Project Number: 34561

A Robotic Platform for Neuro-Intervention

A Major Qualifying Project Report: Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

Submitted by:

Alexander Masiero

Maria Aranda Ramirez

Luka Christianson

Edward Flanagan

Tyler Brown

Submitted: April 25th, 2024

Approved by:

Professor Yihao Zheng, Project Advisor

Professor Ziming Zhang, Project Co-Advisor

Table of Contents

List of Figures

List of Tables

Abstract

Neurointerventional procedures involve a guidewire and a system of catheters that are maneuvered through the circulatory system into the brain. Due to the procedure's precise nature, X-rays enable practitioners to see the catheter inside the patient, leaving them vulnerable to excess radiation even with protective lead clothing. This paper explores the development of a proof-of-concept prototype for remotely controlling modularized telescoping catheters ensuring rotational and linear accuracy and compatibility with various catheter sizes. Several prototypes were developed, systematically refining the model through testing and further literature research. While improved manufacturing techniques and electronics would be required to get the necessary precision, the final iteration showed that this concept for telescoping catheter control is viable. Recommendations for future work are discussed, and this project serves as a strong starting point for future endeavors that will lead to improvements in the field of robotic neurointervention.

Introduction

Our goal was to develop a robotic platform with intuitive controls that allows remote operation and thus prevents interventionalists from exposure to radiation and wearing 5-7 kilos of lead clothing. The device will control catheters of varying sizes with 2 degrees of freedom using the same mechanism as the interventionalists have been trained on. We began this process by designing several different test prototypes to develop a more in-depth understanding of our manufacturing capabilities and the best design to move forward with. This gave us insight into the best way to achieve a final proof of concept design. We hope this device provides a more intuitive platform interventionalists can operate with and removes the need for them to be in the operating room.

Background

Neurointerventional procedures, also known as neural-specific endovascular surgeries, interventional neurology, and neurontervention radiology (Calixte et al., 2023) are minimally invasive, cutting-edge technologies that have significant advantages over traditional operating techniques. They are closed approaches that solve a wide range of neurological issues. The benefits are plentiful - including fewer complications, promoting faster recovery and shorter hospital stays (Cruddas et al., 2021). This type of procedure started as early as the 1970s, pioneered by Pierre Lasjunias and Fedor Serbinenko, to treat brain aneurysms and sinus fistulas with a detachable latex balloon (Goyal et al., 2021). It became increasingly popular in the 2000s as a major randomized trial showed that endovascular coiling greatly surpassed traditional surgical clipping in effects of survival, dependency, seizures, rebleeding, subgroups, and aneurysm occlusion (Molyneux et al., 2002). The field experienced incredible growth in the 2010s and was met with growing demand from patients as interventionalists widened the issues that could be solved via this closed approach (Goyal et al., 2021).

Currently, a plethora of problems can be targeted through neurointerventional procedures. These include acute stroke, aneurysms, carotid stenosis, intracranial atherosclerotic disease, cerebral venous thrombosis, etc. (Saber et al., 2019).

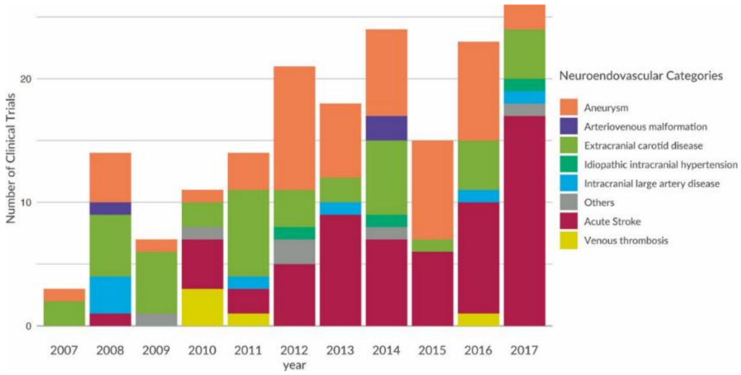


Figure 1: Neuroendovascular Categories, 2007-2017 (Saber et al., 2019)

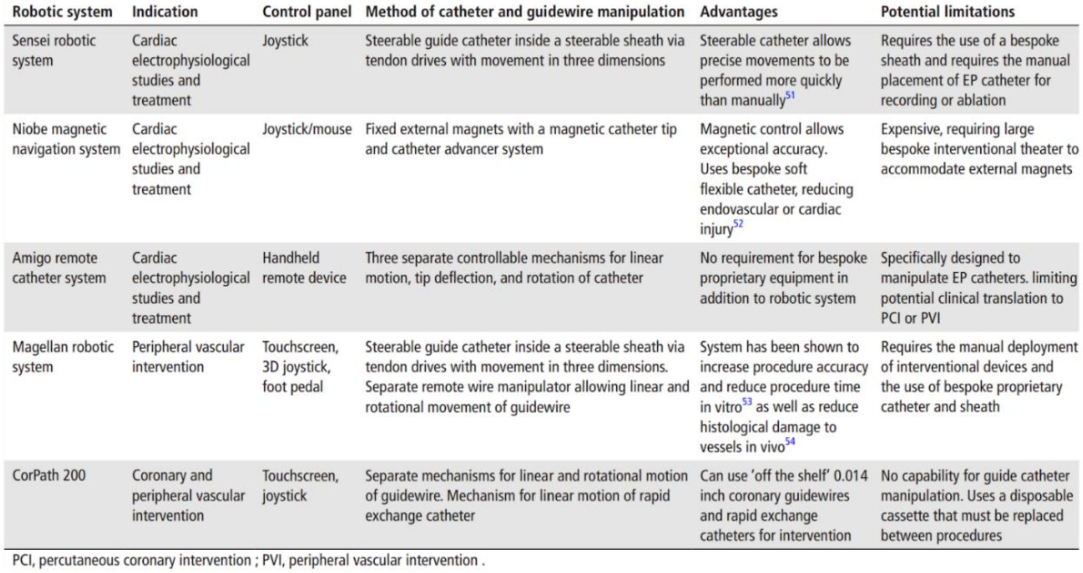
To target these issues, the procedures differ in aim and technique. Starting with thrombolytic therapy, practitioners use clot-dissolving agents to dissipate blood vessels. Endovascular coiling is another, this one specifically addressing brain aneurysms. A thin metal wire is inserted in the aneurysm, forming a coil that obstructs blood flow to the region, and prevents further dilatation. Minimally invasive spine surgery addresses several spinal disorders - fractures, tumors, compressed nerves, and other conditions causing spinal cord pressure. Cerebral angiography is a diagnostic procedure employed to visualize blood flow within the brain. By using contrast dye and imaging technology, interventionalists can assess the integrity of blood vessels and identify abnormalities. Carotid artery angioplasty/stenting is designed to address narrowed carotid arteries, which supply blood to the brain. A small balloon and/or a metallic scaffold are used to widen the narrowed artery, restoring blood flow, and thus reducing the risk of stroke. (Johns Hopkins Medicine, n.d.). Other less common procedures include embolization, mechanical thrombectomy, and radiosurgery (Rochester Regional Health, n.d.).

To be able to lead these procedures, there are a few ways endovascular training of vascular surgeons is done in the United States. The “5+2” method is a traditional training method where interventionalists get trained for an additional 2 years after their residencies. Backgrounds in neurointervention have primary specialties within neuroradiology (6-7 years of residency), neurosurgery (7 years), and vascular neurology (5-6 years) (Calixte et al., 2023). There has been an alternative training method becoming popular, called the “0+5” pathway. It is an integrated pathway of 24 months of core training tailored to vascular surgery and then 36 months specifically on vascular surgery training (Assi and Dardik, 2012). Although practitioners are becoming better trained and better equipped to navigate these complex approaches, neurointerventional procedures are not without their shortcomings.

For these procedures - fluoroscopy is used to obtain real time imaging of the anatomy. Here, X-ray beams are continually emitted to get dynamic images of the body - more resembling a video rather than single X-rays (Cleveland Clinic, n.d.). This means the interventionalists must wear protective gear to be protected from radiation exposure, resulting in 5-7 kilos of gear just to perform the procedure (Cheon et al., 2018).

Even with the protective gear - their eyes and hands are still vulnerable. According to Goldsweig “In the most comprehensive study to date, vascular surgeons at the Cleveland Clinic reviewed patient radiation doses for 2096 consecutive endovascular procedures over a 30-month period ... 52 mSv for cerebrovascular angiograms, 120 mSv for cerebrovascular interventions” (Goldsweig et al., 2017). This is notably greater than the 6.2 msV radiation dose experienced by the average American (The Regents of the University of California, n.d.). Of the procedures surpassing air kerma values of 5,000 mGy, neurosurgical procedures make up about 30.5% of those cases - with an average dose of 7,799 mGy (Bundy et al., 2020). Air kerma is the incident accumulated exposure at a site, it is used to quantify the amount of radiation energy deposited in air (Vajuhudeen, 2023). A "significant radiation dose" is defined as Ka,r > 5,000 mGy by the Society of Interventional Radiology. In cerebral angiography, operators are exposed to approximately 0.08 mSv radiation on their heads per procedure, amounting to an annual exposure of around 150 mSv, exceeding the threshold of 20 mSv per year for the lens of the eye, especially in high-volume centers (Chohan et al., 2014).

To combat this issue - robotics are now being introduced into neurointerventional procedures. This addition would allow interventionalists to be in another room while they control the mechanisms that control the catheters and guidewires into the body. In a study that assessed the effectiveness of lead aprons, the aprons offered limited protection. They shielded just 37.1% of the radiation directed toward the surgeon. However, using robotic systems for minimally invasive procedures resulted in a 62.5% reduction in fluoroscopy dose (Hyun et al., 2016).



Robotic system	Indication	Control panel	Method of catheter and guidewire manipulation	Advantages	Potential limitations
Sensei robotic system	Cardiac electrophysiological studies and treatment	Joystick	Steerable guide catheter inside a steerable sheath via tendon drives with movement in three dimensions	Steerable catheter allows precise movements to be performed more quickly than manually[51]	Requires the use of a bespoke sheath and requires the manual placement of EP catheter for recording or ablation
Niobe magnetic navigation system	Cardiac electrophysiological studies and treatment	Joystick/mouse	Fixed external magnets with a magnetic catheter tip and catheter advancer system	Magnetic control allows exceptional accuracy. Uses bespoke soft flexible catheter, reducing endovascular or cardiac injury[52]	Expensive, requiring large bespoke interventional theater to accommodate external magnets
Amigo remote catheter system	Cardiac electrophysiological studies and treatment	Handheld remote device	Three separate controllable mechanisms for linear motion, tip deflection, and rotation of catheter	No requirement for bespoke proprietary equipment in addition to robotic system	Specifically designed to manipulate EP catheters. limiting potential clinical translation to PCI or PVI
Magellan robotic system	Peripheral vascular intervention	Touchscreen, 3D joystick, foot pedal	Steerable guide catheter inside a steerable sheath via tendon drives with movement in three dimensions. Separate remote wire manipulator allowing linear and rotational movement of guidewire	System has been shown to increase procedure accuracy and reduce procedure time in vitro[53] as well as reduce histological damage to vessels in vivo[54]	Requires the manual deployment of interventional devices and the use of bespoke proprietary catheter and sheath
CorPath 200	Coronary and peripheral vascular intervention	Touchscreen, joystick	Separate mechanisms for linear and rotational motion of guidewire. Mechanism for linear motion of rapid exchange catheter	Can use 'off the shelf' 0.014 inch coronary guidewires and rapid exchange catheters for intervention	No capability for guide catheter manipulation. Uses a disposable cassette that must be replaced between procedures

PCI, percutaneous coronary intervention ; PVI, peripheral vascular intervention .

Figure 2: Commercial robotic systems used for cardiac and peripheral endovascular intervention (Crinnion et al., 2021)

As can be seen above, most of the commercially available endovascular robotic systems have a joystick and touchscreen control panel. This is non-intuitive for interventionalists and thus harder to adopt in the industry. According to Crinnion, "A controller system that allows the operator to control the robot with movements equivalent to manual procedures (advancing and rotating) would serve as an ideal platform on which an interventionalist could appreciate haptic feedback and relate it to their previous manual experience." (Crinnion et al, 2021, p. 6). Another aspect to consider is increasing efficiency and reducing procedural time as target objectives of robotic neurointerventional systems. Initial studies suggest that procedural time right now is prolonged when using these robotic systems. It is unclear whether this is due to limitations of current robotic technology or to interventionalists' limited experience and training within this area. (Beaman et al., 2021).

Previous Work

We are not the first team to tackle this issue as there have been previous teams at WPI that have worked on this task. There are two notable MQP reports from previous years, one written in 2021, and the other written the following year. The first project was focused on the operator side of the system, and the second on the patient side.

The first team was focused on creating a control system that would eliminate the need for long training processes due to different control methods. By creating a control system that mimicked the real procedure, they could reduce the additional training needed while keeping the benefits of teleoperation. Following that project, the second group sought to work on the remote robotic station that operates on the patient directly. They worked to improve the design and create a way to communicate between the operator and the system. In their final design, they used 3D-printed parts set on a linear guide with two stepper motors and one solenoid motor that control the movement functions. The two stepper motors serve to control the torque device and to drive the system along the linear guide. The solenoid motor works in conjunction with the torque-controlling stepper motor to act as the "hand" of the surgeon. The figure below shows their final design.

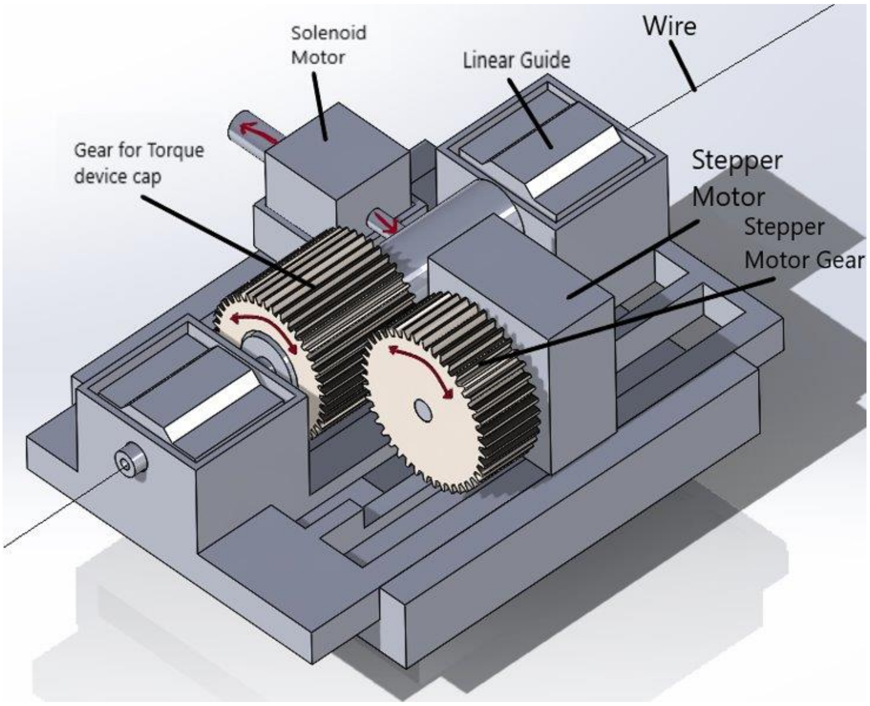


Figure 3: Previous MQP's final robotic guide wire design

Bridging the gap between the hardware and software, they used an Arduino paired with two motor drivers to reduce the load on the system. They use a VIPER system to translate the movements of the operator to the system, filtered through the ROS (Robotic Operating System), then MATLAB, and finally their Arduino. The ROS receives the data generated by the VIPER system and the operator and is then sent to MATLAB. MATLAB converts the data and transfers it to the Arduino which drives the motors to replicate the data created by the operator.

The Viper system is a product line of Polhemus. It is an Electromagnetic (EM) Tracking system that includes a sensor (microsensor), a source (EM generator), and an electronic unit. The EM generator produces a varying magnetic field, and when the sensor enters the magnetic field, a voltage is induced in the sensor based on its orientation and position. This voltage is then transferred to the electronic unit that converts the induced voltage into the sensor pose.

Another project is being completed by Rohit, a PhD candidate. He designed the initial protype for the guide wire control system, which will be operated with the catheter control system in the neurointerventional procedure. In Rohit's design, the guide wire is pushed by two motorized rollers. Smaller rollers are connected by a diagonal shaft with a gap in the center for the guide wire to run through. Each of the smaller rollers are pulled to their respective motorized rollers by springs. This system uses double helical gears to provide smooth rotation with minimal slippage between gear teeth. Slip rings are used on each end of the system to prevent wires from getting tangled and ensure accurate data.

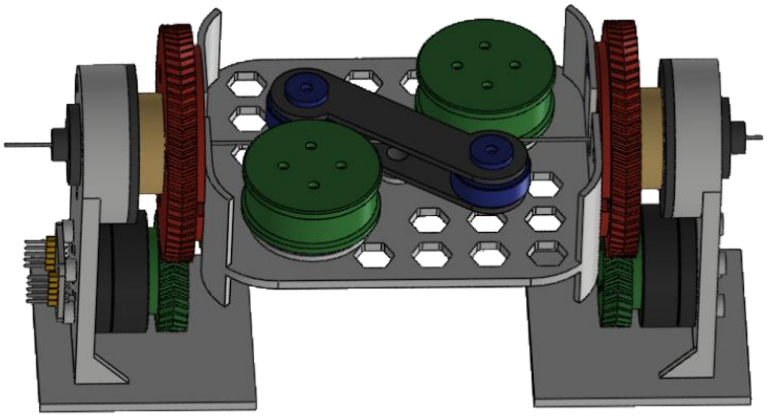


Figure 4: An early version of Rohit's design

Rohit initially used previously created code from the SimpleFOC library on the Arduino IDE. This code was recommended to him by the Makerbase company which produced the control board, the Makerbase ESP32 FOC. He decided to go with the ESP32 FOC board because he was using permanent magnet Brushless DC Motors. The Field Oriented Controls allowed for simple motor control. He initially edited this code so that it could intake two separate variables from ROS and through a Python script to Arduino IDE. The code would then prompt the motors to move. The code has a tolerance collar around the encoder values to prevent excess motor work and heat generation in the system that would extend procedure time. Together, with both the Arduino-based and python/ROS system running simultaneously, Rohit's code controls the robot based on the data collected by the Viper system and mimics the exact movement of the catheter with precision.

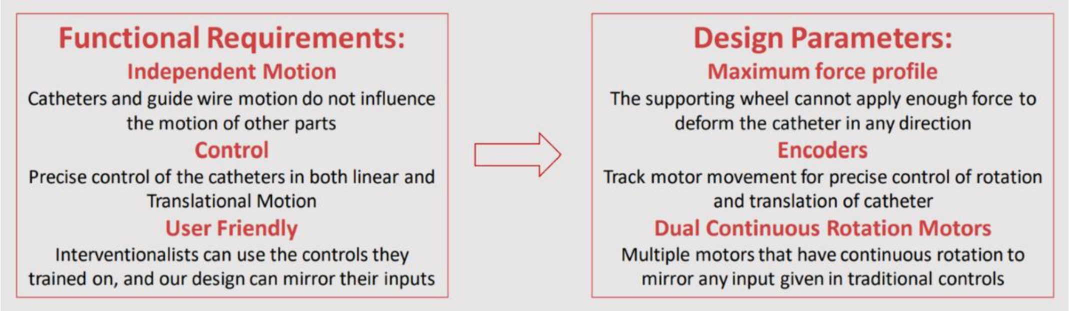
Previous MQP designs used actuators from Stepper Motors which are heavier and less smooth than the BLDC motors Rohit switched to for this and future projects. Unlike stepper motors, which have poor control over their torque and are jerky for high acceleration, BLDC can be controlled with high precision regarding torque and position using Field-Oriented Control

(FOC). It's a complicated control algorithm that requires continuous position feedback of the rotor state and 3-phase current values. Thus, custom motor drivers with built-in inline current sensing were needed to implement FOC. Rohit used the MKS DUAL FOC driver for the task. This motor driver board came with an ESP32 chip with a dual-core and supported W-iFi connection.

Methodology

Phase 1

Once our team was assigned to this project, our first step was to do a literature review. We worked with our advisors to develop functional requirements for our device. The Functional Requirements and Design Parameters we developed can be seen in the figure below.



Functional Requirements:

Independent Motion
Catheters and guide wire motion do not influence the motion of other parts

Control
Precise control of the catheters in both linear and Translational Motion

User Friendly
Interventionalists can use the controls they trained on, and our design can mirror their inputs

Design Parameters:

Maximum force profile
The supporting wheel cannot apply enough force to deform the catheter in any direction

Encoders
Track motor movement for precise control of rotation and translation of catheter

Dual Continuous Rotation Motors
Multiple motors that have continuous rotation to mirror any input given in traditional controls

Figure 5: Functional Requirements and Design Parameters for the device

We completed a Mind Map, seen below, in which we wrote down ideas and illustrated our thoughts. From this Mind Map we produced three potential designs for testing: 3-Wheel, a Straight Catheter, and a S-Shaped Catheter Design.

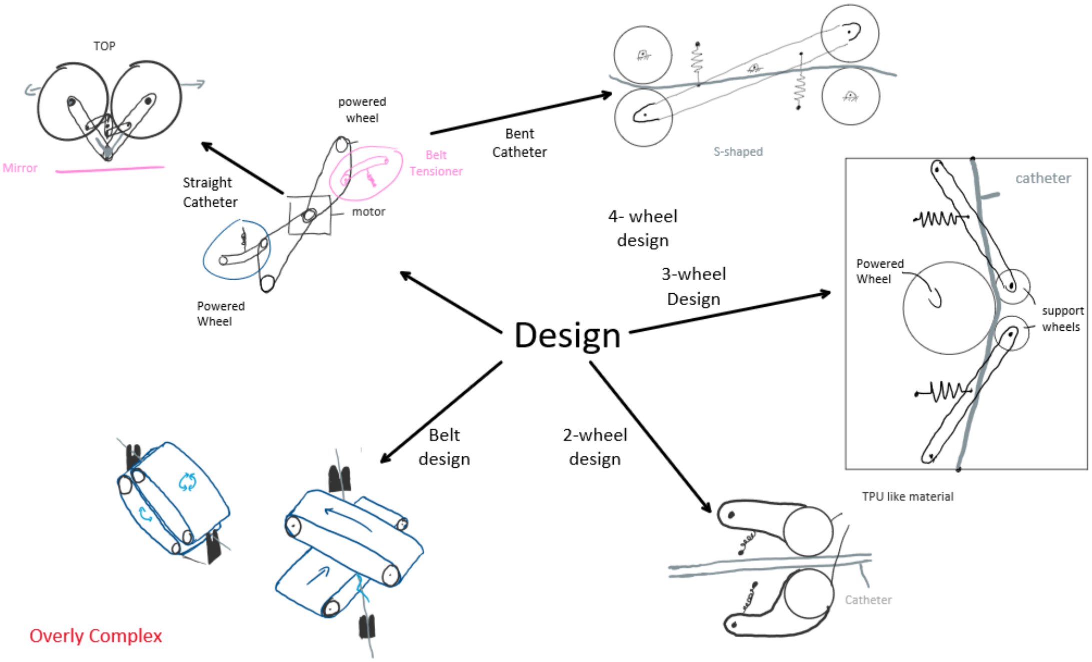


Figure 6: Mind Map

After brainstorming, we began developing the test prototypes to determine the best route to pursue. All test prototypes used wheels on a board to produce the linear motion of the catheter – these work alongside a system that rotates the entire board and wheels to generate the desired motion. We decided to design the linear and rotational systems separately.

Linear Design

This first design was the 3-Wheel Design which is pictured below. One driven wheel with two support wheels was used to move the catheter linearly. We used spring force to keep the catheter in contact with the driven wheel. This system was used to determine if having the catheter bent would help improve rotational motion. The middle wheel was adjustable, changing the angle the catheter experienced from 0 to 25 degrees. Furthermore, the system had passive wheels in the endplates that ensured the catheter would enter and exit the platform straight.

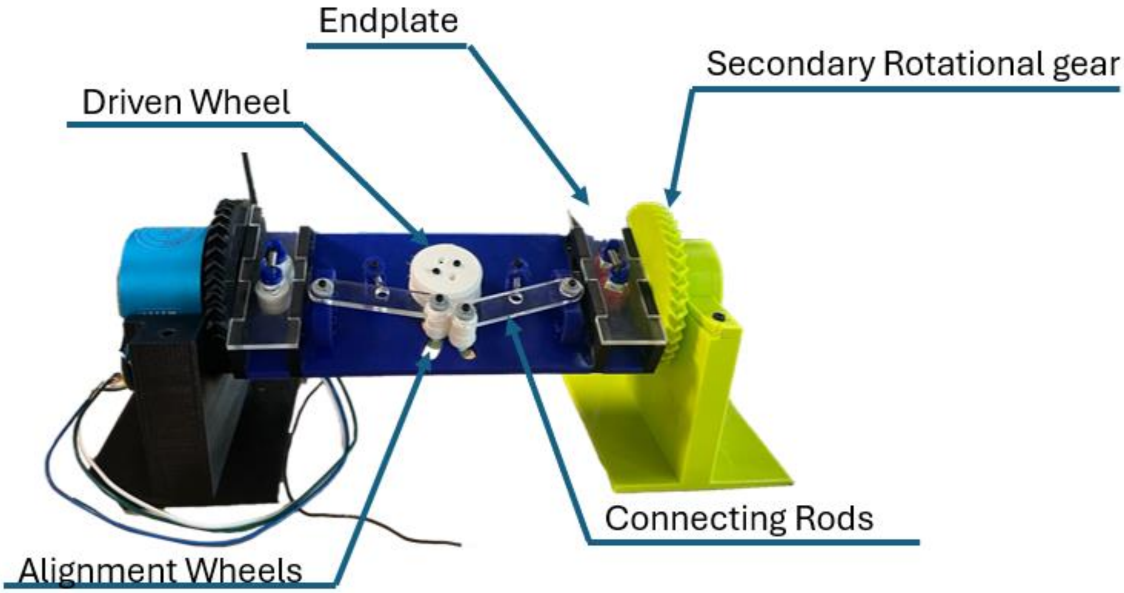


Figure 7: 3-Wheel testing prototype

This system used one centrally located driven wheel to help minimize weight and lessen the force needed to initiate rotation. Lastly, the driving wheel's central location allowed us to disregard the direction of the catheter's movement as it would experience the same forces when driven in either direction.

The second test prototype was called Straight Catheter Design as we wanted to test a system where the catheter remained straight regardless of its diameter. This system was also capable of testing a bent catheter creating an S-shape. When looking at the straight layout we had difficulties ensuring the catheter was straight. This meant we could not have the motor in a fixed position as it needed to change position based on the diameter of the catheter, without changing other parts of the setup. To get this system to work we designed a 6-bar linkage that would ensure that both sides were connected to each other.

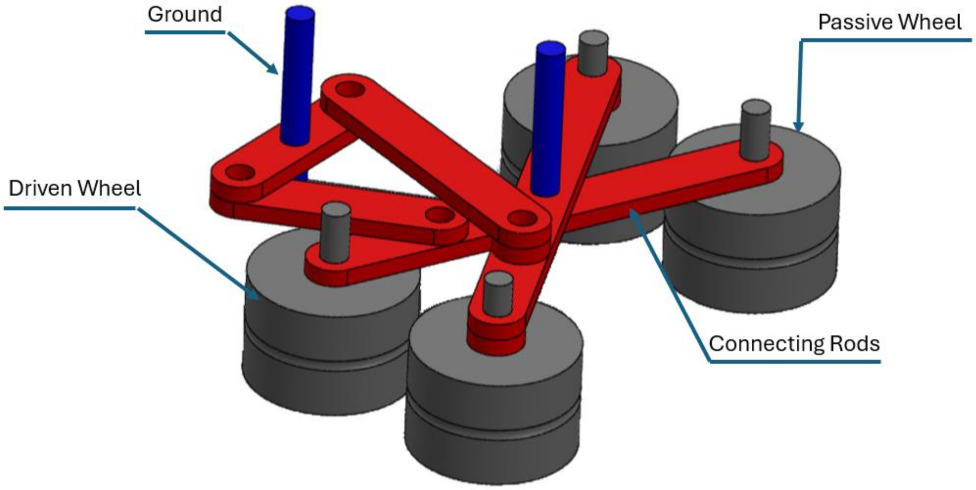


Figure 8: 6-bar linkage for straight catheter

The drivetrain on this system was also something we needed to address. There was no central motor in this design, so it was necessary to decide if we wanted to have one or two driving wheels. This design allowed us to test with both two or four total wheels. Upon testing we realized that every pair of wheels needed to have one driving wheel. We looked at potentially having one central wheel driving multiple wheels to ensure accurate motion. However, with the driven wheels needing to move, a layer of complexity that we were unable to handle was added. Our solution was to have each driving wheel have its own motor inside the rim.

Finally, looking at the S-Shaped System, the design featured four wheels, two of which would be driven like the previous design discussed. A simple bar attached to the two non-driven wheels ensured equal application of force to both. It also enabled a better grip for both linear and rotational motion, while only needing to add half the bend angle to the catheter.

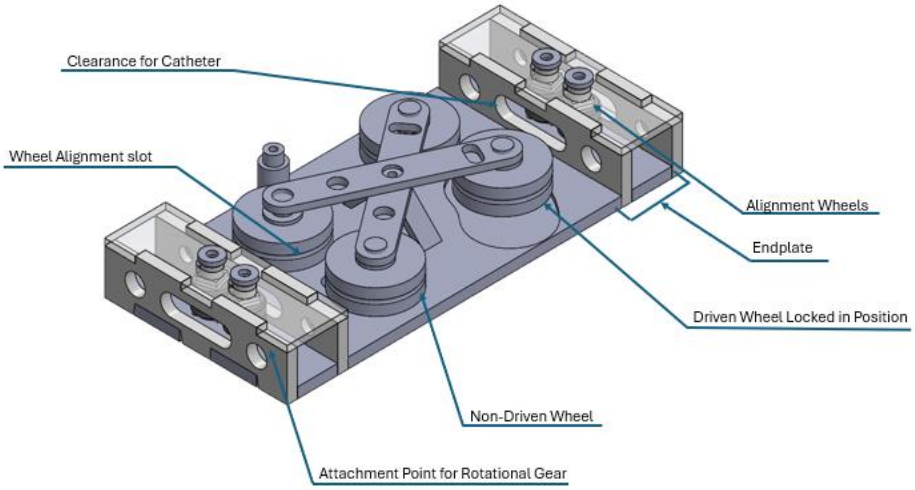


Figure 9: CAD rendering of S-shaped catheter layout of testing prototype

Results of testing linear prototypes

Both the Straight and S-Shaped Catheter Designs were too complex for our manufacturing capabilities, being primarily 3D printed. We noticed that the friction in the system was much larger than anticipated and resulted in several wheels being stuck in place, not operating as designed. The straight catheter layout was unable to rotate the catheter without slippage unless there were more than six wheels. The S-Shaped design rotated the catheter but had excessive friction compared to the 3-Wheel design, and the complexity of having two motors controlling the same output added unnecessary issues.

The 3-Wheel Design had the least friction in the system, and through testing, we determined that having the catheter bent at 10 degrees relative to its linear motion was optimal. This system had the least complexity and was the simplest to realize, while still meeting the Design Parameters.

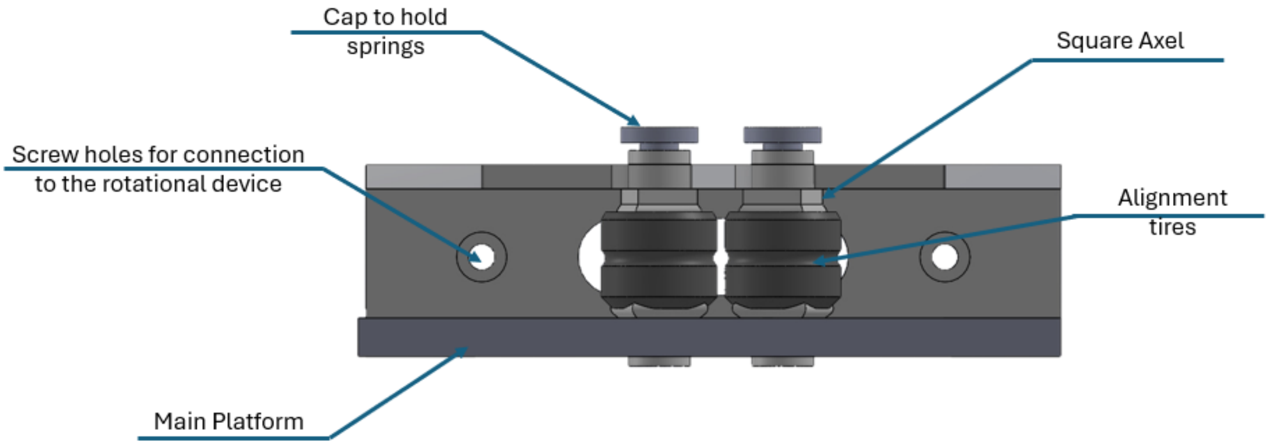


Figure 10: Alignment wheels and endplate

Rotational Design

Looking at the rotational motion, we needed to offset the motor placement to allow for a slip ring. We debated on whether to use a gear or belt drive system. However, we eliminated the belt system idea as it would be too complex and introduce the potential of slip and inaccuracy. We designed our own gears and 3D printed them to ensure proper meshing. We used double helical gears to reduce noise, making it more accurate, and a simple 2:1 gear ratio for easy adjustment. The gears and the platform supporting them were designed based on the metrics of the slip ring and motors we sourced, as seen below.

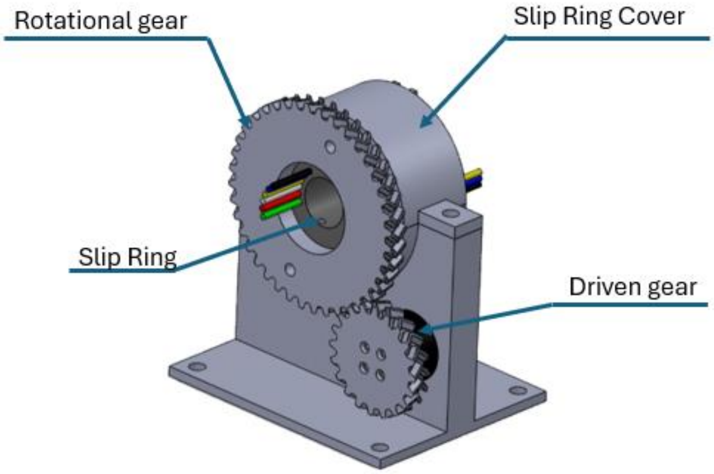


Figure 11: CAD rendering of our first rotational design

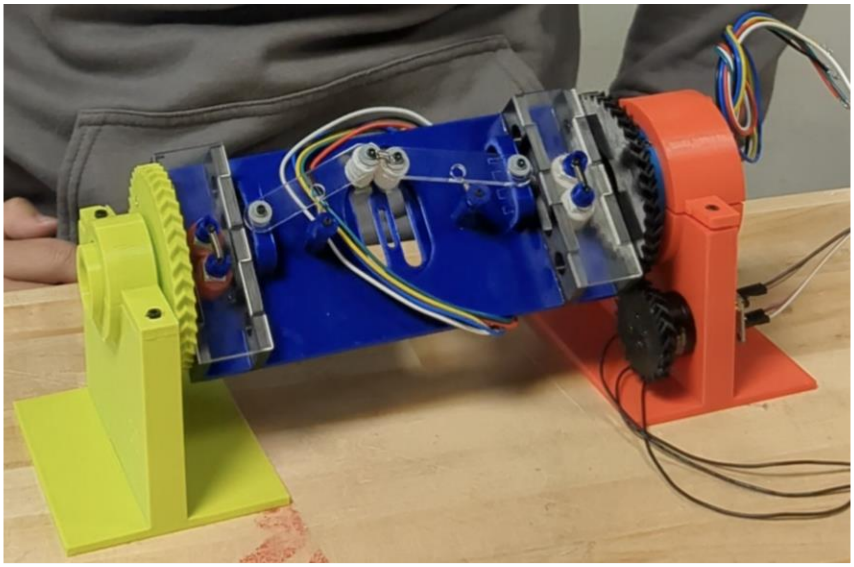


Figure 12: Completed test prototype for 3-Wheel Design

Wheels:

When designing wheels for driving the catheter, we looked for a material that would accommodate various catheter sizes. We wanted to ensure that it would apply sufficient force to the catheter to move it along the system without compromising its structure. We had access to Thermoplastic Polyurethane (TPU) which can be 3D printed and had the desired qualities. This allowed us to print and test several different ratios of wheel to tire, to try and accomplish the goal.

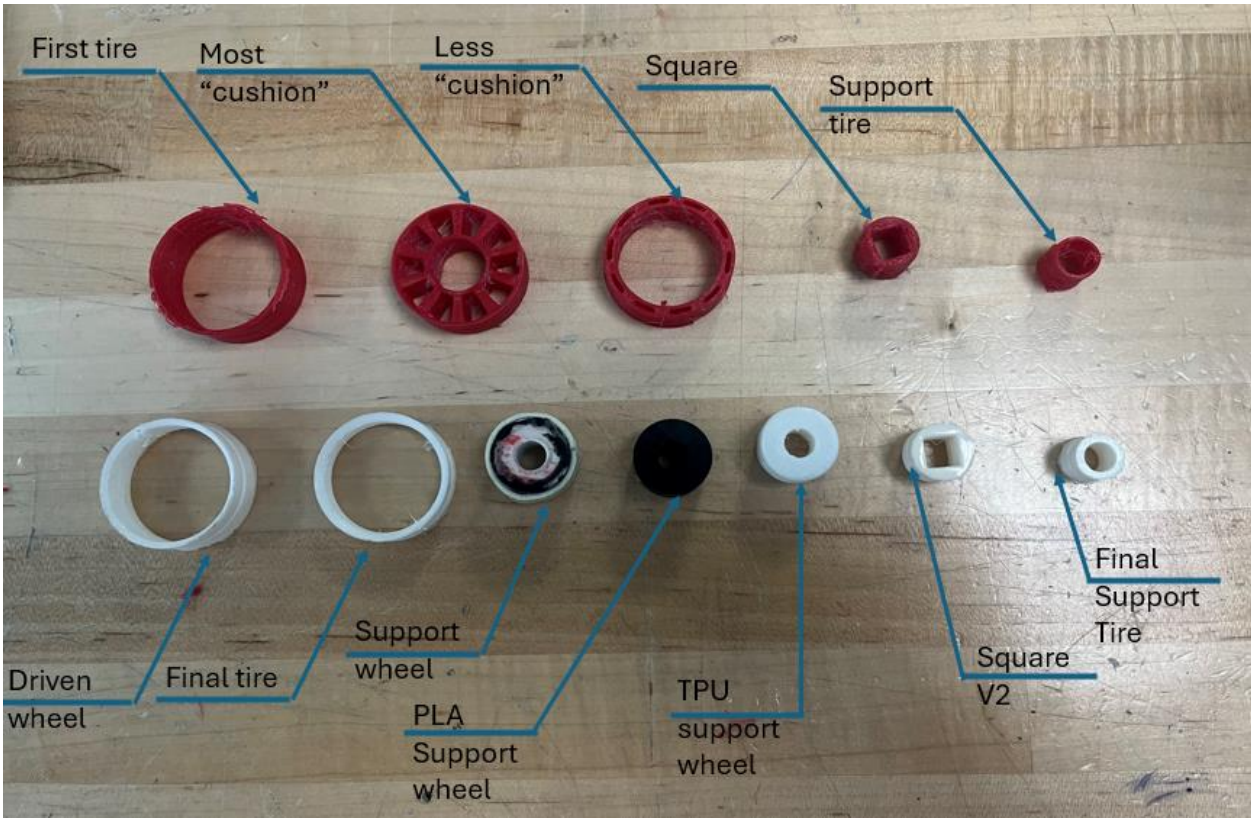


Figure 13: Different tire designs from our initial testing

After we printed and tested the wheels, we realized they were too rigid. We addressed this by modifying the slicing profile. By reducing the infill percentage from 25% down to 12.5% and changing the thickness of the walls from 0.8mm to 0.5mm, these wheels were more flexible and were able to conform to the shape of the catheter. However, the surface on the TPU was too smooth and did not apply the needed friction. The catheter would slip instead of being pushed along the system. To mitigate this issue, we first tried a rubberized coating. This improved the grip, but not enough to fully solve the issue. Instead, we tested a foam mounting tape that improved the contact area between the catheter and the wheels.

Axles:

While designing the wheels, we also needed to design the wheel axles. Originally, we assumed that the TPU wheels would be able to stretch around the axles like a sleeve. The TPU wheels were not as flexible as originally thought and could not fit properly over the axles. To solve this issue, we split the rim into two separate pieces. The bottom part had a catch to ensure the tire was centered correctly and the notches lined up. We then screwed on the top part to further stabilize the location of the wheel. This increased their strength and accuracy as they were not cantilevered but supported on both sides.

Phase 2

After testing with our initial full prototype, we identified the items we wanted to improve. Firstly, the rotational system needed to be more compact. We got a new slip ring that would better suit our needs, so adjustments and improved wire management were needed for this. The linear drive system needed to have a fixed motor at 10 degrees, and reduced friction in the system. Both systems struggled with the accuracy of the magnet placing on the wheel for the encoder and faced problems with the motors overheating.

Rotational Drive System

The new slip ring was of equivalent size to the old one, so minimal adjustments had to be made. While remodeling the gear, we added slots for the wires to be threaded through. One large slot was placed through the gear itself, and two smaller slots were placed on each side of the shaft for the slip ring. The platform’s dimensions were modified to fit the new slip ring. While making these changes, we cut down the size of the platform.

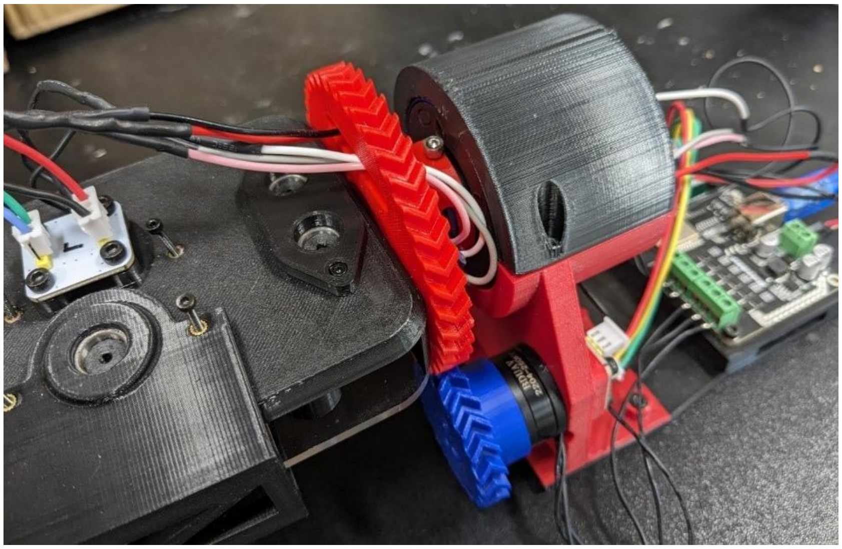


Figure 14: Platform of device

As mentioned before, an issue that we encountered was inconsistent magnet placement on the motor shaft for the encoder. Due to imperfections in 3D printing, the surface of the shaft was not level and resulted in the magnet being off center and placed at an angle. We attempted to fix this by designing a centering and leveling fixture that would be put on the shaft. However, the size of the part was so small, we were unable to fabricate it. Instead, we used fine grit sandpaper to even out the shaft. Afterwards, we designed a holder for the encoder that let us manage its height to match the magnet's location.

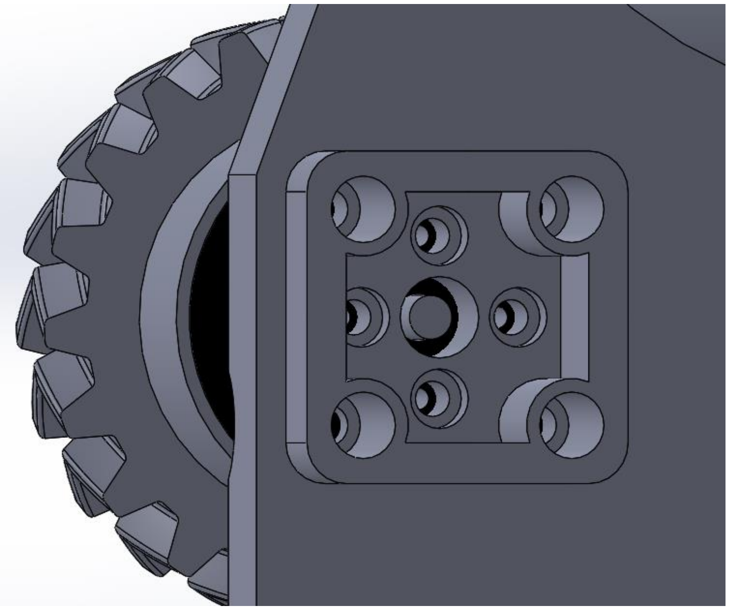


Figure 15: Image of CAD modification to platform

While testing the motors, several PLA parts connected to them faced issues with melting when the motors overheated. We reprinted these parts in ABS to increase their heat resistance, and this was sufficient to mitigate the problem.

Linear Drive System

The primary issue with our original system in phase one was excessive friction on the axles. We added bearings on all the axles to reduce the friction in the system. In addition to incorporating bearings, we designed a trolley for the central support wheel. The wheel was fixed in place, and the trolley slid on three slots, held to the central driven wheel by springs. This allowed the support wheel to adapt to the changing size of catheters while continually applying the needed force. Finally, we adjusted the CAD to fix the center wheel. This ensured that the catheter angle was held at 10 degrees.

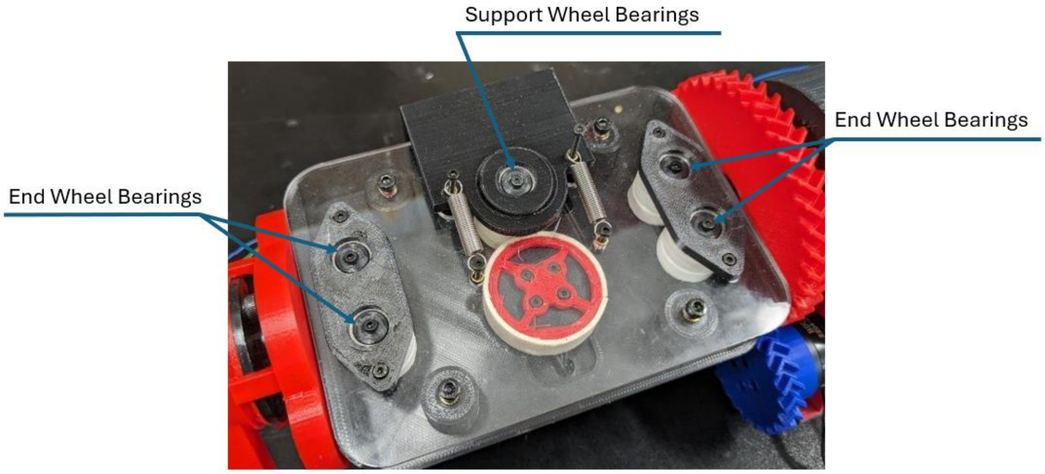


Figure 16: Bearings and support wheel trolley

Final Prototype

The final prototype only had a few modifications from Phase Two. The goal was to make this system more user-friendly and to minimize setup process. The top of our center platform for the linear system was changed to clear acrylic to improve visibility within the system. The trolley is removable, allowing easy access to the central motor for setup, and small holes were made in the support pillars to ensure that all fasteners could be easily accessed. Lastly, the wheels on the motors were changed to ABS for better thermal properties.

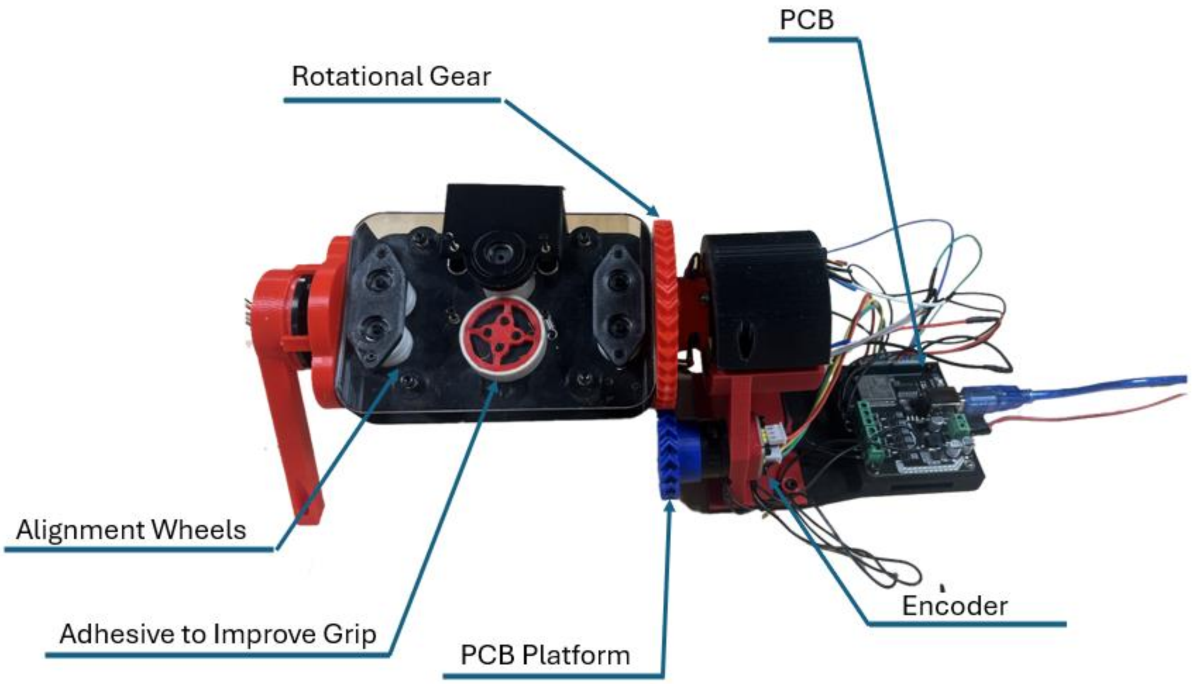


Figure 17: Final prototype

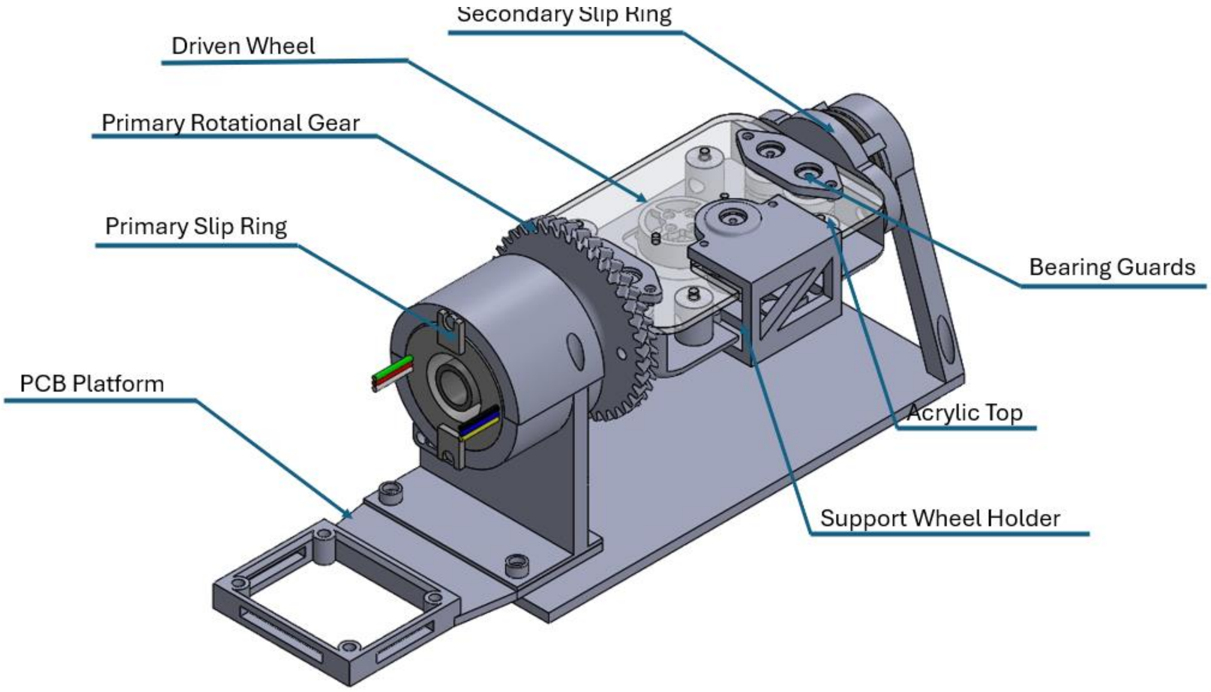


Figure 18: CAD model of final prototype

Software

Initially, we started with base code given to us by Makerbase through GitHub. This code provided board input values to help communicate data to the motors and encoders. We started to manipulate the base code and were able to have the motors spin at different speeds controlled by a variable input that could be modified during use. This allowed us to switch directions and see the power output of the motors. Due to the way the velocity is controlled, PID (Positional Integral and Derivative) controls were not implemented.

Position driven encoder value controls came with many challenges. The effort calculations for the motor were controlled solely by the FOC library. We could only manipulate the motors through two sets of PID values, angular and velocity. Through testing, we found both sets of values controlled the output of the motor, although the velocity controls worked better than angular. We continued to manipulate the values to get a working set that could power the motors enough to achieve the required motion. Our motors were on the lower end of the torque threshold required to manipulate the robot, which resulted in difficulty finding working values. This resulted in higher values which created control difficulties and higher temperatures. Additionally, there was no tolerance collar built into the library control algorithm, which kept the motor under constant load and produced excessive heat. Without access to the effort calculations in the library, a collar was constructed by setting the PID control variables to 0 when the actual encoder value was within 0.05 radians of the goal value.

We had some issues with receiving inaccurate data from the encoder. For example, the values would stay in the range of 0-6.28 or 6.28-12.56, occasionally switching between sets similar to this pattern. Additionally, random values showed up in the data rather than the stream of values linearly changing with motor movement. We deduced this was due to unsecured wired connections, so we soldered wires to prevent this. When we attached the wires through the slip ring, we came across similar issues, mainly with the encoder giving the same inaccurate values. Since nothing else had changed, we realized a flaw in the initial design. The data to the motor and data from the encoder were getting crossed causing errors. We ordered another slip ring to attach to the opposite side of the device and ran the data wires of the encoder through this slip ring.

After this was resolved, we began working on controlling both motors continuously with separate goals. We switched away from Arduino IDE to Virtual Studio (VS) Code to have a Python script running simultaneously with our controls that would send changing goal values to the motors. We utilized PlatformIO to run the Arduino code in VS Code. We leveraged the capabilities of our ESP32 Dev board to turn the system into a truly teleoperated system, as all the communication was now conducted via Wi-Fi. To synchronize the motors and EMT sensor data and ascertain minimum data loss with proven data transfer protocol, we shifted our framework entirely to the ROS2 framework with micro-ROS running in ESP32 microcontroller. The master system (Laptop) ran ROS2, which interfaced with the EMT sensor and with the ESP32 chip running micro-Ros over Wi-Fi. Through the common framework, all the sensors can exchange data independently, which greatly simplifies the process.

Part Supply and Cost Analysis

This section serves as a catalog of the part selection during the project. The table below is the full bill of materials. The parts are listed in chronological order and the total MQP cost is generated. In the second table, the cost per device is calculated.

Name	Cost	QTY	Total Cost
Screws Kit	$ 21.99	2	$ 43.98
BLDC Motors	$ 8.00	4	$ 32.00
Encoders	$ 16.99	1	$ 16.99
Spring Kit	$ 17.99	1	$ 17.99
6 Wire Slip Ring	$ 69.50	1	$ 69.50
MKS ESP32 FOC	$ 25.99	2	$ 51.98
Heat Inserts	$ 17.89	2	$ 35.78
Power Supply	$ 7.77	2	$ 15.54
Encoders	$ 15.99	2	$ 31.98
12 Wire Slip Ring	$ 75.85	2	$ 151.70
Bearings	$ 9.49	5	$ 47.45
PLA Filament	$ 13.99	1	$ 13.99
BLDC Motors	$ 13.69	4	$ 54.76
Rubber Coatings	$ 20.66	1	$ 20.66
Tape	$ 6.33	1	$ 6.33
WPI Makerspace Budget	$ 100.00	1	$ 100.00
Mounting Tape	$ 13.00	1	$ 13.00
Wire Connectors	$ 15.99	1	$ 15.99
Slip Ring	$ 30.14	1	$ 30.14
Total MQP Cost	$ 769.76		

Table 1: Bill of materials

Cost per Device			
Part Name	**Cost Per Part**	**Parts in Device**	**Total Cost**
MKS ESP32 FOC	$ 25.99	1	$ 25.99
Encoders	$ 2.66	2	$ 5.32
Bearings	$ 1.89	10	$ 18.90
12 Wire Slip Ring	$ 75.85	1	$ 75.85
Power Supply	$ 7.77	1	$ 7.77
BLDC Motors	$ 13.69	2	$ 27.38
Slip Ring	$ 30.14	1	$ 30.14
Total Device Cost	$ 191.35		

Table 2: Part list per device

Broader Impact

Engineering Ethics

Throughout the project, we remained committed to the principles outlined in the Code of Ethics of Engineers, particularly considering the device's potential impact on people's health. As we researched and designed the device, we constantly considered its broader societal, environmental, professional, and economic implications. With the sole purpose of serving and protecting people, we adhered to the Code of Ethics of Engineers at every step of the project.

Societal and Global Impact

Artificial Intelligence can assist in these types of procedures in the future. By integrating an AI into the software, it can be used to assist in the procedure (Jiang et al., 2023). This would reduce the procedure time thus eliminating interventionalist fatigue that can lead to human error.

As our system is designed for remote use, the entire procedure can be performed from a different location. By eliminating the need for the interventionalist to be present in the room when doing the operation, it eliminates radiation exposure and the need for heavy protective lead clothing.

Despite women being encouraged to join medical fields and now making up a larger percentage of medical students, they only represent 10% of practicing neurointerventionists (Bell, 2023). In a survey of almost 300 professionals, almost half participants stated no female neurointerventionalists worked in their center (Power et al., 2022). According to Yoon and Slessinger, "United States Nuclear Regulation Commission (USNRC) also recommends total fetus exposure during pregnancy to be less than 5.0 mSv (500 mrem) ... (In Angiography) A physician performing fluoroscopy should utilize essential dose reducing techniques including pulse fluoroscopy instead of continuous fluoroscopy, last image hold rather than full exposure, and colimitation to appropriate field of view." (Yoon and Slessinger, 2024). In a similar field that uses fluoroscopy, according to Weyland "The risk to develop breast cancer and all-cause cancer was shown to be fourfold higher among female orthopedic surgeons regularly experiencing radiation exposure during fluoroscopic procedures in their career" (Weyland et al., 2023). Not only is radiation a concern, but the protective gear also places greater physical strain on the musculoskeletal structure of female practitioners. By allowing the interventionalists to be another room with this device, limitations that prevent women from joining the field are eliminated.

Environmental Impact

The device is intended for single use in operations. However, we firmly believe that the potential to save lives through the development of this platform far outweighs any negative environmental impacts. By considering the product life cycle, there are ways to enhance sustainability. We can opt to source the components from vendors with environmentally friendly practices and recycle the plastics used in the project. Since the platform can conform to various catheter sizes, it can eliminate the need for extra devices. Future work will focus on exploring methods to make the platform more environmentally safe.

Codes and Standards

Our device is designed to work in tandem with a system that mimics the mechanism the interventionalists are trained on, reducing the need for additional training. Many of the systems on the market aiding remote interventional procedures utilize a joystick system, as mentioned in the background. However, this deviates from the control scheme interventionalists are trained on. This forces them to adapt to a non-intuitive interface which may increase procedure time and deter their use of the platform.

Economic Factors

Though the project had somewhat high initial part costs, the device itself has minimal expenses and assembly. It is easily manufacturable and replicable. We do not believe economic factors will be a hurdle for its adoption in the market. The device being affordable makes it easier for people from different financial backgrounds to access procedures. Despite initial costs, prioritizing the improvement of interventionalist's health and the healthcare system justifies the investment.

Future Work

This project serves as a proof of concept and shows potential for improvement. Future projects can work on reducing the size of the device to create a more streamlined product. This can be accomplished using better manufacturing techniques such as high tolerance milling and molding. This improvement in dimensioning will allow for the entire device to be made smaller and with improved accuracy.

Throughout the project we tested multiple solutions for improving the grip between the catheter and the main driving wheel. This included looking into different materials such as coatings and tape. Eventually, we settled on using a double-sided mounting tape to grip the catheter. While this works for a proof of concept, future projects will need to devise a more permanent and medical-grade solution.

This project was focused on the design and construction of the system's hardware. One opportunity for future work would be to reduce the latency between the VIPER system and the hardware. Another opportunity could include sourcing motors with more power to improve performance and continuing to refine the software. Ideally, the code would include multiple feedback loops to increase control and provide tactile feedback to the interventionalist. Force feedback loops could be introduced to monitor the effort exerted by the motors, if excessive force is needed—then they could be signaling that an artery is about to be punctured and the device would emergency stop. Additionally, this loop could be used to apply counterforce to the interventionalist's control device, mimicking the force required to manipulate the catheter.

Conclusion

This project focused on the development and testing of a prototype for a robotic platform for neurointerventional procedures. The goal is to reduce fatigue and radiation exposure. After testing and refinement, the final prototype showed the concept's viability for achieving rotational and linear accuracy while accommodating various catheter sizes. More accurate manufacturing techniques and electronics are necessary to achieve the precision required for clinical application. This platform lays out the groundwork for further development of robotic systems in this field.

References

Assi, R., & Dardik, A. (2012). Endovascular Training of Vascular Surgeons in the USA. Annals of Vascular Diseases, 5(4), 423–427. https://doi.org/10.3400/avd.ra.12.00077

Beaman, C. B., Kaneko, N., Meyers, P. M., & Tateshima, S. (2021). A Review of Robotic Interventional Neuroradiology. AJNR: American Journal of Neuroradiology, 42(5), 808–814. https://doi.org/10.3174/ajnr.A6976

Bell, J. (2023, March 20). Female physicians in neurointerventional surgery: Breaking down barriers. NeuroNews International. https://neuronewsinternational.com/female-physicians-neurointerventional-surgery-breaking-down-barriers/

Bundy, J. J., McCracken, I. W., Shin, D. S., Monroe, E. J., Johnson, G. E., Ingraham, C. R., Kanal, K. M., Bundy, R. A., Jones, S. T., Valji, K., & Chick, J. F. B. (2020). Fluoroscopically-guided interventions with radiation doses exceeding 5000 mGy reference point air kerma: A dosimetric analysis of 89,549 interventional radiology, neurointerventional radiology, vascular surgery, and neurosurgery encounters. CVIR Endovascular, 3(1), 69. https://doi.org/10.1186/s42155-020-00159-6

Calixte, A., Lartigue, S., McGaugh, S., Mathelier, M., Patel, A., Siyanaki, M. R. H., Pierre, K., & Lucke-Wold, B. (2023). Neurointerventional Radiology: History, Present and Future. Journal of Radiology and Oncology, 7(1), 026–032. https://doi.org/10.29328/journal.jro.1001049

Cheon, B. K., Kim, C. L., Kim, K. R., Kang, M. H., Lim, J. A., Woo, N. S., Rhee, K. Y., Kim, H. K., & Kim, J. H. (2018). Radiation safety: A focus on lead aprons and thyroid shields in interventional pain management. The Korean Journal of Pain, 31(4), 244–252. https://doi.org/10.3344/kjp.2018.31.4.244

Chohan, M. O., Sandoval, D., Buchan, A., Murray-Krezan, C., & Taylor, C. L. (2014). Cranial radiation exposure during cerebral catheter angiography. Journal of NeuroInterventional Surgery, 6(8), 633–636. https://doi.org/10.1136/neurintsurg-2013-010909

Crinnion, W., Jackson, B., Sood, A., Lynch, J., Bergeles, C., Liu, H., Rhode, K., Mendes Pereira, V., & Booth, T. C. (2021). Robotics in neurointerventional surgery: A systematic review of the literature. Journal of NeuroInterventional Surgery, neurintsurg-2021-018096. https://doi.org/10.1136/neurintsurg-2021-018096

Cruddas, L., Martin, G., & Riga, C. (2021). Robotic endovascular surgery: Current and future practice. Seminars in Vascular Surgery, 34(4), 233–240. https://doi.org/10.1053/j.semvascsurg.2021.10.002

Endovascular Neurosurgery and Interventional Neuroradiology. (2024, April 16). Johns Hopkins Medicine. https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/endovascular-neurosurgery-and-interventional-neuroradiology

Fluoroscopy: What It Is, Purpose, Procedure & Results. (n.d.). Cleveland Clinic. Retrieved April 23, 2024, from https://my.clevelandclinic.org/health/diagnostics/21992-fluoroscopy

Goldsweig, A. M., Abbott, J. D., & Aronow, H. D. (2017). Physician and Patient Radiation Exposure During Endovascular Procedures. Current Treatment Options in Cardiovascular Medicine, 19(2), 10. https://doi.org/10.1007/s11936-017-0507-9

Goyal, M., van Zwam, W., Moret, J., & Ospel, J. M. (2021). Neurointervention in the 2020s: Where are We Going? Clinical Neuroradiology, 31(1), 1–5. https://doi.org/10.1007/s00062-020-00953-8

How much radiation? (n.d.). The Regents of the University of California. Retrieved April 23, 2024, from https://knowyourdose.ucsf.edu/how-much-radiation

Hyun, S.-J., Kim, K.-J., Jahng, T.-A., & Kim, H.-J. (2016). Efficiency of lead aprons in blocking radiation − how protective are they? Heliyon, 2(5), e00117. https://doi.org/10.1016/j.heliyon.2016.e00117

Jiang, Y., Lv, J., Li, Y., & Zhang, Y. (2023). Editorial: The application of artificial intelligence in interventional neuroradiology. Frontiers in Neurology, 13. https://doi.org/10.3389/fneur.2022.1112624

Molyneux, A. J., Kerr, R. S. C., Yu, L.-M., Clarke, M., Sneade, M., Yarnold, J. A., Sandercock, P., & International Subarachnoid Aneurysm Trial (ISAT) Collaborative Group. (2005). International subarachnoid aneurysm trial (ISAT) of neurosurgical clipping versus endovascular coiling in 2143 patients with ruptured intracranial aneurysms: A randomised comparison of effects on survival, dependency, seizures, rebleeding, subgroups, and aneurysm occlusion. Lancet (London, England), 366(9488), 809–817. https://doi.org/10.1016/S0140-6736(05)67214-5

Neurointerventional Surgery. (n.d.). Rochester Regional Health. Retrieved April 23, 2024, from https://www.rochesterregional.org/services/neurosciences/neurointerventional-surgery

Power, S., Biondi, A., Saatci, I., Bennett, K., Mahadevan, J., Januel, A. C., Singhara Na Ayudhaya, S. P., & Agid, R. (2022). Women in neurointervention, a gender gap? Results of a prospective online survey. Interventional Neuroradiology: Journal of Peritherapeutic Neuroradiology, Surgical Procedures and Related Neurosciences, 28(3), 311–322. https://doi.org/10.1177/15910199211030783

Saber, H., Jadhav, A. P., Rajah, G. B., Narayanan, S., Sheth, S. A., Liebeskind, D. S., & Somai, M. (2019). Clinical trials of neurointervention: 2007–2018. Journal of NeuroInterventional Surgery, 11(12), 1277–1282. https://doi.org/10.1136/neurintsurg-2019-015117

Vajuhudeen, Z. (2023). Dose area product | Radiology Reference Article | Radiopaedia.org. Radiopaedia. https://doi.org/10.53347/rID-80058

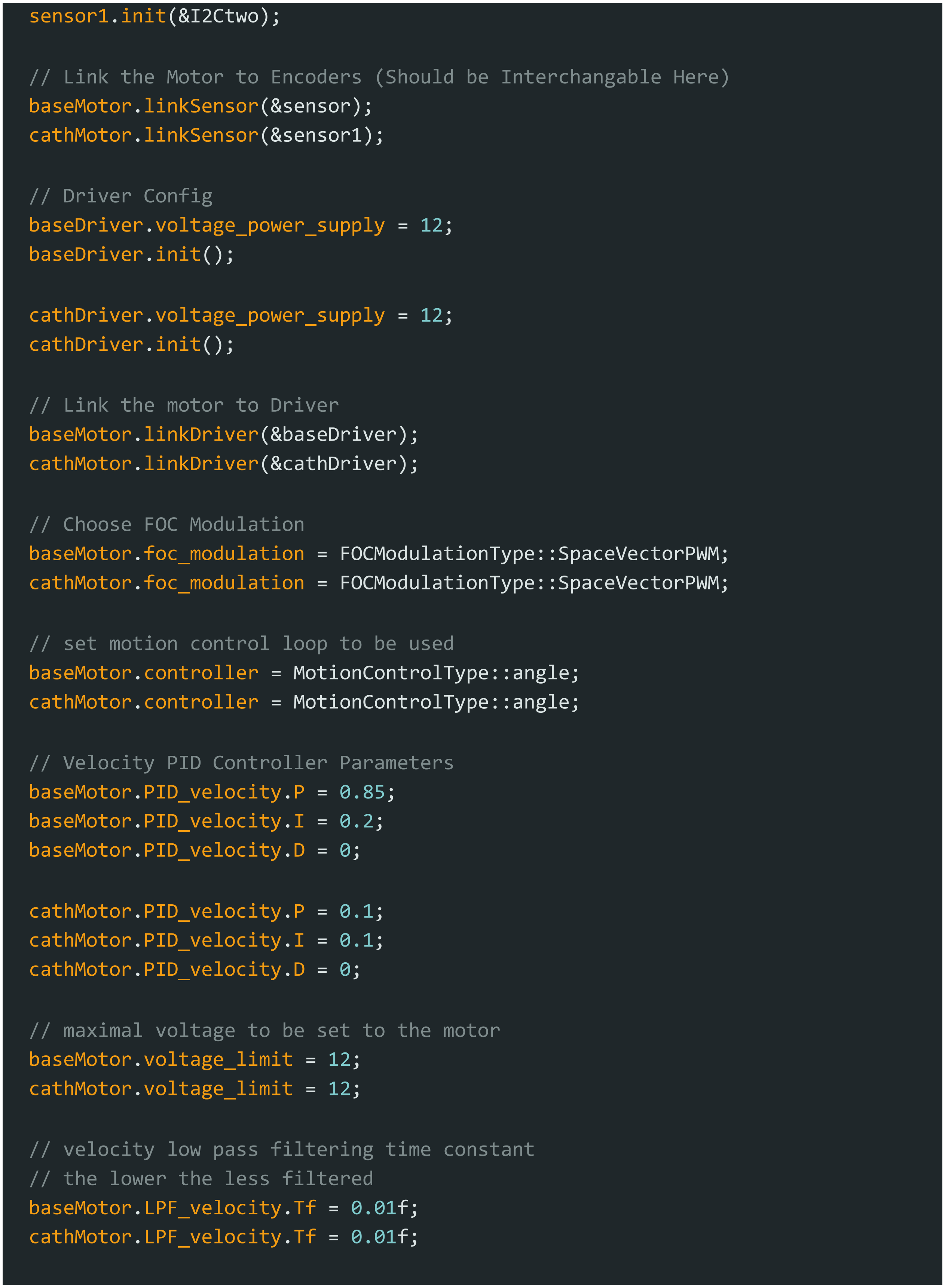
Weyland, C. S., Jesser, J., Bourgart, I., Hilgenfeld, T., Breckwoldt, M. O., Vollherbst, D., Schmitt, N., Seker, F., Bendszus, M., & Möhlenbruch, M. A. (2023). Occupational radiation exposure of neurointerventionalists during endovascular stroke treatment. European Journal of Radiology, 164, 110882. https://doi.org/10.1016/j.ejrad.2023.110882

Yoon, I., & Slesinger, T. L. (2024). Radiation Exposure In Pregnancy. In StatPearls. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK551690/

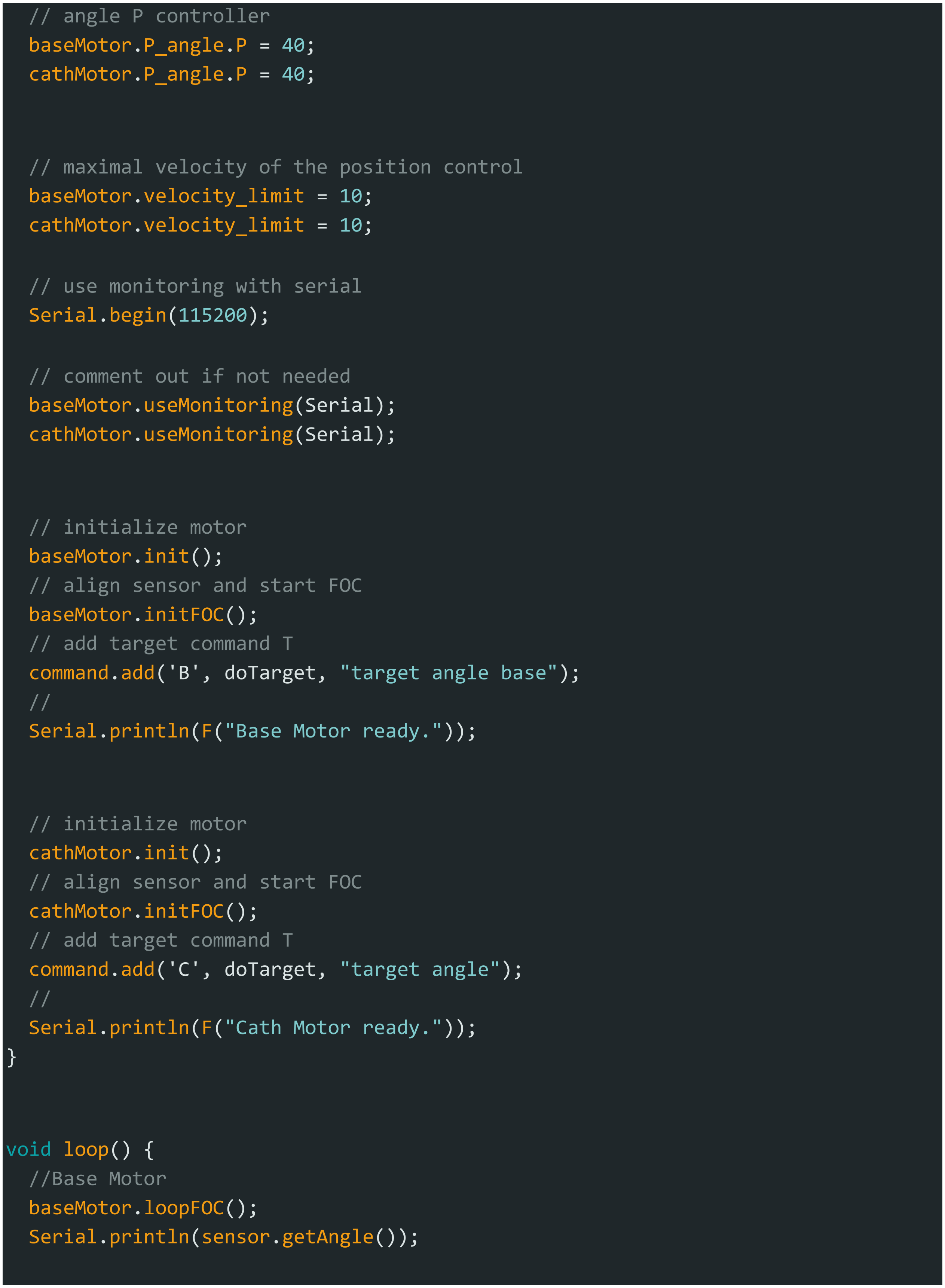
Appendix

Appendix A: Dual Motor Control Code from Arduino

```
#include <SimpleFOC.h>

// Magnetic Encoders
  //Encoder 1 (Closer to Motors)
MagneticSensorI2C sensor = MagneticSensorI2C(AS5600_I2C);
TwoWire I2Cone = TwoWire(0);

  //Encoder 2 (Closer to Power Input)
MagneticSensorI2C sensor1 = MagneticSensorI2C(AS5600_I2C);
TwoWire I2Ctwo = TwoWire(1);


// Motors (Should be Interchangable here)
  //Motor for Rotation (Closer to Edge)
BLDCMotor baseMotor = BLDCMotor(7);
BLDCDriver3PWM baseDriver = BLDCDriver3PWM(32, 33, 25, 22);

  //Motor for Catheter (Closer to Center)
BLDCMotor cathMotor = BLDCMotor(7);
BLDCDriver3PWM cathDriver = BLDCDriver3PWM(26, 27, 14, 12); //Values Work


// Starter Angle (in Radians)
float targetAngleBase = 0;
float targetAngleCath = 0;

// Instantiate the Commander
Commander command = Commander(Serial);
void doTarget(char* cmd)
{
  command.scalar(&targetAngleBase, cmd);
  command.scalar(&targetAngleCath, cmd);
}

void setup()
{
  // Initialize Encoders
  I2Cone.begin(19, 18, 400000);
  I2Ctwo.begin(23, 5, 400000); // Second Encoder Values Confirmed to Work
  sensor.init(&I2Cone);
```



```
sensor1.init(&I2Ctwo);

// Link the Motor to Encoders (Should be Interchangable Here)
baseMotor.linkSensor(&sensor);
cathMotor.linkSensor(&sensor1);

// Driver Config
baseDriver.voltage_power_supply = 12;
baseDriver.init();

cathDriver.voltage_power_supply = 12;
cathDriver.init();

// Link the motor to Driver
baseMotor.linkDriver(&baseDriver);
cathMotor.linkDriver(&cathDriver);

// Choose FOC Modulation
baseMotor.foc_modulation = FOCModulationType::SpaceVectorPWM;
cathMotor.foc_modulation = FOCModulationType::SpaceVectorPWM;

// set motion control loop to be used
baseMotor.controller = MotionControlType::angle;
cathMotor.controller = MotionControlType::angle;

// Velocity PID Controller Parameters
baseMotor.PID_velocity.P = 0.85;
baseMotor.PID_velocity.I = 0.2;
baseMotor.PID_velocity.D = 0;

cathMotor.PID_velocity.P = 0.1;
cathMotor.PID_velocity.I = 0.1;
cathMotor.PID_velocity.D = 0;

// maximal voltage to be set to the motor
baseMotor.voltage_limit = 12;
cathMotor.voltage_limit = 12;

// velocity low pass filtering time constant
// the lower the less filtered
baseMotor.LPF_velocity.Tf = 0.01f;
cathMotor.LPF_velocity.Tf = 0.01f;
```



```
  // angle P controller
  baseMotor.P_angle.P = 40;
  cathMotor.P_angle.P = 40;


  // maximal velocity of the position control
  baseMotor.velocity_limit = 10;
  cathMotor.velocity_limit = 10;

  // use monitoring with serial
  Serial.begin(115200);

  // comment out if not needed
  baseMotor.useMonitoring(Serial);
  cathMotor.useMonitoring(Serial);


  // initialize motor
  baseMotor.init();
  // align sensor and start FOC
  baseMotor.initFOC();
  // add target command T
  command.add('B', doTarget, "target angle base");
  //
  Serial.println(F("Base Motor ready."));


  // initialize motor
  cathMotor.init();
  // align sensor and start FOC
  cathMotor.initFOC();
  // add target command T
  command.add('C', doTarget, "target angle");
  //
  Serial.println(F("Cath Motor ready."));
}


void loop() {
  //Base Motor
  baseMotor.loopFOC();
  Serial.println(sensor.getAngle());
  baseMotor.move(targetAngleBase * -1);
  if(abs(targetAngleBase - sensor.getAngle()) > .05)
  {
    baseMotor.PID_velocity.P = .8;
    baseMotor.PID_velocity.I = .2;
    baseMotor.PID_velocity.D = .1;
  }
  else
  {
    baseMotor.PID_velocity.P = 0;
    baseMotor.PID_velocity.I = 0;
    baseMotor.PID_velocity.D = 0;
  }

  //Catheter Motor
  cathMotor.loopFOC();
  //Serial.println(sensor1.getAngle());

  cathMotor.move(targetAngleCath * -1);
  if(abs(targetAngleCath - sensor1.getAngle()) > .05)
  {
    cathMotor.PID_velocity.P = 0;
    cathMotor.PID_velocity.I = 0;
    cathMotor.PID_velocity.D = 0;
  }
  else
  {
    cathMotor.PID_velocity.P = 0;
    cathMotor.PID_velocity.I = 0;
    cathMotor.PID_velocity.D = 0;
  }



  command.run();
}
```

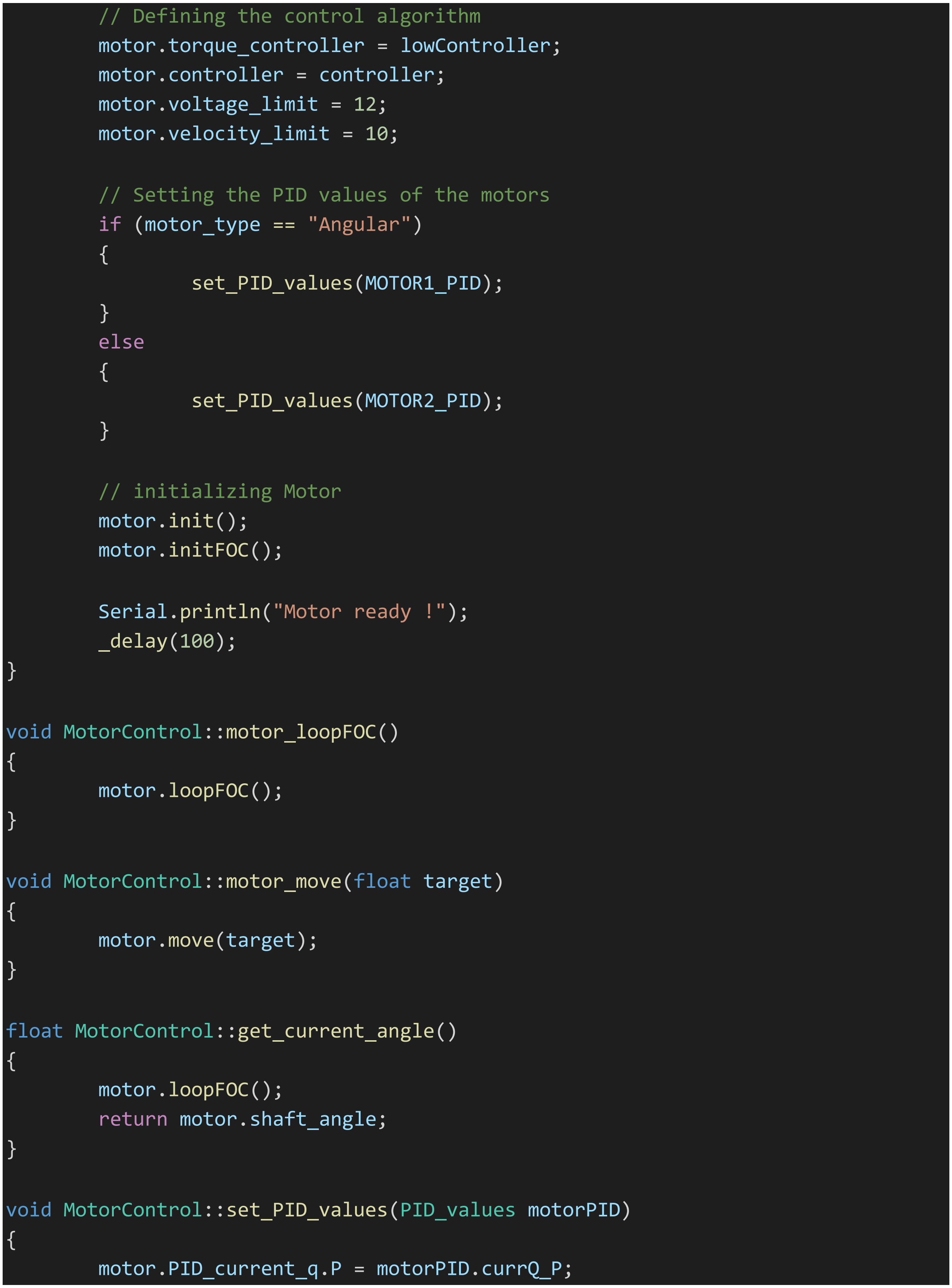
Appendix B: main.cpp from VS Code

```
#include <Arduino.h>
#include "motor_control.h"

MotorControl angular_motor = MotorControl(MOTOR1);
MotorControl linear_motor = MotorControl(MOTOR2);

```

```
TargetAngles goal_angles;
float angular_motor_angle, linear_motor_angle = 0.0;
long start = 0;

void setup()
{
  Serial.begin(115200);
  angular_motor.motor.useMonitoring(Serial);
  linear_motor.motor.useMonitoring(Serial);
  _delay(1000);

  // Initialize the Motors
  angular_motor.initialize_motor_driver_sensor(MotionControlType::angle,
TorqueControlType::foc_current, "Angular", 1.0);
  linear_motor.initialize_motor_driver_sensor(MotionControlType::angle,
TorqueControlType::foc_current, "Linear", 1.4);
  _delay(1000);
}

void loop()
{
  angular_motor_angle = angular_motor.get_current_angle();
  linear_motor_angle = linear_motor.get_current_angle();

  // Change value here manually or get the value from serial and set it (But you
have to write the code for that)
  goal_angles.angularTarget = 3.14;
  goal_angles.linearTarget = 3.14;

  angular_motor.motor_move(goal_angles.angularTarget);
  linear_motor.motor_move(goal_angles.linearTarget);

  printf("%f\n", angular_motor_angle);
}
```

Appendix C: motor_control.cpp

```
#include "motor_control.h"

MotorConfig MOTOR1 = {
    .magnetic_sensor = AS5600_I2C,
    .sensor_num = 0,
    .encoder_pins = {19, 18},
    .pole_pair = 7,
    .driver_pins = {32, 33, 25, 22},
    .inLine_Pins = {39, 36}};

MotorConfig MOTOR2 = {
    .magnetic_sensor = AS5600_I2C,
    .sensor_num = 1,
    .encoder_pins = {23, 5},
    .pole_pair = 7,
    .driver_pins = {26, 27, 14, 12},
    .inLine_Pins = {35, 34}};

PID_values MOTOR1_PID = {

    .currQ_P = 0.5,
    .currQ_I = 0.5,
    .currQ_D = 0.0,
    .currQ_Tf = 0.002,

    .currD_P = 0.5,
    .currD_I = 0.5,
    .currD_D = 0.0,
    .currD_Tf = 0.002,

    .velP = 0.1,
    .velI = 1.0,
    .velD = 0.0,

    .angP = 20.0

};

PID_values MOTOR2_PID = {

    .currQ_P = 0.5,
    .currQ_I = 2,
    .currQ_D = 0.0,
    .currQ_Tf = 0.001,

    .currD_P = 0.5,
    .currD_I = 2,
    .currD_D = 0.0,
    .currD_Tf = 0.001,
    .velP = 0.1,
    .velI = 1.0,
    .velD = 0.0,

    .angP = 20.0

};

MotorControl::MotorControl(MotorConfig MOTORC) : sensor(MOTORC.magnetic_sensor),
                                                 I2Cone(MOTORC.sensor_num),
                                                 motor(MOTORC.pole_pair),
                                                 driver(MOTORC.driver_pins[0],
MOTORC.driver_pins[1], MOTORC.driver_pins[2], MOTORC.driver_pins[3]),
                                                 current_sense(float(0.01),
float(50.0), MOTORC.inLine_Pins[0], MOTORC.inLine_Pins[1])
{

        encoder_outlet[0] = MOTORC.encoder_pins[0];
        encoder_outlet[1] = MOTORC.encoder_pins[1];
}

void MotorControl::initialize_motor_driver_sensor(MotionControlType controller,
TorqueControlType lowController, String motor_type, float gear_ratio)
{

        // Establishing Encoder Connection to Motor
        I2Cone.begin(encoder_outlet[0], encoder_outlet[1], 400000);
        sensor.init(&I2Cone);
        motor.linkSensor(&sensor);

        // Assigning the motor driver
        driver.voltage_power_supply = 12;
        driver.init();
        motor.linkDriver(&driver);

        // Assigning the current sensor to motor
        motor.current_limit = 2;
        current_sense.linkDriver(&driver);
        current_sense.init();
        current_sense.gain_b *= -1;
        current_sense.gain_a *= -1;
        motor.linkCurrentSense(&current_sense);
```



```
        // Defining the control algorithm
        motor.torque_controller = lowController;
        motor.controller = controller;
        motor.voltage_limit = 12;
        motor.velocity_limit = 10;

        // Setting the PID values of the motors
        if (motor_type == "Angular")
        {
                set_PID_values(MOTOR1_PID);
        }
        else
        {
                set_PID_values(MOTOR2_PID);
        }

        // initializing Motor
        motor.init();
        motor.initFOC();

        Serial.println("Motor ready !");
        _delay(100);
}

void MotorControl::motor_loopFOC()
{
        motor.loopFOC();
}

void MotorControl::motor_move(float target)
{
        motor.move(target);
}

float MotorControl::get_current_angle()
{
        motor.loopFOC();
        return motor.shaft_angle;
}

void MotorControl::set_PID_values(PID_values motorPID)
{
        motor.PID_current_q.P = motorPID.currQ_P;
        motor.PID_current_q.I = motorPID.currQ_I;
        // motor.PID_current_q.D= motorPID.currQ_D;

        motor.PID_current_d.P = motorPID.currD_P;
        motor.PID_current_d.I = motorPID.currD_I;
        // motor.PID_current_d.D= motorPID.currD_D;

        motor.LPF_current_q.Tf = motorPID.currQ_Tf; // 1ms default
        motor.LPF_current_d.Tf = motorPID.currD_Tf; // 1ms default

        // motor.PID_velocity.P = motorPID.velP;
        // motor.PID_velocity.I = motorPID.velI;
        // motor.PID_velocity.D = motorPID.velD;

        // motor.P_angle.P = motorPID.angP;

        motor.velocity_limit = 12;
}
```