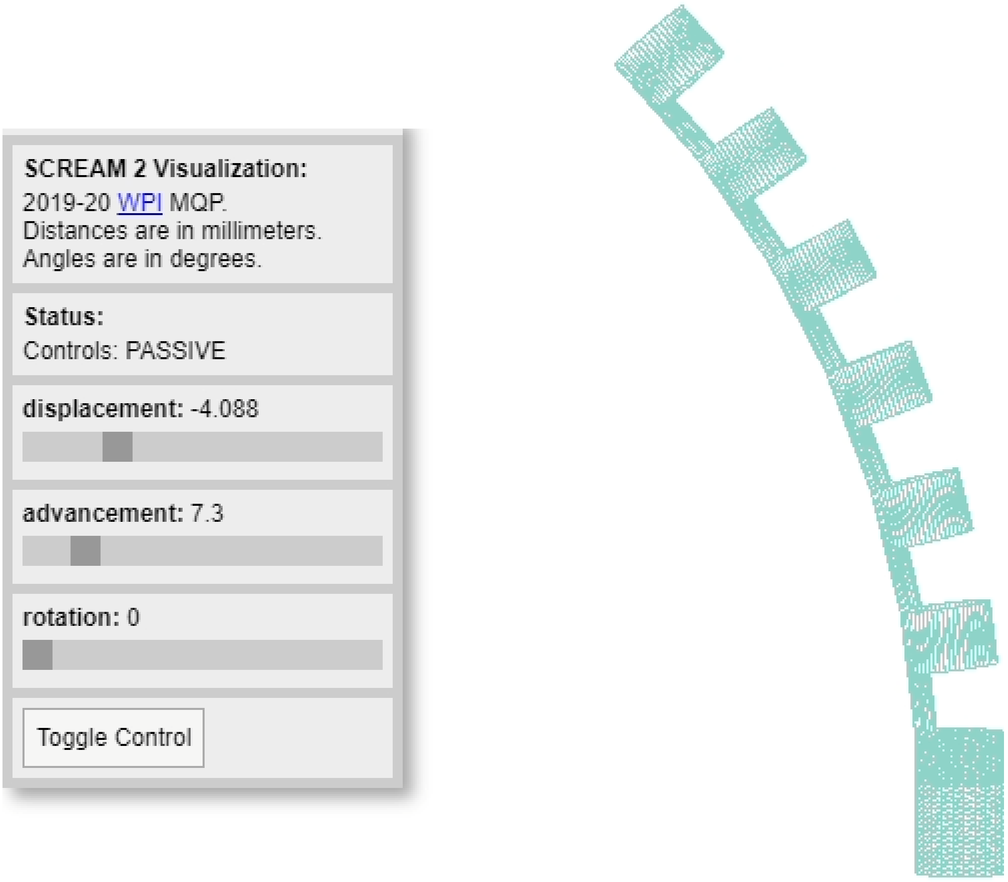
As shown in Figure 2.26, the FSM has four states: updateJoystick, sendingPacket, receivingPacket, and waitingForPacket. Execution begins in the updateJoystick state, where it polls for user input through the two joysticks. Together, the two joysticks are used to control the tip's XYZ setpoint. Moving the left joystick left and right moves the tip's z-position forward and backward respectively. The right joystick increases the y-position when moved left, and decreases the y-position when moved right. The right joystick also increases the x-position when moved down, and decreases the x-position when moved up.

Once the NUCLEO detects a change in setpoint through the joysticks, it changes state to sendingPacket, where the new XYZ position is sent via USB to an external computer running MATLAB. The state then changes to receivingPacket, where the Nucleo waits for the first byte of data. Once MATLAB has solved the inverse kinematics, the corresponding actuator values are sent to the Nucleo board. While this message is being transmitted over, the waitingForPacket state is entered until the entire message is received. A PID control loop is ran on the Nucleo board at a frequency of 500 Hz, independently of the main control loop. Once the new actuator values are received from MATLAB, the PID controller will use them as reference.

2.4.3 Visualization Design

Besides implementing joystick control, we created a visualization interface as described below. This interface was created using Javascript, based on the implementation of a similar software previously developed by the SCREAM 1.0 team. The SCREAM 2.0 version of this tool differs from version one in multiple ways, including the utilization of a tube model that reflects the size and movement of our CAAR tube, the implementation of MATLAB communication, and the addition of passive and active modes to monitor and control the robot.

The application's communication protocol had to be completely changed from SCREAM 1's implementation in order to achieve a reasonable latency. The application runs locally in a web browser on the machine that is running MATLAB. Communication with MATLAB was achieved by accessing shared memory space in the form of a .txt file. Both MATLAB and the SCREAM 2 application can access this file and read/write data in the form of x,y,z, where x,y, and z are doubles. In passive mode, the application reads from this file, which contains the robot's current setpoint. In active mode, the application writes a setpoint to the file, which MATLAB reads, and passes to the robot. This mode can be toggled in the menu shown in Figure 2.27.



SCREAM 2 Visualization:
2019-20 WPI MQP.
Distances are in millimeters.
Angles are in degrees.

Status:
Controls: PASSIVE

displacement: -4.088

advancement: 7.3

rotation: 0

Figure 2.27: *Visualization control panel (left) and tube status (right).*

Chapter 3

Testing and Validation

In this chapter, we report on the testing and verification of the kinematics of our robot. We created a test setup, shown in Figure 3.1, that enabled us to record the position of the tip of the end effector as it moves to reach a setpoint. For each test, we focused on movement in the x-z plane. Any movement in the x-y-z space can be reduced by removing base rotation, similar to how we solve the inverse kinematics analytically. The resulting videos of the tests were analyzed in a video tracking and physics modeling software called Tracker. This enabled accurate tracking of the position of the tip of the end effector as it moved between setpoints. **Because the WPI campus was closed due to the COVID-19 outbreak in D term of 2020, the creation of the experimental setup and the experiments were carried out at home by one of the team members.**

The test setup consisted of a table, with pieces of wood to keep the device in the same position every time. Two white boards functioned as backgrounds for the cameras. It had a platform above with a hole for a camera to look down at the end effector; this was for recording the y-z plane. On the side was an area to hold a second camera to record the device from the x-z plane. Both camera areas had guides and holders to ensure that the camera would not move while recording. A ring light was mounted to the side to provide consistent lighting and remove shadows so that the video analysis would be as clear as possible.

3.1 Actuation Unit Forward Kinematics Test

To test the forward kinematics on the actuation unit, we performed a series of tests to measure the translation and bending of the end effector after being mounted in the system. For our first test, we focused on the pure translation of the tubes. We brought the device to four different translation setpoints: 5, 10, 15, and 20mm. After it reached each point and settled, the robot was

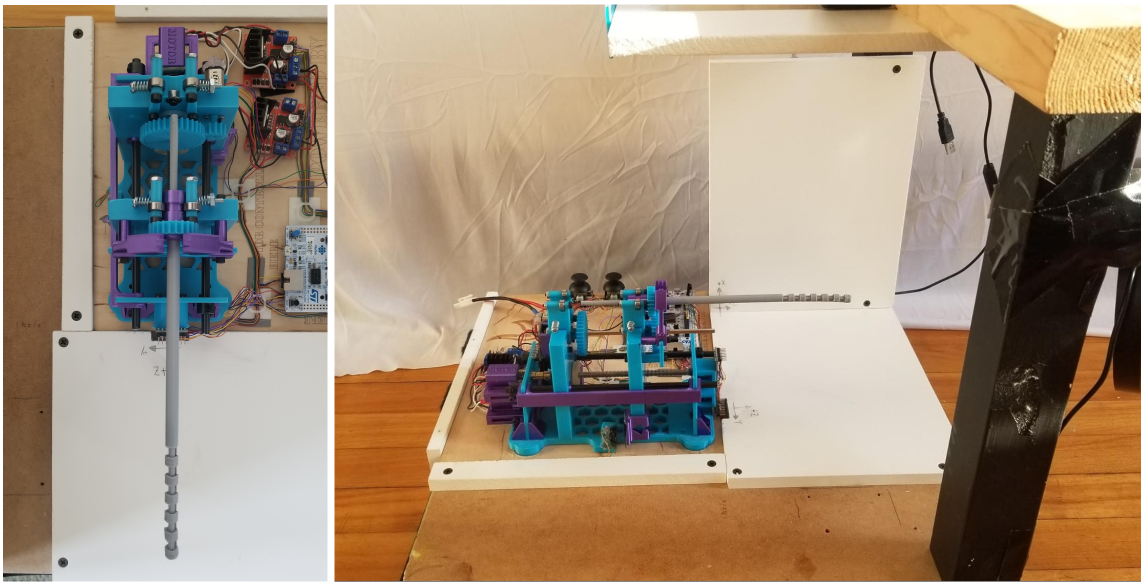


Figure 3.1: *Top view of the device from the camera on the raise platform (left) and side view of the device and the entire test setup (right).*

set to return to 0mm translation. This allowed us to measure how well the robot could maintain its zero position. The second set of tests involved having the robot bend the end effector and validating that it had bent to the proper x-z position. For this test, we moved it from 0 to 7mm of tube displacement in 0.5mm increments. We did not reset the tube displacement to 0mm after each trial in an attempt to see the effect of the persistent force of the tubes on the actuator plates as they were moved apart. The Tracker results for this trial can be seen in Figure 3.2.

After reviewing the results, it was found that the position of the end effector did not match the output from the forward kinematics as the two tubes were displaced (see Figure 3.3). Part of this error could be attributed to the assumption, made at modeling time, that each notched section of the tube will bend an equal amount, while in practice this is not always the case. This phenomenon has been observed by other researchers as well [3]. Another suspected source of error was in the lack of rigidity of most of the components in the actuation unit. Being mainly 3D printed, its tolerances and rigidity can not match up to machined metal parts. Upon deeper analysis, we discovered significant deformation during actuation, especially under the strain of pulling the tubes, which caused the tubes to not be displaced as much as intended. In one such case, seen in Figure 3.4, an expected tube displacement value of 7mm resulted in a displacement of only about 4mm.

3.2 Manual Forward Kinematics Test

To verify that the inaccuracies observed in our first experiment were to be attributed primarily to the actuation unity, and not on mistakes possibly made at modeling time, we tested the CAAR

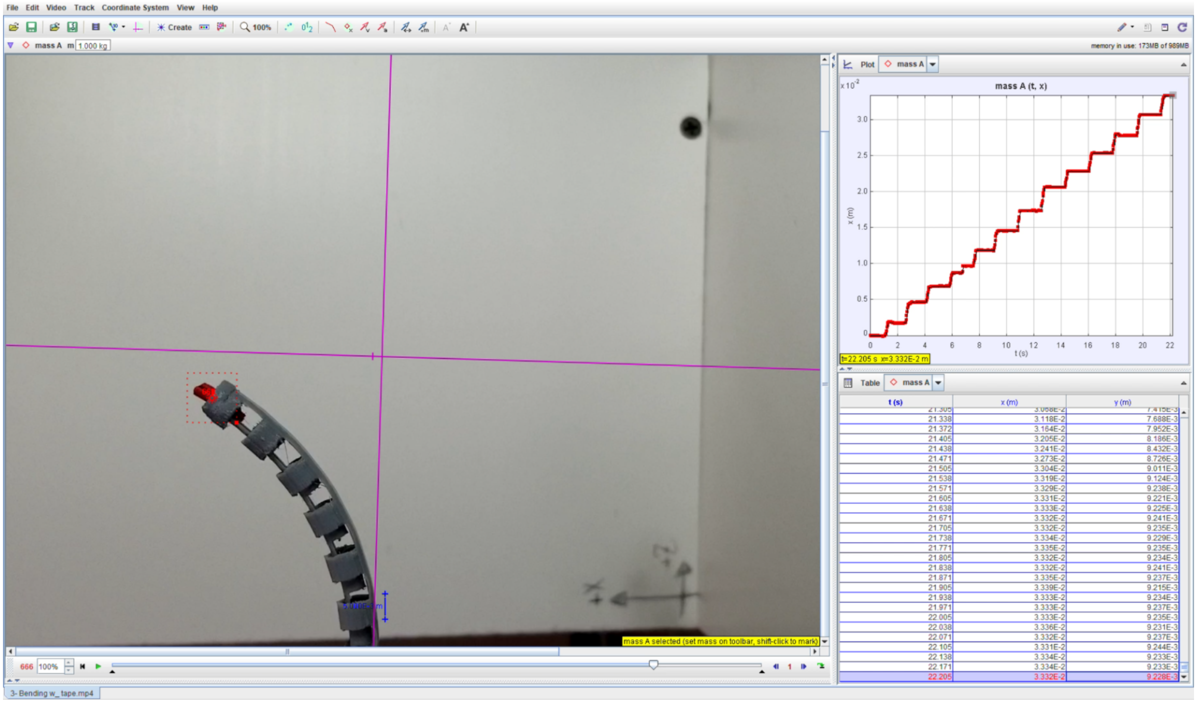


Table mass A

t (s)	x (m)	y (m)
21.305	3.068E-2	7.415E-3
21.338	3.118E-2	7.688E-3
21.372	3.164E-2	7.952E-3
21.405	3.205E-2	8.186E-3
21.438	3.241E-2	8.432E-3
21.471	3.273E-2	8.726E-3
21.505	3.304E-2	9.011E-3
21.538	3.319E-2	9.124E-3
21.571	3.329E-2	9.238E-3
21.605	3.331E-2	9.221E-3
21.638	3.333E-2	9.225E-3
21.671	3.332E-2	9.241E-3
21.705	3.332E-2	9.235E-3
21.738	3.334E-2	9.229E-3
21.771	3.335E-2	9.235E-3
21.805	3.332E-2	9.234E-3
21.838	3.332E-2	9.241E-3
21.871	3.335E-2	9.237E-3
21.905	3.339E-2	9.215E-3
21.938	3.333E-2	9.234E-3
21.971	3.333E-2	9.237E-3
22.005	3.333E-2	9.235E-3
22.038	3.336E-2	9.231E-3
22.071	3.332E-2	9.237E-3
22.105	3.331E-2	9.244E-3
22.138	3.334E-2	9.233E-3
22.171	3.334E-2	9.233E-3
22.205	3.332E-2	9.228E-3

Figure 3.2: *Tracker program measuring the location of the tip of the end effector in one of the bending tests with the actuation unit. Tracker outputs graphs of the X and Z positions over time alongside a table of all measured values.*

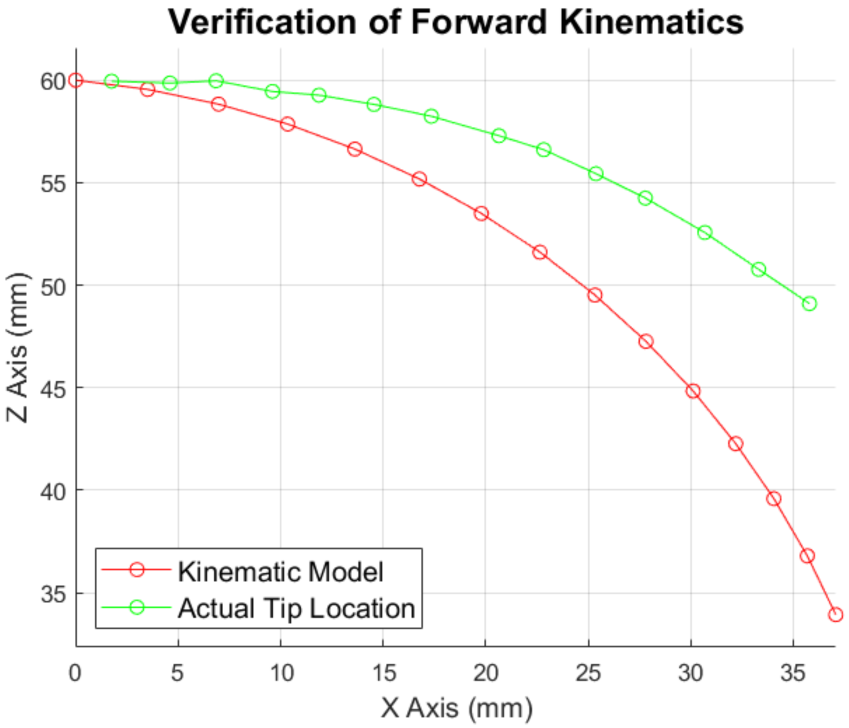


Figure 3.3: *Comparing the actual position of the tip when actuated in the actuation unit to the desired position based on the forward kinematics. The RMSE was 9.72mm.*

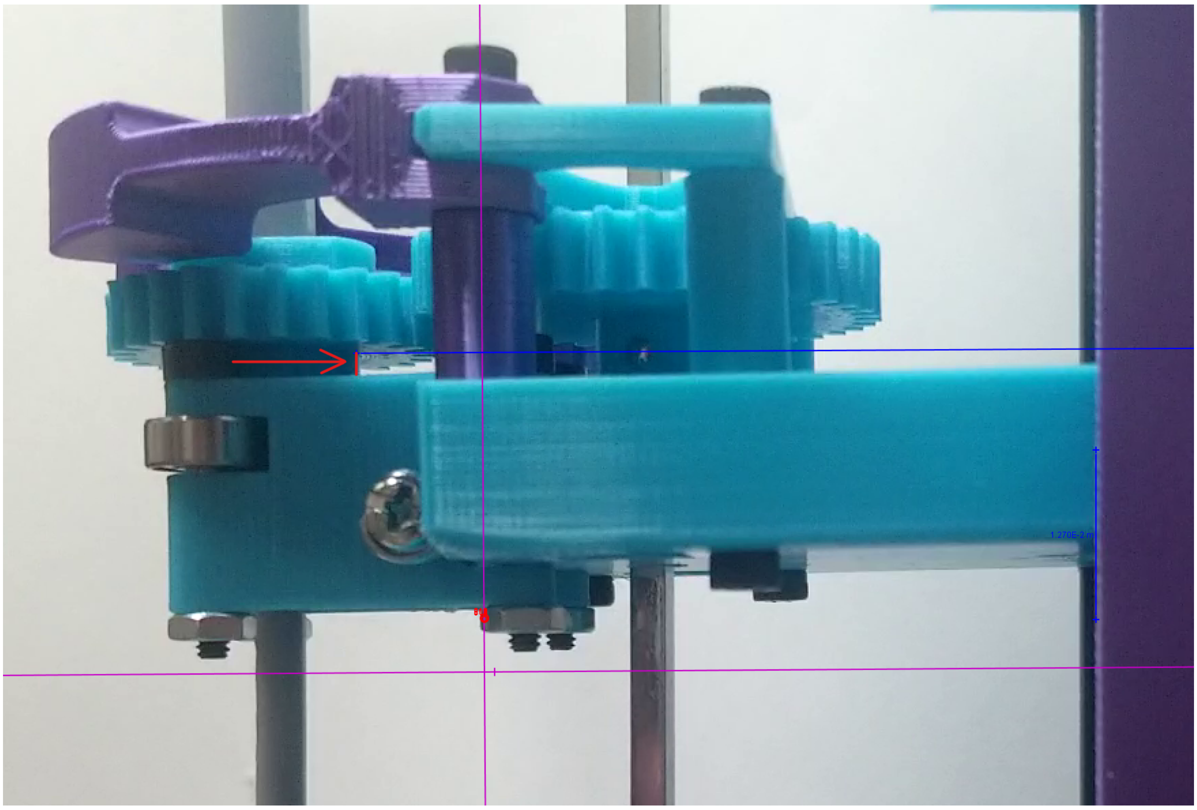


Figure 3.4: *Example of plate bending in the system. The red arrow shows the distance that the front plate has bent backwards from its expected position (blue line) when given a tendon displacement of 7mm.*

manipulator individually by mounting it on a linear slide (Velmex Inc., Bloomfield, NY) and displacing tubes manually. By affixing the outer tube to a clamp on the end of the slide base, a calibrated dial could be used to pull or push the inner tube with a lead screw. Multiple tests were performed, measuring the bending of the 3D printed end effector prototypes with 0.5mm and 1mm increments of tendon displacement. As these tests were performed, an overhead camera setup recorded the resulting bending. These videos were later analyzed in a computer program called Tracker which uses video processing to track the coordinates of the tip of the end effector.

A graph of the results of the expected and actual points are shown in Figure 3.5. The results of the manual bending test show that the device gets less accurate as it bends. This is most likely due to the constant curvature assumption and the assumption that each notch bends uniformly in our kinematics. Part of the issue could have also been in aligning the tube in the apparatus. If the two tubes are not lined up perfectly when they are secured to the linear slide, then the tube will be less likely to follow the kinematics.

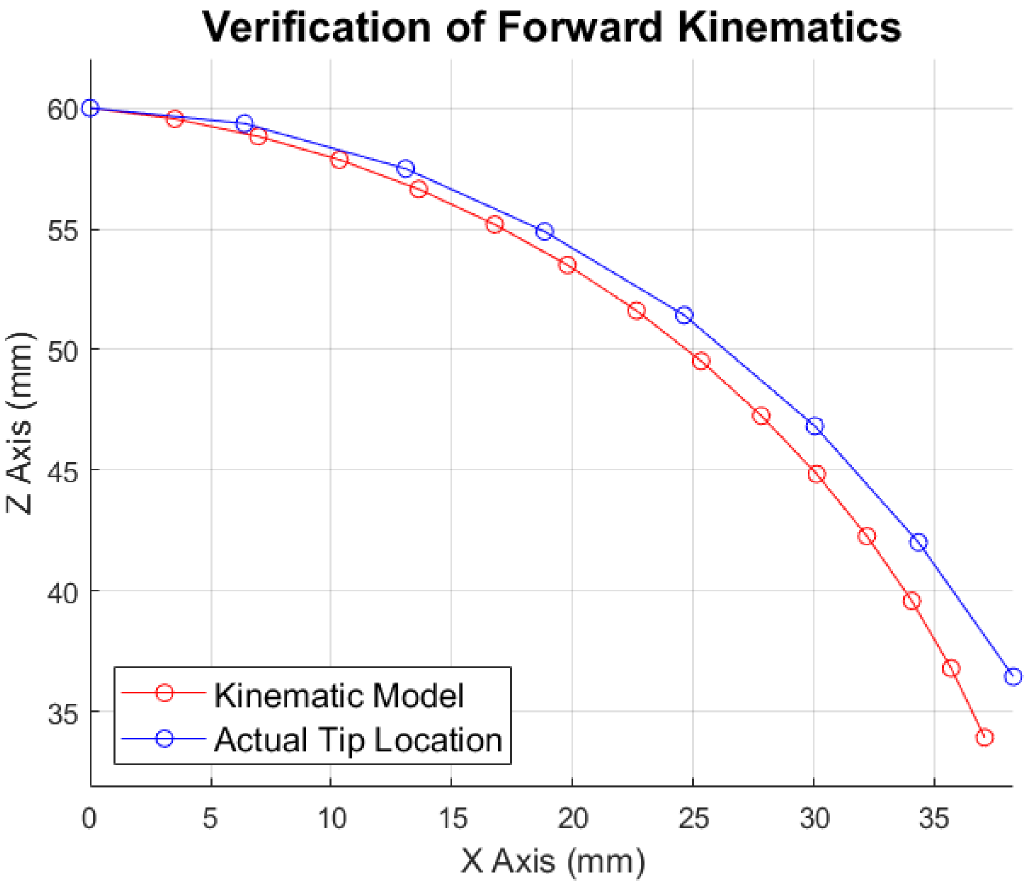


Figure 3.5: *Comparison of the manual bending test with the desired output from the forward kinematics. The RMSE was 1.81mm.*

Chapter 4

Discussion and Future Work

From the work completed this year, there are several areas in which the design, accuracy, and ease-of-use of the system can be improved.

4.1 Advancement Over SCREAM 1.0

Our work advances the work done by the SCREAM 1.0 MQP in the previous academic year in a few key ways. First and foremost, our system is fully motorized, incorporating motors and sensors for all moving parts. This improves on the mostly manual SCREAM 1.0 design that only had a motor for tendon displacement and sensors for rotation and translation. Improvements have also been made to the end effector, opting to replace SCREAM 1.0's single notched tube wrist with a CAAR manipulator design that is more rigid and allows for bending in both the forward and backward directions. Full forward and inverse kinematic control to reach selected setpoints in 3-dimensional space was also achieved, which was something that was missing from the SCREAM 1.0 prototype. Finally, our system supports the hot swapping of tools instead of having a single fixed end effector. We believe that these advancements bring the SCREAM device closer to the capabilities needed for clinical use.

4.2 Technology Readiness Level

According to the Technology Readiness Level measure created by NASA [14], the device developed as a result of this project has a technology readiness level (TRL) of 3; it serves as a proof of concept for a tendon-less robotic notched tube manipulator. To elevate the TRL of the device to be sufficient for clinical deployment, more work must occur. This includes the machining of a small Nitinol end effector, rebuilding the prototype out of metal that can be sanitized, and more extensive testing the robot kinematics and control algorithms.

4.3 Verification of Requirements and Impact of COVID-19

Unfortunately, toward the end of development of this project, progress was stalled due to the COVID-19 pandemic. The closure of machine shops prevented the assembly of a final, aluminum prototype. As a result of this, the final prototype was constructed using 3D printed parts in place of machined metal. This is true of the end effector as well. A Nitinol end effector could not be machined due to lab closures. The testing and validation of the device was also affected by the pandemic. Closure of labs meant that equipment, such as the electromagnetic tracker we planned to use for validation, was no longer available for use. As a result, a testing and recording fixture was built, and video tracking software was used to track the position of the tip of the device.

In terms of the requirements specified in the initial design process, the final robot manages to meet all but one: the end effector's diameter is too large to transit through the working channel of an endoscope. However, the robot has a distal steering mechanism to enable bending and a motorized actuation system. Also, the device is comprised of off-the-shelf components when possible, and used custom components that were easy to manufacture.

4.4 Future Work

4.4.1 Kinematics

There is still work to be done in developing an initial guess to the desired solution for the inverse kinematics when the translation is incorporated. Additionally, the testing showed that the assumptions made when generating the forward kinematics break down as the tube bends. Profiling how much the different sections of the tube bend would increase the accuracy of the forward kinematics.

4.4.2 End Effector

The 3D printed end effector served as an effective prototype for kinematic testing, however, it failed to meet the superelasticity, size, and sterilization requirements that the proposed surgical robot would need to have. Future teams could build upon the aforementioned EDM machine workflow in order to machine a smaller CAAR tube out of Nitinol. Doing this would take the robot beyond a proof-of-concept device and would propel it into the realm of clinical feasibility. Additionally, work to mitigate uneven bending across the length of the end effector could be done, likely taking the form of experimenting with notch geometries.

4.4.3 Actuation Unit

A major source of error within the final prototype actuation unit was internal bending due to the lack of rigidity of 3D printed parts. To combat this, future teams could machine the actuation unit components (namely the translation plates that sit on the lead screw) out of aluminum or another metal material. This is predicted to greatly reduce the error between predicted versus actual end effector position when controlling the robot.

4.4.4 Visualization

During the development of the visualization tool, there were a few features that were fully implemented but omitted in the final design for the sake of streamlining the app for use with the physical robot. This includes controller support, STL loading, and collision detection. It may be beneficial for a future group to add a training mode where an STL can be loaded for the robot to collide within virtual space. Additionally, controller support could be implemented to provide more intuitive joystick control than what is already available on the prototype actuation unit. Overall, finding a way to balance robot communication latency with more complex features is a substantial area for future work.

4.5 Conclusion

In conclusion, this project expanded on SCREAM 1.0's work and laid the groundwork for future projects to continue with ample area to expand our research. Our final device prototype demonstrated the possibility of a motorized version of the manual endoscopic tool. Additionally, our novel work in the kinematics of continuum end effectors, especially with CAAR tubes, will serve as a basis for future research into the area. We hope that our work will eventually be used in a clinical setting to improve the lives of many people.

Bibliography

[1] M. Remacle and H. E. Eckel, *Surgery of larynx and trachea*.
Springer, 2010.

[2] D. Stoeckel and A. Melzer, "The use of ni-ti alloys for surgical instruments," *Materials in Clinical Applications, ed. by P. Vincenzini. Techna Srl*, pp. 791–798, 1995.

[3] P. J. Swaney, P. A. York, H. B. Gilbert, J. Burgner-Kahrs, and R. J. Webster, "Design, fabrication, and testing of a needle-sized wrist for surgical instruments," *Journal of Medical Devices*, vol. 11, Dec. 2016.

[4] K. O'Brien, Z. R. Boyer, B. G. Mart, C. T. Brolliar, T. L. Carroll, and L. Fichera, "Towards flexible steerable instruments for office-based laryngeal surgery," in *2019 Design of Medical Devices Conference*, American Society of Mechanical Engineers, Apr. 2019.

[5] K. Oliver-Butler, Z. H. Epps, and D. C. Rucker, "Concentric agonist-antagonist robots for minimally invasive surgeries," in *Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling* (R. J. Webster and B. Fei, eds.), vol. 10135, pp. 270 – 278, International Society for Optics and Photonics, SPIE, 2017.

[6] R. J. Hendrick, C. R. Mitchell, S. D. Herrell, and R. J. Webster, "Hand-held transendoscopic robotic manipulators: A transurethral laser prostate surgery case study," *The International Journal of Robotics Research*, vol. 34, no. 13, pp. 1559–1572, 2015.

[7] P. J. Swaney, J. M. Croom, J. Burgner, H. B. Gilbert, D. C. Rucker, R. J. Webster, K. D. Weaver, and P. T. Russell, "Design of a quadramanual robot for single-nostril skull base surgery," in *ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference*, pp. 387–393, American Society of Mechanical Engineers Digital Collection, 2012.

[8] S. Amack, M. Rox, J. Mitchell, T. Ertop, M. Emerson, A. Kuntz, F. Maldonado, J. Akulian, J. Gafford, R. Alterovitz, and R. Webster, "Design and control of a compact modular robot for transbronchial lung biopsy," p. 17, Mar. 2019.

[9] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[10] "Office-based laryngoscopy." https://www.olympus.pl/medical/rmt/media/Content/Content-MSD/Documents/Brochures/Laryngoscopy_Manual_Prof.-Hess_15937.pdf.
Accessed: 2020-5-17.

[11] A. Chiluisa, F. Van Rossum, J. B. Gafford, R. Labadie, R. J. Webster, and L. Fichera, "Computational Optimization of Notch Spacing for a Transnasal Ear Endoscopy Continuum Robot," in *2020 International Symposium on Medical Robotics (ISMR)*, In Press.

[12] J. H. Hah, S. Sim, S.-Y. An, M.-W. Sung, and H. G. Choi, "Evaluation of the prevalence of and factors associated with laryngeal diseases among the general population," *The Laryngoscope*, vol. 125, pp. 2536–2542, July 2015.

[13] R. G. Budynas, J. K. Nisbett, and J. E. Shigley, *Shigleys mechanical engineering design*. McGraw-Hill Education, 2011.

[14] "Technology readiness level." https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt_accordion1.html.
Accessed: 2020-5-17.

APPENDIX A

APPENDIX A: AUTHORSHIP

Section	Title	Author
1	Introduction	Everyone
1.1	Motivating Example: In-Office Laryngeal Procedures	Everyone
1.2	Paper Outline	Everyone
2.1	Gathering of Requirements	Andrew Gulotta
2.2	Robot Design	Andrew Gulotta
2.2.1	CAAR Manipulator	Andrew Gulotta
2.2.2	Manufacturing of the End Effector	Andrew Gulotta
2.2.3	Design and Construction of the Actuation Unit	Jesse d'Almeida
2.2.4	Circuit Design and Components Selection	Jesse d'Almeida
2.3	Kinematic Model	Nicholas Pacheco
2.3.1	Forward Kinematics	Nicholas Pacheco
2.3.2	Inverse Kinematics	Nicholas Pacheco
2.4	Software Design	Joe Bartone
2.4.1	Implementation of Inverse Kinematics	Joe Bartone
2.4.2	State Machine Design	Joe Bartone
2.4.3	Visualization Design	Joe Bartone
3	Testing and Validation	Everyone
3.1	Actuation Unit Forward Kinematics Test	Andrew Gulotta Nicholas Pacheco
3.2	Manual Forward Kinematics Test	Andrew Gulotta
4.1	Technology Readiness Level	Joe Bartone

4.2	Impact of COVID-19	Joe Bartone
4.3	Kinematics	Nicholas Pacheco
4.4	End Effector	Andrew Gulotta
4.5	Actuation Unit	Joe Bartone
4.6	Visualization	Joe Bartone