

EN. 601.656 Computer Integrated Surgery II

# **Tele-operation Control of a High Dexterity Robot for Vitreoretinal Surgery**

**Team 21:** Xiao Wei

**Mentors:** Mojtaba Esfandiari, Adnan Munawar, Dr. Iulian Iordachita

**Date:** 10/05/2023

## Table of Contents

Introduction.....	3
Clinical motivation.....	3
Background.....	3
Goals .....	4
Technical approach.....	5
AMBF Simulation:.....	5
Teleoperation Control: .....	6
Structure .....	7
Forward kinematics.....	7
Control .....	8
Results.....	11
AMBF simulation .....	11
Teleoperation control .....	11
Management Summary .....	13
Conclusion .....	14
References.....	15

# Introduction

## Clinical motivation

Vitreoretinal surgery is a highly delicate and difficult intraocular surgery that requires the operation on the retina deep within the eye. Examples of this surgery are epiretinal membrane peeling and retinal vein cannulation. Such surgeries are very demanding, requiring exceptional precision to operate on micron scale targets presented by the retina while also maneuvering in a tightly constrained and fragile workspace <sup>[1]</sup>.

Therefore, it poses many challenges for physicians. The primary issue arises from physiological hand tremor, which will cause trouble during surgery. For example, any slight jerking of the hand formed during retinal peeling can result in retinal tears, causing irreversible blindness to the underlying retina with almost no regenerative capacity <sup>[2]</sup>. Similarly, exerting forces  $> 7.5$  mN to the retina could result in retinal tears, forces which are too fine to be felt by a human. All these factors result in an exceptional level of training that surgeons must undergo to be able to perform these surgeries <sup>[3][4]</sup>.

## Background

To resolve the above stated issues, researcher at Johns Hopkins University, AMIRO LCSR, developed a surgical robotic system, a Steady Hand Eye Robot (SHER), which performs robot-controlled surgery, allowing surgeons to mitigate (filter) physiological tremor affects and provide them with haptic force feedback. Systems like SHER can potentially address a lot of the issues that eye surgeons face in the OR. This impedance style robot (Fig. 1, B) drastically reduced the effects of tremor in surgery.

However, even with SHER, there are still some operative procedures that may not be easily executable, given the limited flexibility of the end effector. A straight needle attached to the end-effector of SHER does not provide enough dexterity for surgeons operating inside the eye, with surgeons often having to move the eye to navigate. Therefore, a 2-DoF Integrated Robotic Intraocular Snake (IRIS) (Fig. 1, C) was designed and attached to the SHER end-effector. This eliminated the need to move the eye by employing a more flexible end-effector.

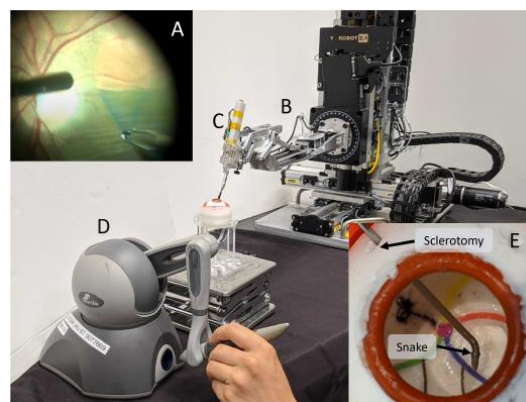


Figure 1. Envisioned high dexterity intraocular manipulator: (A) Epiretinal membrane peeling; (B) Steady Hand Eye Robot; (C) Integrated robotic intraocular snake robot; (D) Phantom Omni; (E) Distal snake-like tool-end inside eye phantom [5].

Existing approaches include teleoperation of both the snake and the eye robot, but separately, as well as a cooperative approach, which involves moving the eye robot by hand, and teleoperating the snake after moving it to the desired position. In recent work done by Kaiyu Shi <sup>[5]</sup>, the 7-DoF integrated system

allows the clinician to directly control the position and orientation of the tool tip via a 6-DoF Phantom Omni device (Fig. 1, D).

## Goals

The motivation to implement this teleoperation setup is multifold. First and foremost, teleoperating a device like the eye robot helps the user/surgeon perform stable and meticulous manipulation, which can understandably improve surgical performance. In addition, replacing the traditional stiff end effector with a snake robot can bring benefits, such as expanding the working space, enhancing flexibility, and reducing the risk of causing injury to the eye to some extent.

Therefore, the main goal of the project is to implement the simulation of the combined teleoperation system in Asynchronous multi-body framework (AMBF)<sup>1</sup>. Simulation is one of the common means of testing robots, it enables researchers to quickly develop, validate and test control algorithms without worrying about damaging the robot. To achieve the goal, the model of the snake robot should be built in Blender according to the CAD drawing and then connected to eye robot in the simulation. Meanwhile, I will develop an AMBF plugin to manipulate the snake robot attached to SHER and interact with an OCT Scan of the eye.

Once the simulation of the teleoperation system is done successfully, we aim to develop and implement a control algorithm on the AMBF model using a haptic device. Moreover, we expect to apply the controller on the real robot. Finally, we can design and execute an experimental scenario to validate our control algorithm and establish a framework for moving this system closer to clinical assimilation.

To be specific, the deliverables proposed at the checkpoint presentation are listed below:

### Minimum

- Developing a simulation model for the Eye Robot and the Eye Snake systems in AMBF
  - Learn AMBF + ADF + Blender addon
  - Learn about how to use various plugin
  - Get familiar with volume and 3D scans

### Expected

- Develop and implement a control algorithm on the AMBF model using a haptic device

### Maximum

- Implement the controller on the real robots

---

<sup>1</sup> More detailed information about AMBF is on the website <https://github.com/WPI-AIM/ambf/wiki>

# Technical approach

This project aims to implement the simulation of the combined teleoperation system in AMBF, which employs the model of the snake robot in Blender add-on and usage of different plugins. In addition to studying plugins for our implementation, we also aim to develop a controller using haptic device in AMBF and further test on the real robot.

## AMBF Simulation:

AMBF offers a real-time dynamic simulation of robots, free bodies, and multi-link puzzles coupled with real-time haptic interaction via several haptic devices [6]. Compared with Gazebo, AMBF has the advantage of being able to interact with and remove parts of a volumetric model which is of vital importance in medical manipulation like drilling. In the simulation work, the following portions will be included:

### 1. Blender add-on

The blender plugin is to ease the creation of multi-body config files that are used in AMBF. With the aid of the add-on, the user can model the snake robot from a given CAD drawing or using other modeling software. By running the AMBF blender add-on, the user can configure the joints and rigid bodies that will be used in the simulation.

### 2. AMBF+ plugin

A virtual surgical simulation framework AMBF+ is used for this project, enabling users to interact with a patient model for surgical practice. The simulation provides both visual and haptic feedback for assistance and realism [7].

The AMBF+ framework uses an AMBF Description Format (ADF) to describe different objects such as rigid and soft-bodies, sensors, actuators, and objects in the world scene. These ADF files can be split into smaller, modular files, allowing models with different components to be loaded together. The framework also allows for flexible parenting of different objects, so they don't need to exist in the same ADF file, making it useful for simulating surgical robots that often have interchangeable tools.

The ADF files are of three different types: (1) world files, (2) input devices files and (3) model files [7]. A meta-data file called the launch file contains the paths of a world file, an input devices file and several model files. Fig. 2 shows the flowchart of loading ADF files into AMBF+.

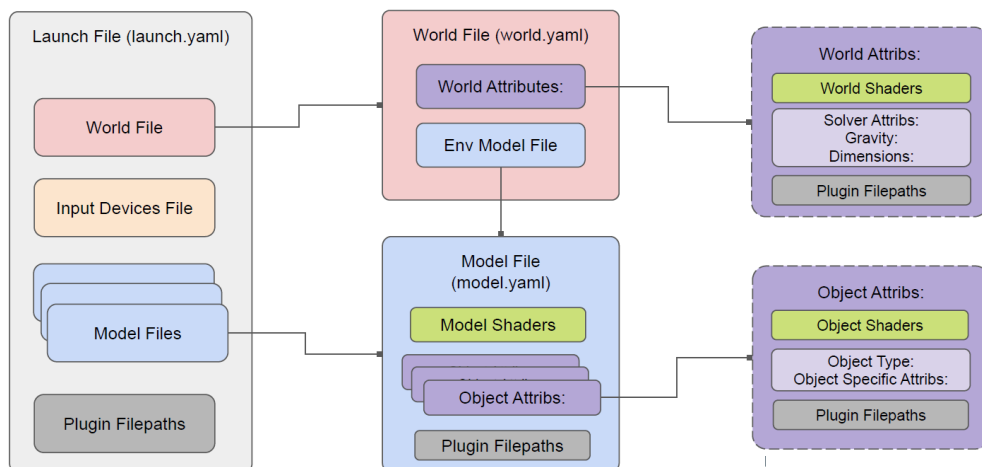


Figure 2. Flowchart of loading models and world scene objects into AMBF+ [7].

In AMBF+ simulation, the different computational scopes are in a sequence like afSimulator -> afWorld -> afModel -> afObject. To clarify, in the AMBF simulator, we first define a world by creating different models. Then, we can define each model using various objects such as bodies,

sensors, and joints, while also incorporating environmental elements like cameras and lighting. Therefore, plugins can be designed to expand the framework on different scales. For example, volumetric drilling plugin is a widely used one for volumetric drilling for skull base surgery with a handheld drill as shown in Fig. 2, with which we can actively modify anatomy with a virtual drill in the simulation. Although the application is not the same as this project, learning how to use this plugin can better help understand the functionality as well as the usage of the plugin.

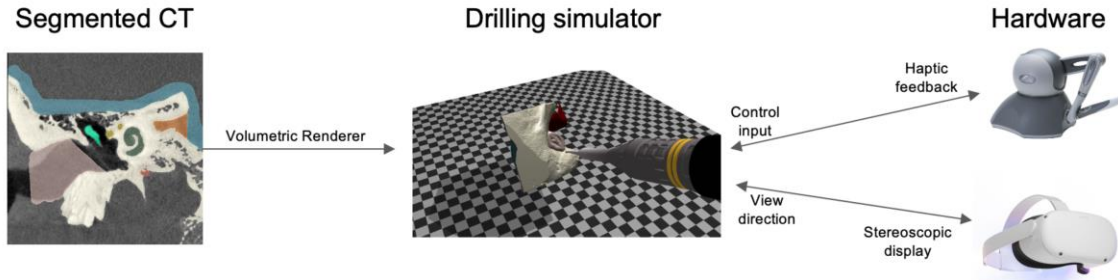


Figure 3. Overview of the virtual drilling simulator for skull-base surgery developed using AMBF [7].

There is another plugin built based on the volumetric drilling plugin to cater to a different application, i.e., a robot with a snake attached to it for spine/ortho surgery. Inspired by this plugin, we can build the combined system of eye robot and snake robot for vitreoretinal surgery. In this case, the functionality and performance of the snake robot can be tested in AMBF. Also, the simulation will be used to develop and test control strategies and allow for visualization of feasibility more rapidly.

## Teleoperation Control:

Fig. 4 provides an overview of the system. The AMBF framework and packages are used to facilitate communication between different components of the system. The control processing runs on python and C++. The Phantom Omni communicates with simulation by using ROS tf topic via a previously developed ROS package. Communication between the control loop and the AMBF simulation is done using the python client<sup>2</sup>.

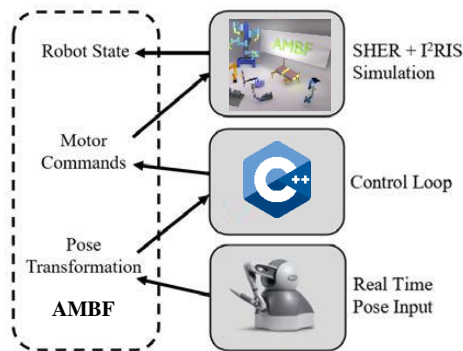


Figure 4. System overview: (L) AMBF; (R) Components

To develop control algorithms for snake robots, it is essential to possess a strong understanding of the robot's kinematics. Given that modeling the kinematics of a continuum robot is not as straightforward as a traditional manipulator, our goal is to develop an experimental forward kinematics and calibrate the

<sup>2</sup> The AMBF python client tutorial: <https://github.com/WPI-AIM/ambf/wiki/The-Python-Client>

motion of a new snake robot (I<sup>2</sup>RIS) to produce a mapping between the execution space of the snake robot and its configuration space, and finally to its task space. This will allow for accurate forward kinematic mapping, which is essential for modeling any form of inverse kinematics.

## Structure

The SHER comprises five actuated degrees of freedom, including three translational DoFs ('X', 'Y', 'Z') and two rotational DoFs ('Roll', 'Pitch'), as shown in Figure 5. Additionally, a mechanical Remote Center of Motion (RCM) mechanism is integrated into the robot's design to enable clinicians to perform surgical procedures in the vitreous space through a sclerotomy without requiring significant translational movement by the SHER.

We are using the latest iteration I<sup>2</sup>RIS design. Its snake-like end has a diameter of 0.9 mm with a designed  $\pm 45^\circ$  bending range [8]. Its 2-DoF actuation is made possible by four wires on a drive pulley perpendicular to the actuation direction.

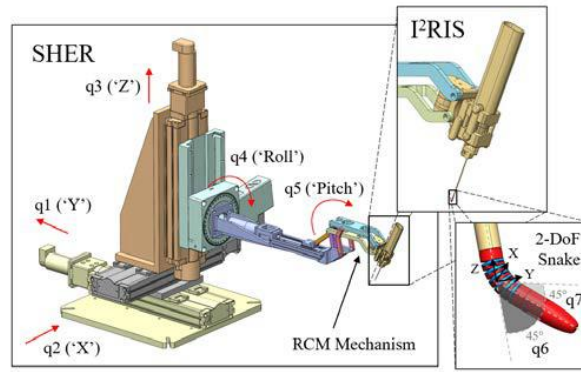


Figure 5. Joints of integrated SHSER and I<sup>2</sup>RIS system

After the user moves the Phantom Omni, the transformation from the device's base to its tip is published and extracted from the ROS topic. This transformation becomes the target input, which is then scaled to match the dimensions of the vitreous space. The device motion space has an approximate diameter of 130 mm, while the vitreous space has a diameter of approximately 21 mm. Scaling the input in this manner enhances the precision of the user's tool tip control by approximately a factor of 6.

## Forward kinematics

The forward kinematics equations of the system describe the position and orientation of the tool tip, given the joint states vector  $\vec{q}$ , as a homogeneous  $4 \times 4$  transformation  $F$ . From the transformation we can extract the rotation  $R$  and translation  $\vec{p}$ .

$$F = \begin{bmatrix} R(\vec{q}) & \vec{p}(\vec{q}) \\ \vec{0} & 1 \end{bmatrix} = f(\vec{q})$$

Where  $F$  is the transformation between the global reference frame and the tool tip. To determine this, we can multiply the forward kinematics functions of the SHER and I<sup>2</sup>RIS robots, denoted as  $F_{SHER}$  and  $F_{I^2RIS}$ , respectively. These functions involve both variable and rigid transformations.

$$F = F_{SHER}(q_{1-5}) \times F_{I^2RIS}(q_{6-7})$$

The SHER's kinematics are characterized by three linear transformations and two rotational transformations that depict the joint motion between each joint's positions. The kinematic chain is described in the same way as the joint chain within the robot simulation file.

I<sup>2</sup>RIS kinematic has two angles as parameters to determine the transformation of the tool tip to control the amount and direction of rotation between alternating sets of pairs of segments of the snake (Fig. 4, B). Each line of contact between the segments exhibits an alternating orthogonal direction of rotation. In Fig. 5,  $q_6$  refers to bending about the I<sup>2</sup>RIS Y-axis, while  $q_7$  pertains to bending about the I<sup>2</sup>RIS X-axis. To visualize the contact between two segments of the snake, we can think of two cylindrical surfaces that roll together without slipping, as demonstrated in (Fig. 6, C). Two virtual rotational joints are located at the center of virtual circles, which are concentric with the cylindrical surfaces and describe the rotation between two adjacent segments (Fig. 6, A). As the virtual circles for the two surfaces on each segment do not share the same center, there are effectively two virtual joints for every segment of the manipulator.

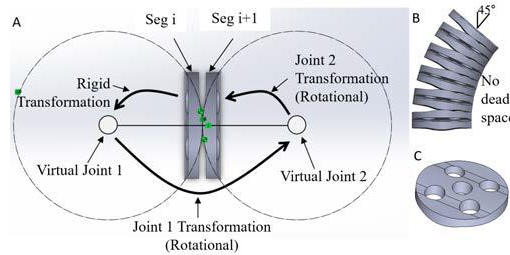


Figure 6. I<sup>2</sup>RIS kinematics: (A) Virtual joint representation; (B) Segment bend side-view; (C) Contact region.

## Control

### A. Constraints

Fig. 7 shows constraints that are applied on the target pose input by the clinician or command trajectory to produce the target pose [7]. If the input pose is outside the vitreous space, the target pose will be constrained to the nearest point on the inner surface of the eyeball's vitreous space. This constraint guarantees that the tool tip remains within the vitreous space, as shown on the left side of Figure 7.

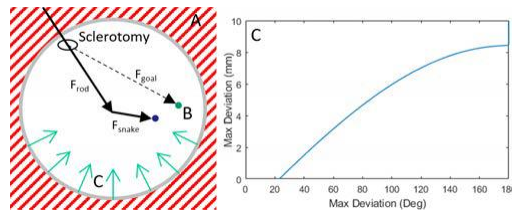


Figure 7. Constraints on input pose: (A) Position restriction; (B) Velocity limit; (C) Orientation restriction [7].

To ensure patient safety, the next pose goal is generated between each movement with respect to defined maximum speed limits for both positional and rotational velocities. These limits, namely max speed pos and max speed rot, respectively, restrict the speed at which the tool tip can be moved. This process can be seen in Figure 7, Left (B).

As shown in Figure 6, Left (C), the orientation is determined based on the distance from the surface, which constrains it towards the surface normal. When the tool is at the surface, the orientation is limited to within about 23 degrees from the normal direction, as shown in Figure 6, Right (C). By doing so, it guarantees that the bending body of the I<sup>2</sup>RIS will not collide with the surface while allowing the clinician enough flexibility to orient the tool to avoid obstructing the line of sight through the eye lens regardless of its position.

### B. Manipulator Jacobian

Instead of requiring a certain level of dexterity to generate smooth trajectory in solving inverse kinematics, we will control both eye robot and snake robot as one hybrid robot with a new Jacobian matrix



$$V_{st}^s = J_{Hybrid} \dot{q}$$

$$J_{hybrid} = [J_{SHER} \quad Ad_{g_{sB}} Ad_{g_{Bb}} J_{I2RIS}]$$

$$\dot{q} = \begin{bmatrix} \dot{q}_{SHER(5 \times 1)} \\ \dot{q}_{I2RIS(2 \times 1)} \end{bmatrix}$$

To perform autonomous position control or teleoperation with virtual fixtures using snake robot, it is necessary to have a good model for input-output mapping that will enhance operation accuracy and performance. Several experimental and analytical methods for kinematics modeling in snake robots have been developed, a probabilistic data-driven kinematic model with a hysteresis compensation algorithm will be used for this project, which is more suitable for nonlinear mapping between the input signals and the outputs (snake bending angles, yaw and pitch) [9].

By using the AMBF python client, an algorithm to control the robot simulation is designed as shown In Fig. 9:

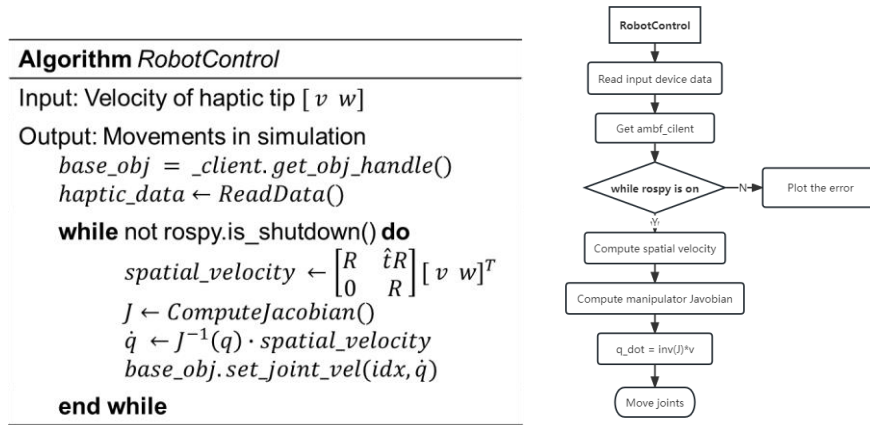


Figure 9. Flowchart of control loop

### C. Teleoperation Control

The AMBF framework and packages provide a powerful and flexible platform for facilitating communication between different components of a system. By enabling seamless communication between hardware and software components, AMBF helps to ensure that the system is well-integrated and functions effectively as a whole. The system overview is shown in Fig. 10.

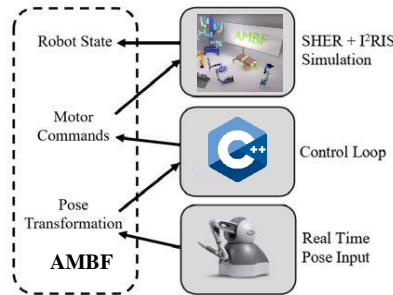


Figure 10. Control system overview

In the case of the Phantom Omni, communication with the simulation is achieved using ROS topics. This allows the robot to send and receive messages with the simulation, providing real-time feedback and enabling precise control.

Meanwhile, the control loop communicates with the AMBF simulation using a Python client that has been specifically designed for this purpose. This client allows the control loop to send commands and receive feedback from the simulation, ensuring that the system remains tightly integrated and responsive to user input.

Overall, effective communication is essential for the success of any system, and the use of AMBF and ROS topics, in conjunction with the Phantom Omni and Python client, help to ensure that communication between the various components of the system is reliable, efficient, and effective.

## Results

### AMBF simulation

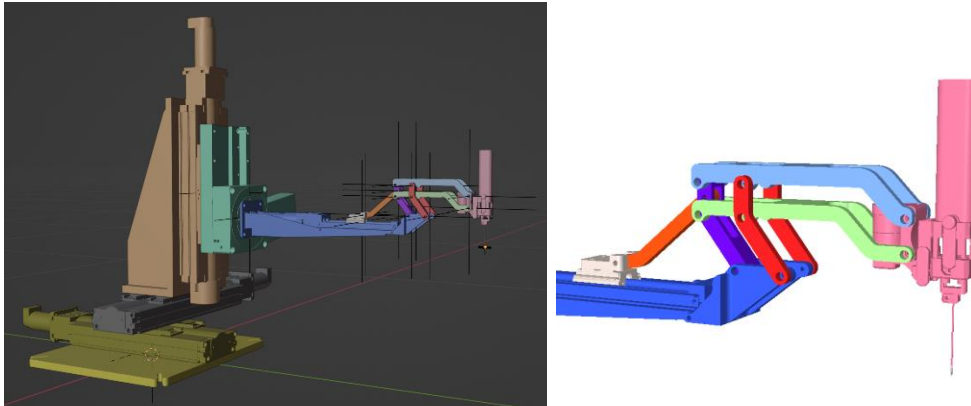


Figure 11. Simulation of the hybrid 7 DoF Snake & Eye Robot (I2RIS + SHER 2.1) system

The model simulation for the SHER robot is carried out using Blender 3.4.1, as shown in Figure 11 (left). In the simulation, the joints are defined and can be used to control the movement of the robot.

Although SHER has only five input joints for control, it actually has additional joints in the RCM structure, as shown in Figure 11 (right). This structure allows for movement around a fixed point, creating a pivoting effect, as long as the robot's base translational stages remain stationary. This is due to the parallelogram setup of the RCM structure, which provides additional degrees of freedom and enables precise control of the robot's movement.

Understanding the additional joints and degrees of freedom provided by the RCM structure is important for effectively controlling the SHER robot and achieving the desired outcomes. By taking advantage of the full range of movement provided by the robot's design, it is possible to optimize its performance and achieve greater precision and control in a variety of applications.

### Teleoperation control

As the working laptop does not support the connection with the Phantom Omni, the 3Dconnexion device will be used as an alternative. The 3Dconnexion device also offers six degrees of freedom, and

although it does not provide haptic feedback like the Phantom Omni, it can still be used to test the control algorithm for teleoperation and virtual reality applications.

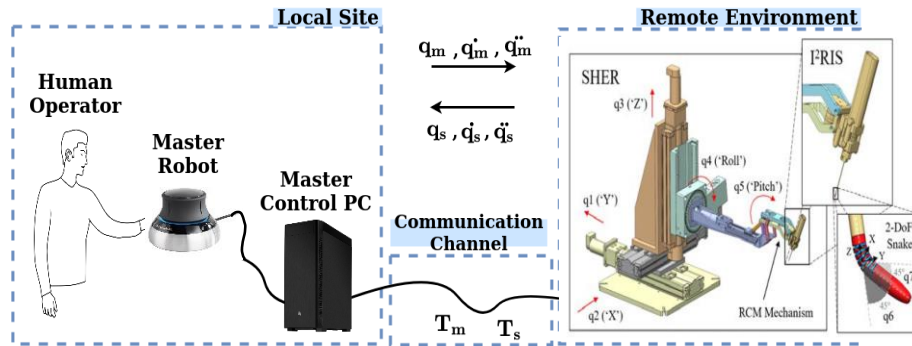


Figure 12. Teleoperation control framework, 6 DoF 3D Connexion as a master robot on the left side, and the hybrid 7 DoF Snake & Eye Robot (I2RIS + SHER 2.1) as remote robot on the right.

By testing the control algorithm on the 3Dconnexion device, we can verify that it is functional and can also run on the Phantom Omni once a connection is established. This allows us to ensure that the algorithm is optimized for both devices and can be used in a variety of scenarios that require precise control and haptic feedback. Overall, while the 3Dconnexion device may not offer haptic feedback like the Phantom Omni, it can still be an effective tool for testing and validating control algorithms and can be a suitable alternative for situations where the Phantom Omni is not available or cannot be used.

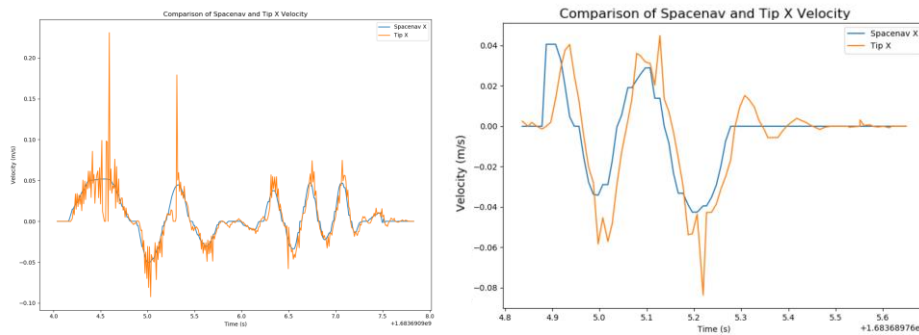


Figure 13. Real-time error of input and tip velocity

The control method used for this system is velocity control, which provides several advantages over inverse kinematics. By controlling the velocity of the robot's movement, rather than trying to calculate and control the precise positions of each joint, the computation required for control can be greatly reduced. This simplifies the control process and allows for more efficient and responsive control of the robot. Moreover, velocity control can improve the accuracy of the system by reducing the impact of inaccuracies or errors in the robot's position or orientation. By focusing on controlling the velocity of the robot, rather than the precise position of each joint, the control system can adjust for any errors or discrepancies in real-time, resulting in a more accurate and reliable control system.

However, only the position of rigid body can be obtained using AMBF python client rather than the velocity of the tip. Therefore, the differential equation of the change of coordinates versus time is used to calculate the velocity, resulting in a margin of error between the calculated and actual velocity.

While this error can affect the accuracy of the simulation, the overall margin of error is relatively small, at approximately 0.01m/s. It is considered acceptable for most applications and should not significantly impact the results of the simulation.

## Management Summary

### Task credit

As a one-member team, all the work on this project is being carried out by Xiao Wei. While this can be a challenging undertaking, it also provides several advantages, for example, making it possible to focus on individual strengths and interests, tailoring the project to her specific skills and abilities.

### Deliverables accomplishment

As discussed in the proposal, the deliverables are shown in the table below. Minimum and expected deliverables are finished. For expected part, 3D Connexion are used as a input device instead of Phantom Omni due to connection problems of working laptop.

The deliverables for this project, as outlined in the proposal, are listed in the table below. Both the minimum and expected deliverables have been completed successfully. However, I did encounter some technical issues during the project that required me to make an adjustment to the expected deliverables. Specifically, due to connection problems with the working laptop, the Phantom Omni cannot be used as the input device as originally planned. Instead, I have opted to use 3D Connexion as an alternative input device. I have thoroughly tested the device and have found it to be responsive and precise, allowing for accurate and intuitive control of the simulation.

Deliverables		Progress
<b>Minimum</b>	Developing a simulation model for the Eye Robot and the Eye Snake systems in AMBF	Learn AMBF + ADF + Blender addon
		Learn about how to use various plugin
		Get familiar with volume and 3D scans
<b>Expected</b>	Develop and implement a control algorithm on the AMBF model using a haptic device	Done (Using 3D Connexion instead)
<b>Maximum</b>	Implement the controller on the real robots	Not yet completed

Overall, the progress made on the project is good and the completed deliverables meet the expectations outlined in the proposal.

### What is learned?

I learned a lot from this project. This project has provided me with valuable experience in using Blender and AMBF simulation. By working on this project, I have gained a deeper understanding of the underlying principles and techniques involved in creating and simulating complex robotic systems.

Specifically, I have become proficient in using the AMBF framework and plugins, which have proven to be powerful tools for facilitating communication and integration between different components of the system. This knowledge will be invaluable in future projects, allowing me to develop more effective and efficient control systems and simulations.

Overall, this project has provided me with a solid foundation for continued learning and growth in the field of robotics. I look forward to applying the skills and knowledge gained through this project to future endeavors, and to exploring new possibilities and techniques as I continue to develop my expertise in this exciting and rapidly evolving field.

### **Future work**

I look forward to continuing to refine and improve the system in the future. In order to further validate the control algorithms developed in this project, I plan to continue testing using the Phantom Omni as planned from the beginning.

Additionally, I am excited to take the next step in this project by implementing the controller on real robots. This will require additional planning and coordination, but I am confident that the knowledge and experience gained from this project will be invaluable in ensuring successful implementation. By working with real robots, I will have the opportunity to see the impact of my work in a real-world setting and to identify any additional challenges or opportunities for improvement. This will provide valuable insights and help to further advance the field of robotics.

What's more, snake robot control is not the same as in the simulation. It is important to note that controlling a snake robot in reality is more complex than in a simulation. While we were only able to use two simple joints to define the robot in the simulation, the real-world control of a snake robot requires more sophisticated and nuanced control strategies.

Overall, I am eager to continue building on the progress made in this project and to further develop my skills and knowledge in the field of robotics. I am confident that the work done so far has provided a strong foundation for future success and innovation.

## **Conclusion**

A simulation of the integrated system that combines SHER and I2RIS has been successfully implemented in AMBF. By simulating the integrated system in AMBF, it is possible to model and test a wide range of scenarios and configurations, without the need for expensive or time-consuming physical testing. This allows for more rapid iteration and development, as well as greater flexibility and control over the testing environment.

In addition to the simulation model, a controller has been designed using the AMBF Python client. This controller provides an intuitive and efficient means of controlling the simulation, allowing us to interact with and manipulate the system using the input device, which features 6 degrees of freedom.

Through extensive testing and experimentation, we have successfully demonstrated the effectiveness of the controller in controlling the simulation model. The overall error is approximately 0.01m/s which is acceptable.

## References

- [1] Roizenblatt, M., Edwards, T. L., & Gehlbach, P. L. (2018). Robot-assisted vitreoretinal surgery: current perspectives. *Robotic Surgery: Research and Reviews*, 1-11.
- [2] Fleming, I., Balicki, M., Koo, J., Iordachita, I., Mitchell, B., Handa, J., ... & Taylor, R. (2008). Cooperative robot assistant for retinal microsurgery. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008: 11th International Conference, New York, NY, USA, September 6-10, 2008, Proceedings, Part II 11* (pp. 543-550). Springer Berlin Heidelberg.
- [3] “The peripheral aspects of vitreoretinal surgery,” *Centre for Sight*, 07-Mar-2020. [Online]. Available: <https://www.centreforsight.net/blog/the-peripheral-aspects-of-vitreoretinal-surgery/>. [Accessed: 01-Mar-2022].
- [4] “Eye anatomy,” *Glaucoma Research Foundation*. [Online]. Available: <https://www.glaucoma.org/glaucoma/anatomy-of-the-eye.php>. [Accessed: 01-Mar-2022].
- [5] Shi, K., Zhou, Y., Ebrahimi, A., Li, G., & Iordachita, I. (2022, April). Optimization-based Concurrent Control of a High Dexterity Robot for Vitreoretinal Surgery. In *2022 International Symposium on Medical Robotics (ISMR)* (pp. 1-7). IEEE.
- [6] Munawar, A., Wang, Y., Gondokaryono, R., & Fischer, G. S. (2019, November). A real-time dynamic simulator and an associated front-end representation format for simulating complex robots and environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1875-1882). IEEE.
- [7] Munawar, A., Li, Z., Kunjam, P., Nagururu, N., Ding, A. S., Kazanzides, P., ... & Unberath, M. (2022). Virtual reality for synergistic surgical training and data generation. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 10(4), 366-374.
- [8] Jinno, M., & Iordachita, I. (2020, November). Improved integrated robotic intraocular snake. In *2020 International Symposium on Medical Robotics (ISMR)* (pp. 1-7). IEEE.
- [9] (unpubl. MS)Mojtaba, E., Yanlin, Z..(2023). A Data-Driven Model with Hysteresis Compensation for I<sup>2</sup>RIS Robot. In *2023 International Conference on Robotics and Automation (ICRA)*. IEEE.