# CIS II Final Report

## ACTUATION DESIGN OF SNARE LOOP ROBOT

WENPENG WANG

# Contents

## Introduction

The goal of this project is to design and realize an actuation system of a snare loop robot. A snare loop will normally perform a job of grabbing and placing in the human body, but, for the case of this project, the goal is different. The snare loop robot will provide a curvature that fits the shape of the eye. With the curvature provided, function of delivering medicine to the backside of the retina can be added to the finished actuation system.

The project team is responsible for developing the physical model of the actuation system and designing the controlling algorithm (including GUI and the method of commanding the system). A detailed explanation of the technical approach will be stated in the later section in this report.

This project will potentially create benefits to the patient for performing the cell therapy for restoring vision. This method of treatment will be expanded in the next section of Problem Statement.

## Background

Cell therapy is envisioned as a future treatment for degenerative retinal diseases. The word “cell” indicated that this method will use cell-based method in treating the degenerated retina. There are two main strategies for using the cell therapy. The first one is paracrine therapy. Paracrine therapy is focusing on the concept that transplanted cells will release diffusible factors that will eventually promote the viability or function of existing photoreceptor cells. For this approach to be therapeutically efficacious, treatment must be delivered before the onset of widespread retinal atrophy.

The second method is called cell-replacement approach. This approach will replace either RPE or photoreceptors. RPE-replacement therapy aims to replace degenerated native RPE cells with the cells that derived from stem cells directly into the subretinal space at or around the lesion site (further information please refer with Background Reading “Bioengineering strategies for restoring vision” on the course wiki).

## Problem Statement

There does not exist a matured method in performing the cell therapy that can easily access to the backside of retina. The project team is to provide a control foundation that can be used for performing cell therapy. The snare loop must control the degree of curve into four different directions: up, down, left, and right. With a good control logic, the snare loop can be controlled accordingly by user inputs to navigate through between choroid and retina. The overall pathway will be shown in the next figure.

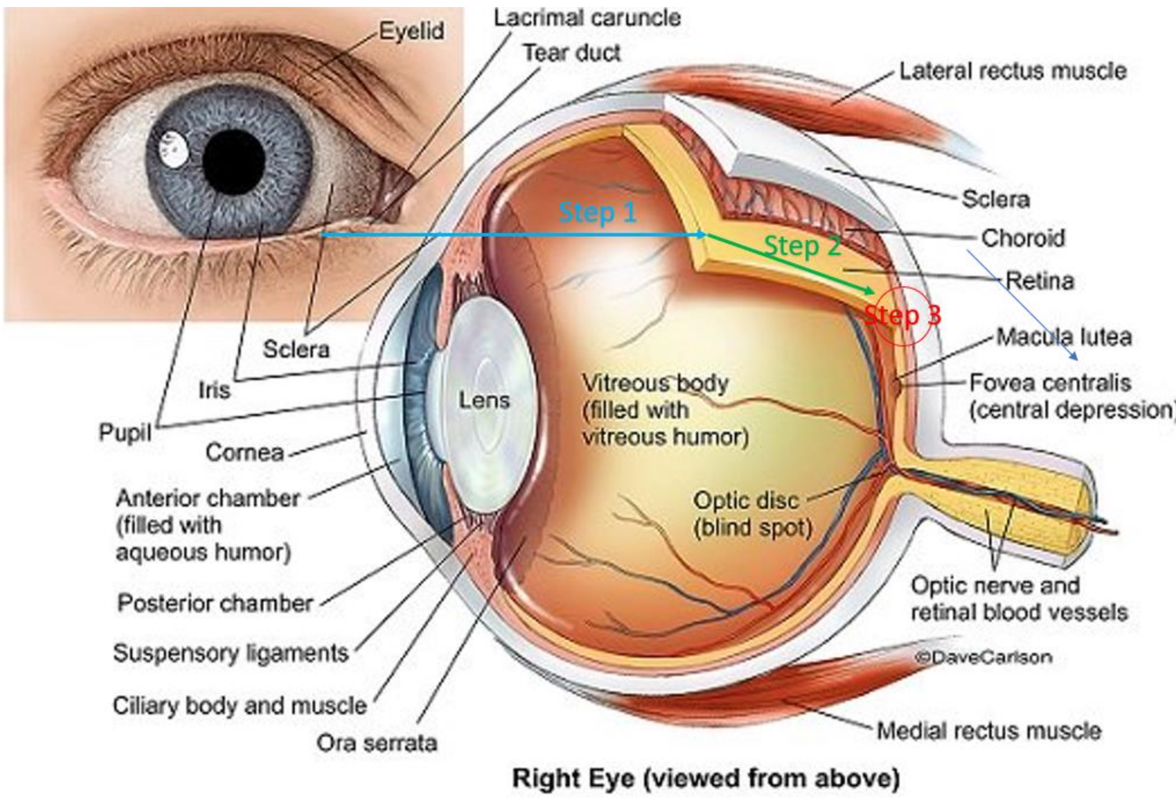


**Figure 1.** Pathway to the backside of retina.

This project is mainly a proof of concept of snare loop as an interface in eye surgery. The project team is focusing on designing a physical casing for the actuation system and developing the control package that includes multiple control algorithms (keyboard and GUI).

## Deliverables and Finished Works

In this section, the team will state the stages of deliverables and their corresponding status.

1. Minimum Deliverable:

a. Description: proof of concept that the snare loop can go up and down in a simple model.
b. Status: finished.

2. Expected Deliverable:
   a. Description: add linear actuators to provide motion to left and right, implement them into a new housing that can contain all the moving parts together.
   b. Status: finished by using stepper motors instead of linear actuators.
3. Maximum Deliverable:
   a. Iterate design to get a handheld tool.
   b. Status: can not be finished due to lack of linear actuators (the size of the stepper motor is considerable big). The team is now focusing on design a housing of the wires, which will clean up the workspace.

# System Architecture & Approach

## Hardware

The following is simple schematic of the demanded motions of the system.

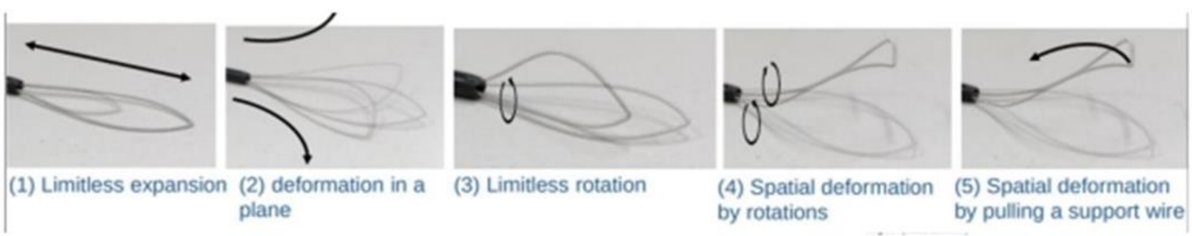


**Figure 2.** General motion of snare loop.

As one can notice from the schematic above, the stepper motors will provide the degree of freedom of commanding the loop to left and right. The two maxon motors are used for providing the degree of freedom to move the loop up and down. The command of stepper motors will be provided by Arduino. Serial communications will be used to provide the input to the Arduino. A High and low signal through the serial will be used to specific the two states of moving left and moving right. Each maxon motor is controlled by an EPOS motion controller. The system has been putted in a housing as demanded by the expected deliverable, and the following figure is the current actuation system to date.

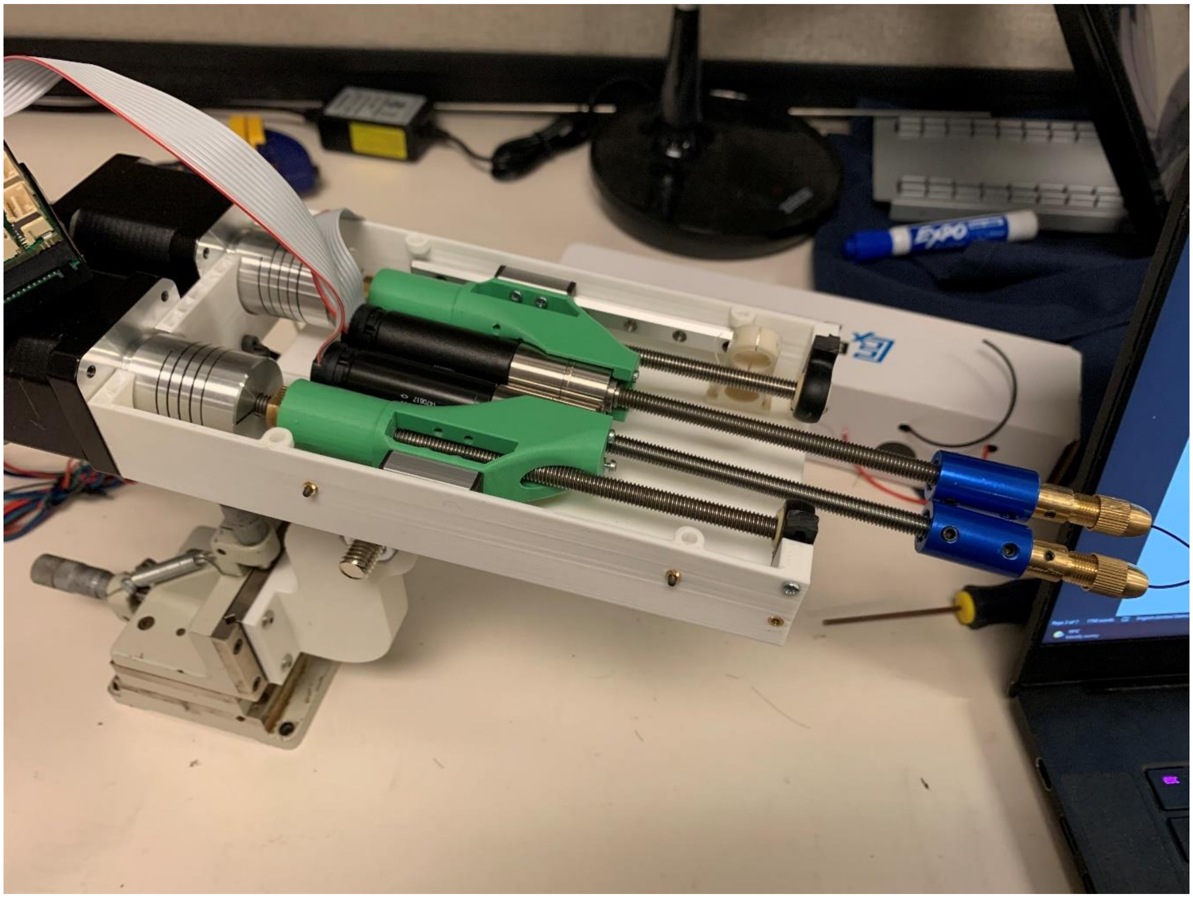

**Figure 3.** Actuation system setup.

## Software

In this section, the result of running the control package. The team compose the following control method.

The first control algorithm the team compose is a keyboard control. This function is using WASD keys to control the actuation system. The W and S keys will handle the loop motion of up and down, while the A and D will control the stepper motor to move the loop left and right. The running python code will read the PC keystrokes and register the strokes with the specified keys stated above. The program will use a pop-up window to obtain the user demanded velocity value for the maxon motors. The documentation of how to use the package is already in the repository, and one can also refer this document on the course wiki. The python prompt will also provide some information on using the code.

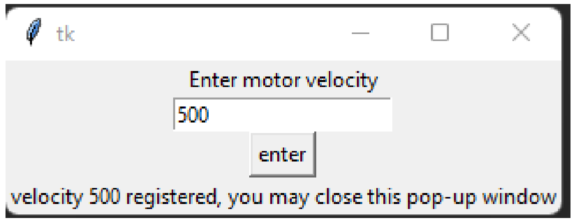


**Figure 4.** User input velocity window for keyboard control.

```
W: 10, A: 0, S: 2, D: 0, press (R) for setting velocity, (esc) to terminate the program
ss is pressed
W: 10, A: 0, S: 3, D: 0, press (R) for setting velocity, (esc) to terminate the program
ss is pressed
W: 10, A: 0, S: 4, D: 0, press (R) for setting velocity, (esc) to terminate the program
ss is pressed
W: 10, A: 0, S: 5, D: 0, press (R) for setting velocity, (esc) to terminate the program
ss is pressed
W: 10, A: 0, S: 6, D: 0, press (R) for setting velocity, (esc) to terminate the program
ss is pressed
W: 10, A: 0, S: 7, D: 0, press (R) for setting velocity, (esc) to terminate the program
ss is pressed
W: 10, A: 0, S: 8, D: 0, press (R) for setting velocity, (esc) to terminate the program
```

**Figure 5.** Python callback result of keyboard control.

The second algorithm is controlled by a GUI. The GUI will provide a place for commanding speed, direction, and position all together. Similar to the keyboard control, there will be specific key for commanding the motors to move the loop to different direction. The main improvement introduced is the position control. With the position control, one can move the motor to any specific position and also to command the loop to move back to the home position, which makes the GUI a better way to test with.

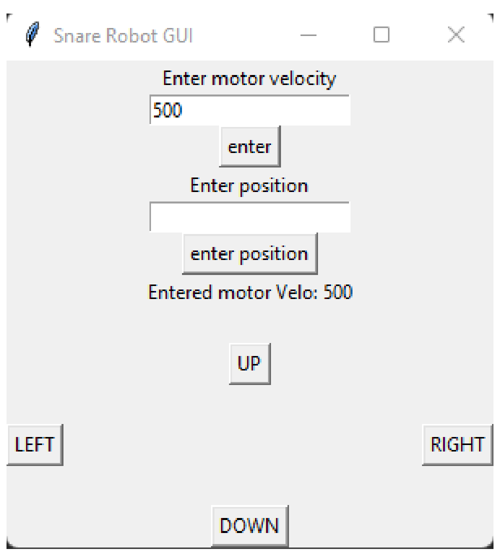


**Figure 6.** GUI of GUI control.

### *Documentation*

The documentation is provided in the repository and on the course wiki page. The user should check their python packages, since the package is using not a common build (include some libraries that are not available for most IDE). The other thing one needs to make sure is the port number that the maxon motor connects to. The step for obtaining the COM port number has been detailed stated in the document, please reference the How To Use document.

Readme file provides the basic information of each program inside the repositor. The readme file provides user with starting information of the function of each program. One might notice that there is a EposCmd64.dll. This is the file for commanding the EPOS motion controller with python. If this dll file is not found in this repository, user can also obtain it in the Maxon Drivers folder included.

## Test of Performance & Results

In this section, the team will provide some figures using the control programs stated in the above section. The better viewing media is video, but due to the nature of report the basic output will be indicated here. For viewing the video, the video will be later uploaded to course wiki page or YouTube (the place will be update later).

There are 2 stepper motors that attach outside the white housing of the maxon motors. The stepper motor will move the maxon motors forward and backward, which will make the loop goes either to the left and to the right.

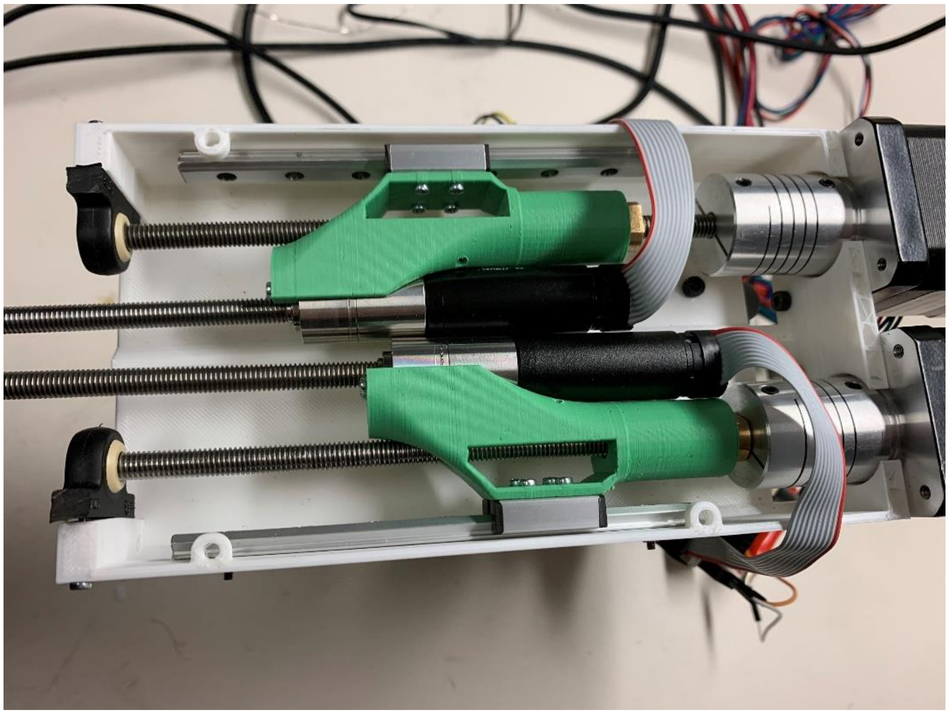

**Figure 7a.** Stepper motor provided motion 1.

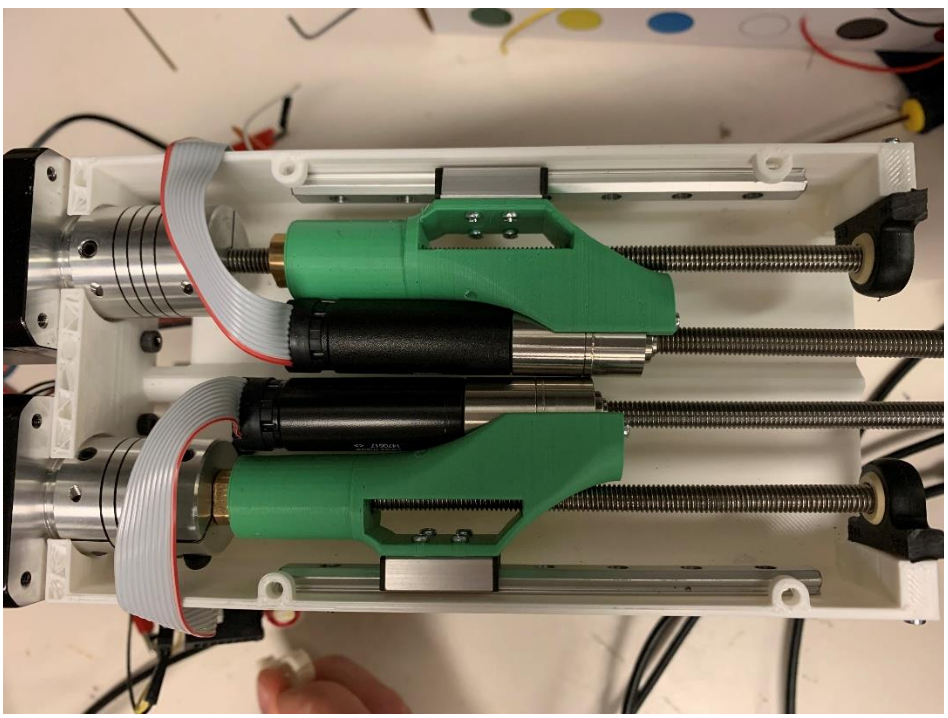

**Figure 7b.** Stepper motor provided motion 2.

One might notice that inside those figures above, it is hard to tell if the loop is moving to left and right direction. The first reason could be that the mount of difference provided from the stepper motors is not enough to bend the loop large enough. Since we apply the maxon motor at the end of railway, there is not much space to move the maxon motor. The second reason could be the radius of the loop. The team is using a pretty big diameter for the proof of concept. The bigger the diameter, the stiffer the loop will be, which it sustains a bigger amount of change. If the diameter of loop could be smaller, a bending to left and right should be easier to obtain. The recommendation will be included in the Future Recommendation below.

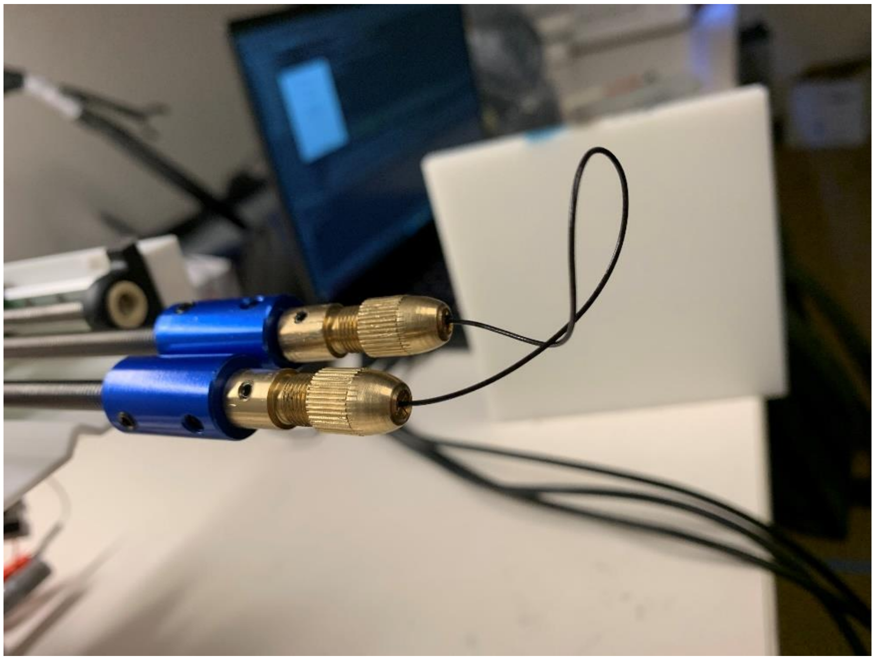

**Figure 8a.** Loop motion up with maxon motors.

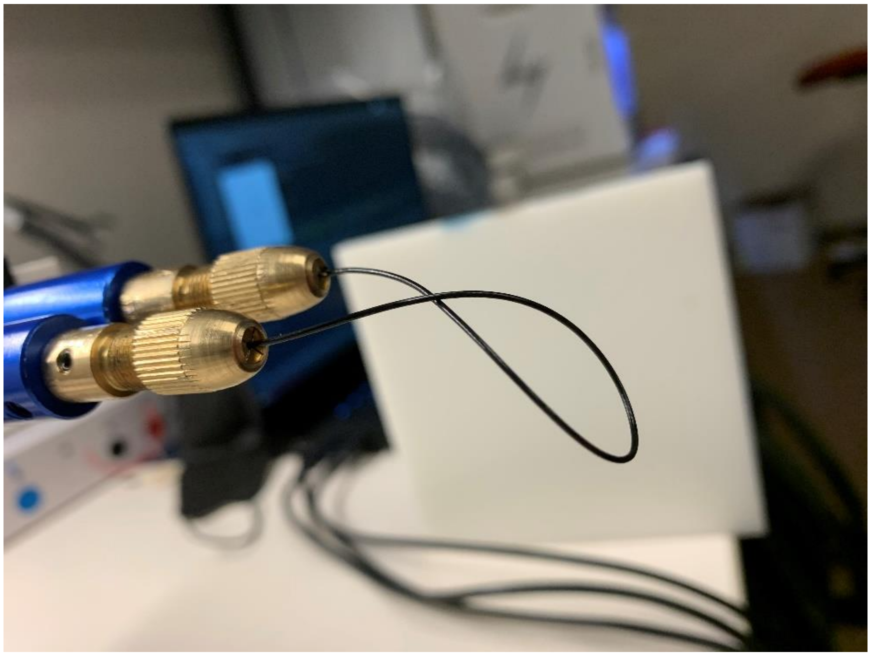

**Figure 8b.** Loop motion down with maxon motors.

The maxon motors can create a much bigger torque compared to the stepper motors, which the degree of bending up and down are noticeable. This also means that it is safe to use a smaller diameter for the future iteration of design.

## Significance

The team successfully obtained the motioned demanded in the problem statement. These motions are the initial foundation of a new type of robot. With the introduction of the position control, the team is now able to command the position with the desire angle input. For example, the user could command to move the maxon motor by 90 degrees, and then the maxon motors will move to that angle based on the encoder reading obtained. With this position-angle control, it provides the future ability of checking the result with the FEA simulation. Currently, the mentor is working on the developing the FEA, while the team could update the existing code to include this feature in the next generation of the control package. The team will also update the documentation with the steps corelated. Hopefully, this interface of controlling the system could provide the future participants with all the information needed and easy enough for them to update and improve the code.

## Management Summary

### *Work Distribution*

This CIS II project is a single person project. I personally took all the responsibility for all the tasks involves in this project. I would like to thank Andrew for his help on developing the CAD files and printing out those developed files. I made all my contribution on developing the control package and obtaining all the features mentioned above.

### *Future Recommendations*

As one can notice from above, the system contains 2 EPOS motion controller with 2 maxon motors and 2 stepper motors that controlled with Arduino. The first recommendation is on the controller used. The one should be advance is the Arduino. There are several reasons that it needs to be further advance during future. The first reason is that Arduino is not a real-time system. A real-time system could be crucial to maintain the loop position. In other words, a real-time system could provide the user a sense on where the loop is while the linear actuator (stepper motor in the current state) is moving and provide user ability to stop the loop motion needed.

The second reason that the Arduino should be changed is that the stepper motors that controlled by Arduino has a size that not easy to make the design smaller. The dimension of the stepper motor limit the team ability to shrink the design in the horizontal plane.

The next recommendation is to use a close fit guide rail. One can reference with the figures above, one can easily notice that the holders of the loop are not level in the horizontal plane. The reason that this is happening due to the wrong fitting on one of the guide rails used.

The next step of this project is to further advance the system. The first direction is to get the physical housing smaller. The second direction is to bring more feature to the current control package. The team think that introducing a live graphing utility can be a good next step to implement to show the decoder position more directly. The interface of reading the encoder position is already in the control package, which can be used whenever its needed.

## *Conclusion and Lesion Learned*

The motion demanded and methods of control are successfully obtained during this semester's CIS II project. This project will be served as a foundation to a new surgery robot that can be used to perform cell-replacement therapy that deliver either stem cells or RPE to the backside of the retina.

The first lesion I learned is that the during a project, it is good to have a detailed weekly plan. There are lots of tools online that can be used to develop a plan during the week. Also, this type of weekly plan created can be shared with other people (mostly team members). With this information shared, the team could better coordinate together.

The second lesion is on the documentation. There should be documentation on almost everything with this project. The documentation will provide the future groups with all the information they need for understanding the technical approach and the developed codes. Once they understand the materials better, they can modify and improve the existing package better.

## Reference

Cehajic-Kapetanovic J, Singh MS, Zrenner E, MacLaren RE. Bioengineering strategies for restoring vision. Nature biomedical engineering. January 2022. doi:10.1038/s41551-021-00836-4