Final Report

Eye Tracker Tester

Robotic Arm

Presented to Jon Campbell, Microsoft Research

Mechanical Systems Design

Washington State University

Fall, 2021

The Robotic Arm Team:

Abdulrahman Aljuhani, Adrian Szymanski, Bandar Bahattab, Chad Jerald, Jiwon Kim, Roderick Kimball, Trevon Anderson

Executive Summary

For the past few semesters, Microsoft has engaged with students from Washington State University's Senior Mechanical Engineering Design class. The purpose of this collaboration has been to create a tool that tests eye tracking devices. These devices are relatively new and therefore, there are no standards or methods by which to test or report their performance.

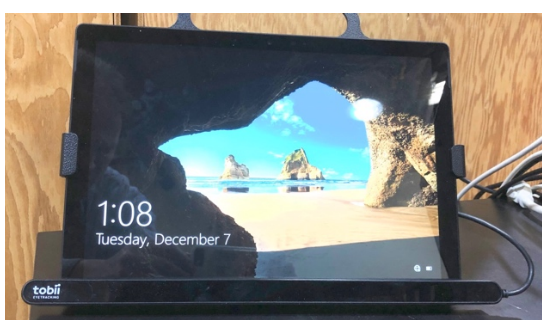
This semester, this project has been split into two parts. Our team was tasked with controlling a robot arm, the end of which is to hold a robot head. Another team was assigned the job of creating a face and eyes. We achieved the objective of controlling the robot arm to some degree of satisfaction. We outsourced the task of creating the head to the other team since they were doing that anyway.

Originally, we believed we might make a complete, functioning tester. We were not able to complete this task. However we created hardware that could be part of a complete tester. We designed and built a table for mounting the robot, a tablet stand to hold the tablet and eye tracker accurately in place in front of the robot, and a squaring device for mating the positions of the robot and the tablet stand. We designed but did not build and laser holder/pointer and mounting hardware for the head and eyes. We made but did not design a smaller 3-D printed laser holder.

We created a calibration procedure for the laser holder to allow the exact direction orthogonal to the end of the robot arm to be known. By performing this procedure, an accuracy of +/- .05 degrees was achieved. This satisfies the specified criteria of +/- 0.1 degrees.

A test of sustained accuracy was run in which the robot was run through a square pattern involving all six motors for 4000 repetitions. A laser was attached to the end of the robot arm. The positions of the laser dot on a surface about 6 meters away were recorded before and after the test. The creep in position during this test was found to be at most 0.013 degrees.

In this document, we discuss each of the components of the test apparatus and recommendations we believe will improve their performance. In the appendix to this report, we list all of the deliverables that accompany this report, including tools, code, CAD, and built components. Also included are the calculations used in determining robot positions.



Microsoft Surface Tablet with Tobii Eye Tracker Attached

Introduction

We are the Robotic Arm Team working on the Eye Tracking Device Tester for Microsoft Research. Team members are Abdulrahman Aljuhani, Bandar Bahattab, Chad Jerald, Jiwon Kim, Roderick Kimball, and Trevon Anderson. We are all senior mechanical engineering students at Washington State University. Working with us is our mentor Gene Jones, formerly of LRP Creative Solutions and Hewlett Packard.

An eye-tracking device allows a user to select items on a screen by looking at them. This can allow a person of very limited mobility to use a computer. There are many eye-tracking devices on the market. Naturally, their manufacturers all say that their product works wonderfully. However, the industry has no standard by which to rate or compare devices. Thus, even the most sincere testimonial by someone regarding their own product has very little meaning. Microsoft Corporation would like to establish standard metrics for comparing eye-tracking devices. To this end, Microsoft would also like to develop standard testing equipment for the purpose of assessing the functionality of any eye-tracking device.

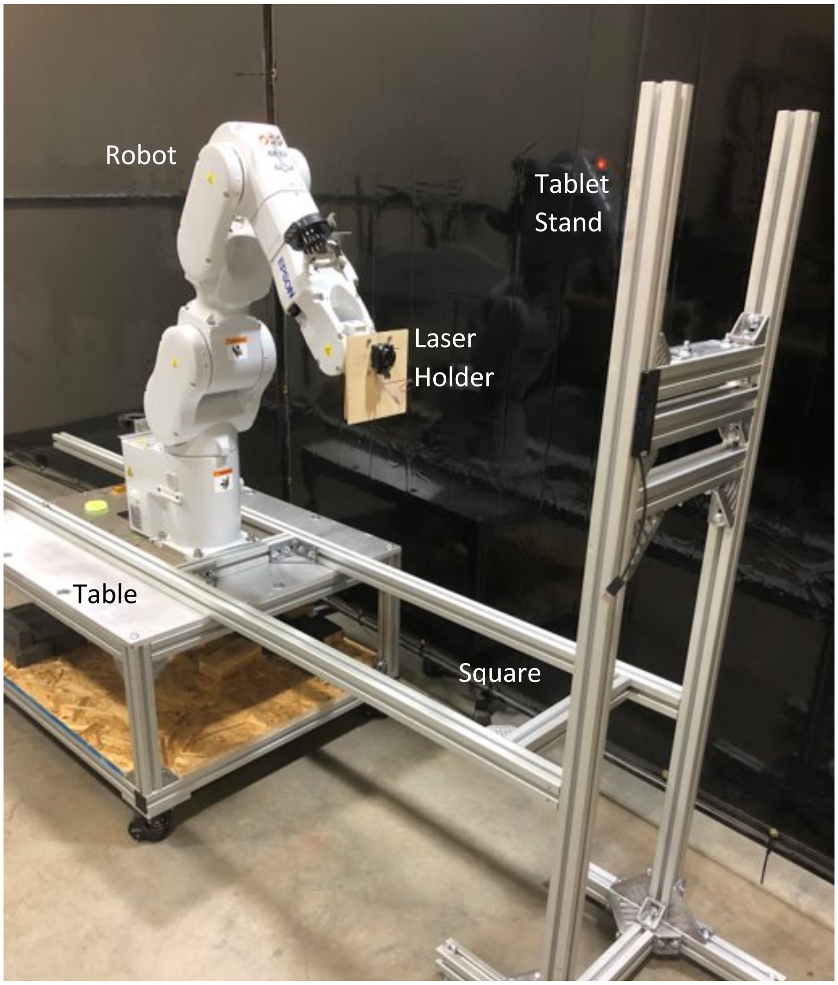
The device to be designed uses a human-like face mounted on a robotic arm. The face will include realistic-looking eyes that can independently rotate on two axes. The robotic arm will allow the face to be positioned in any realistic way, simulating the possible head positions of a real user. In an eye tracker test, virtual objects, including a keyboard will appear on a screen. The eye-tracking device to be tested will be installed, and the Eye Tracking Device Tester will assess how well it can “use” the computer by executing a programmed series of activities, including typing on the virtual keyboard.

Two teams from our ME 416 – Mechanical Systems Design class were dedicated to this project. Our team was tasked with making the robotic arm function and move to a programmable sequence of positions and orientations using three axes of motion and three axes of rotation. The other team’s task was to create the realistic face and control the motion of the eyes. Their results can be found on the eye tracker eye redesign team basecamp or the respective Github upon Jon Campbell’s approval.

What We Made

The proposed eye tracker tester has a human-like face and eyes. It is programmed to direct its gaze to locations on a screen to accomplish tasks on a computer. Information is collected as to how well it was able to complete the tasks. The apparatus must be precisely designed and constructed. This precision allows any difficulty in operating the computer to be attributed to the eye tracker itself. Thus, the performance of the eye tracker can be assessed.

We did not make a complete eye tracker tester. What we did was to design, build, and test an apparatus we believe can be used as part of a complete system. The components we made are the Table, Square, Tablet Stand, Head Mount, and Laser Holder shown below.



The Apparatus We Created

Robot Table

Goals:

- Ensure stability to prevent tipping.
- Allow the robot to be moved from place to place.
- Minimize vibration creep in the system.
- Allow for an all-in-one package with the robot and controller
- Allow for cable management
- Last indefinitely (at least 10 years with occasional tightening.)

We were able to design a table that achieved all of these goals. Upon recommendation from the Cougar Shop, we decided to use Bosch aluminum 45mm square T-rail tubing to make frame. This tubing is exceptionally sturdy and provides nice square angles when put together as shown.

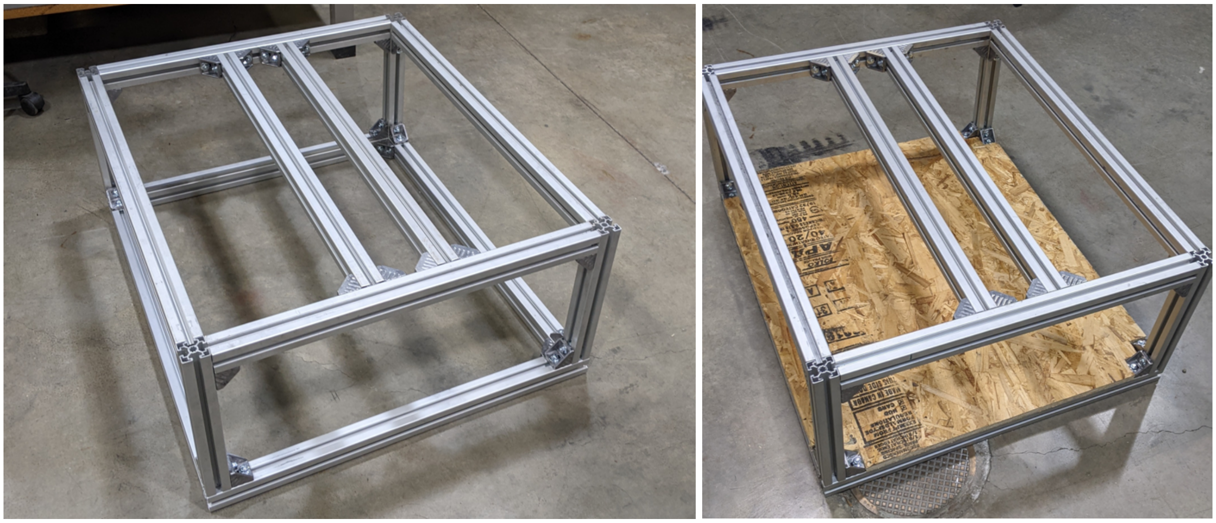


Table Frame & Bottom Shelf

The cross beams across the center top of the table support the weight of the robot while allowing cables to pass between them and through a rounded square hole in the top plate of the table, keeping them out of the workspace. Once this was put together, a plywood shelf was added to the bottom to hold the controller box and the weights we determined were needed for stability. With the addition of the top plate, casters, and weights the table was now complete:

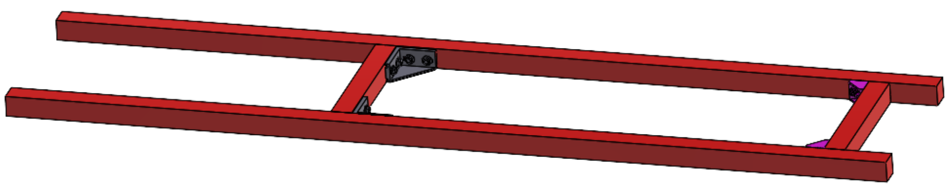
Completed table including top and weights

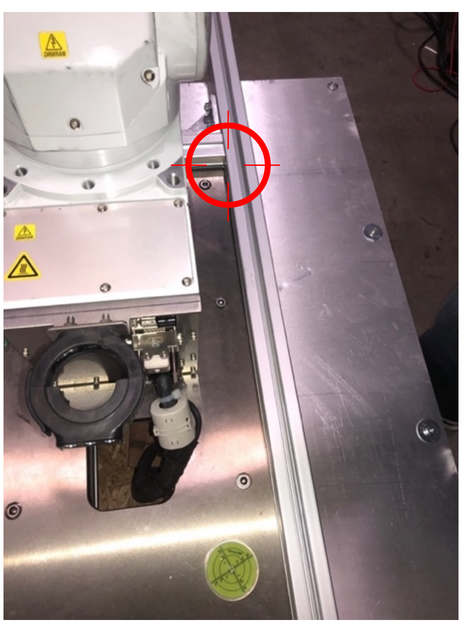
Casters with retractable feet for switching between mobility and immobility

Once the table was complete, it was ready for the robot. With several people lifting and ensuring the table casters were stopped, the robot was lifted into place and bolted down to the table.

Square

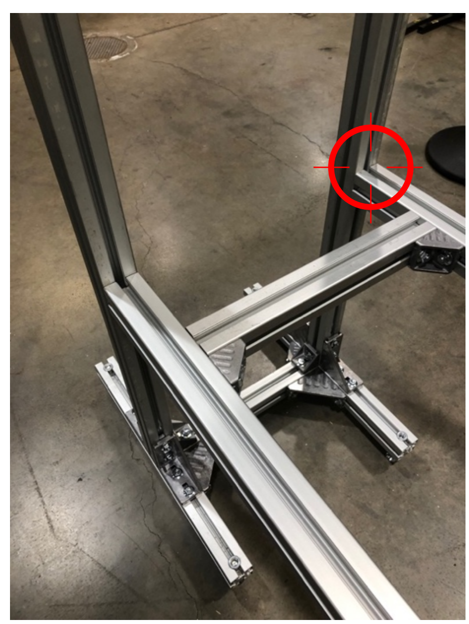
The square is a tool that aids in placing the Tablet Stand in a precise position relative to the robot.





The Square is placed at the corner of the robot mounting plate as shown

The tablet stand is placed at the end of the square at the spot indicated.



Tablet Stand

Goals:

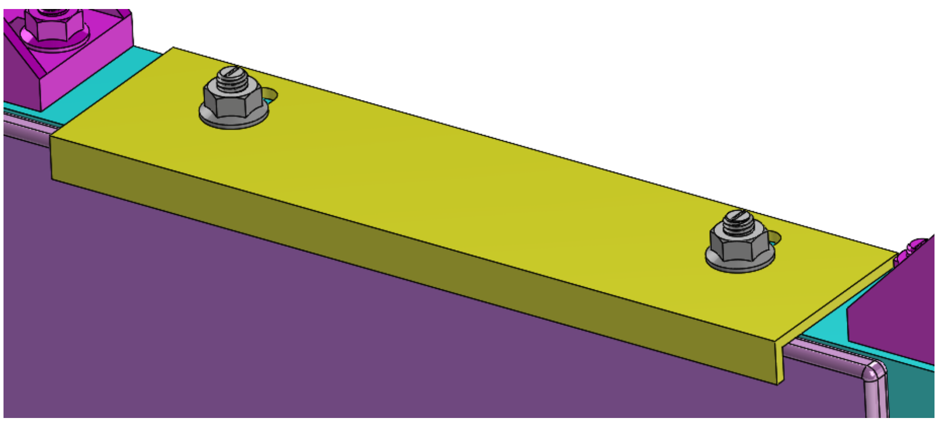
- Ability to be leveled and vertical on unlevel floors
- Precision tablet holding
- Ability to adjust for use with various tablets at various heights
- Ease of use

The team went through several ideas on how this could be accomplished from attaching the tablet to a strut on the table to having the tablet on a free-moving tripod and measuring each time a test would take place. After deliberation and iteration, we decided tablet stand would be made of the same Bosch tubing that the table was made of. This decision was made to preserve squaring accuracy as it was determined to be the most difficult to prove and the most important to maintain.

The tablet stand can be leveled on two axes using a carpenter's level and by adjusting the feet individually.

Modular hardware makes the tablet stand adjustable for tablets of all heights, widths, and thicknesses.

The first version of the tablet stand included small bar clamps to hold the tablet in place. These worked, but the brackets in the latest design are much easier to use and are less likely to damage the tablet. The brackets we made are still not the final design. The final design is shown here. It has closed slots for the bolts, so the brackets will not fall off, even if they are loosened. We recommend the final design over the ones we made.



Latest version of the Tablet Bracket (padding not shown)

Head Mount

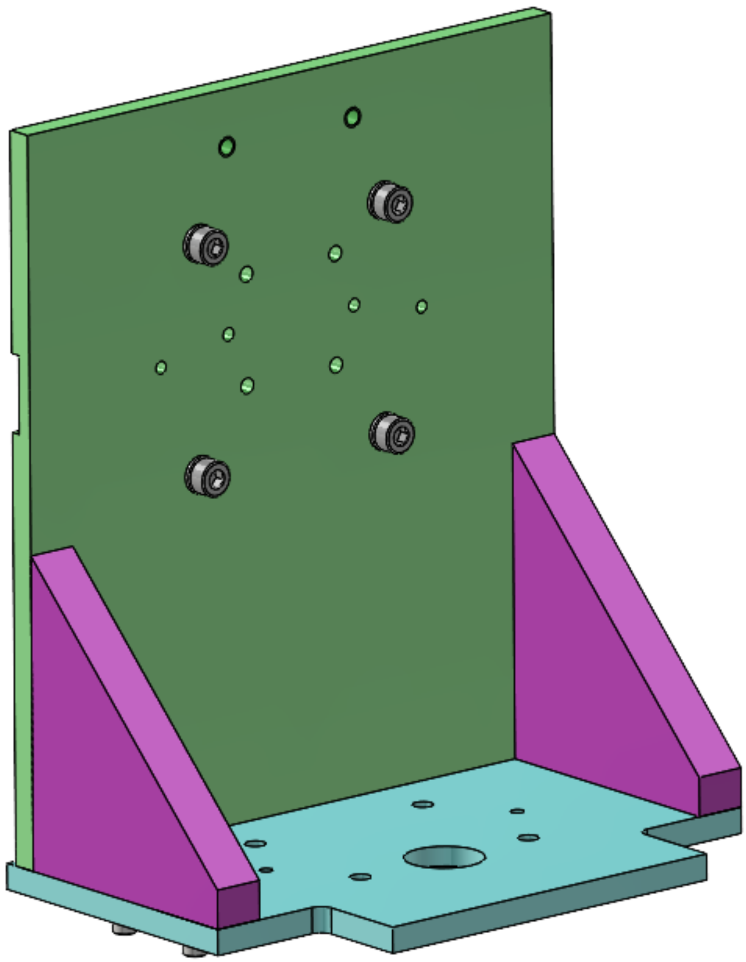
Goals:

- Compatibility with eye and face assembly and with laser holder
- Precision construction
- Rigidity
- Low weight

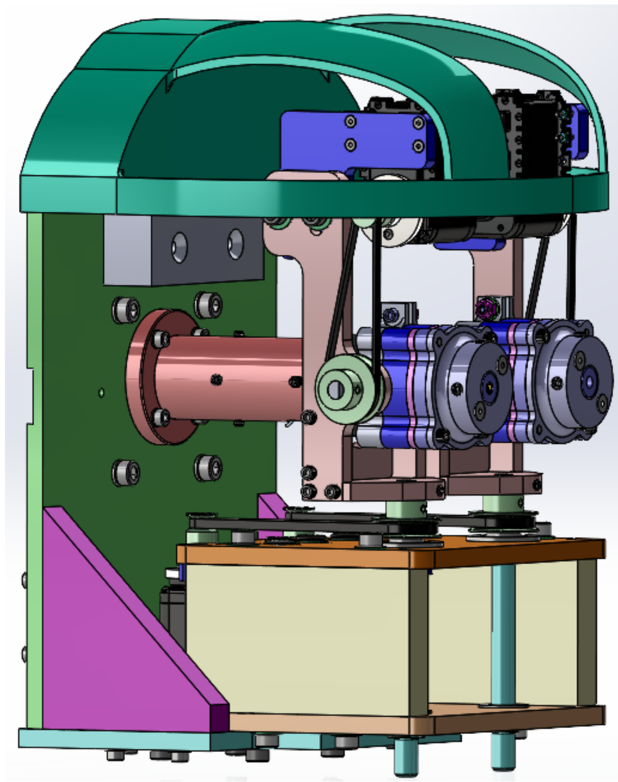
Th Head Mount is made of 6061-T6 Aluminum and stainless-steel fasteners. Dowel pins are inserted where the baseplate and bottom plate meet to ensure that all parts are aligned to

±0.01 of an inch. Other materials such as ABS plastic and steel were considered for this design, but ABS plastic does not provide the precision needed and steel is unnecessarily heavy for this end effector. The robot has an 8kg weight limit for end effectors, so it is important to consider weight reduction.

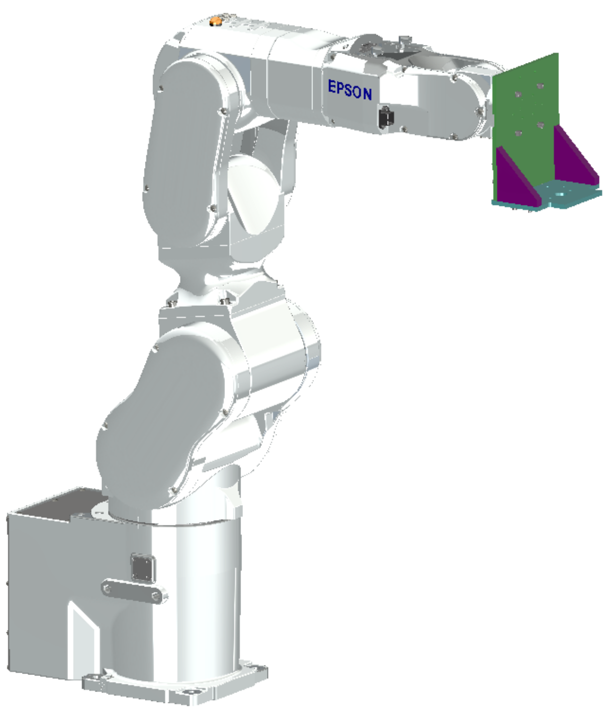
Due to manufacturing time constraints, we were not able to build this head mount. In order to perform our calibrations, we made a temporary base plate of 1/8” birch plywood.



The Head Mount We Designed



The Head Mount with the Eye/Head team's face & eye assembly and the Laser Holder

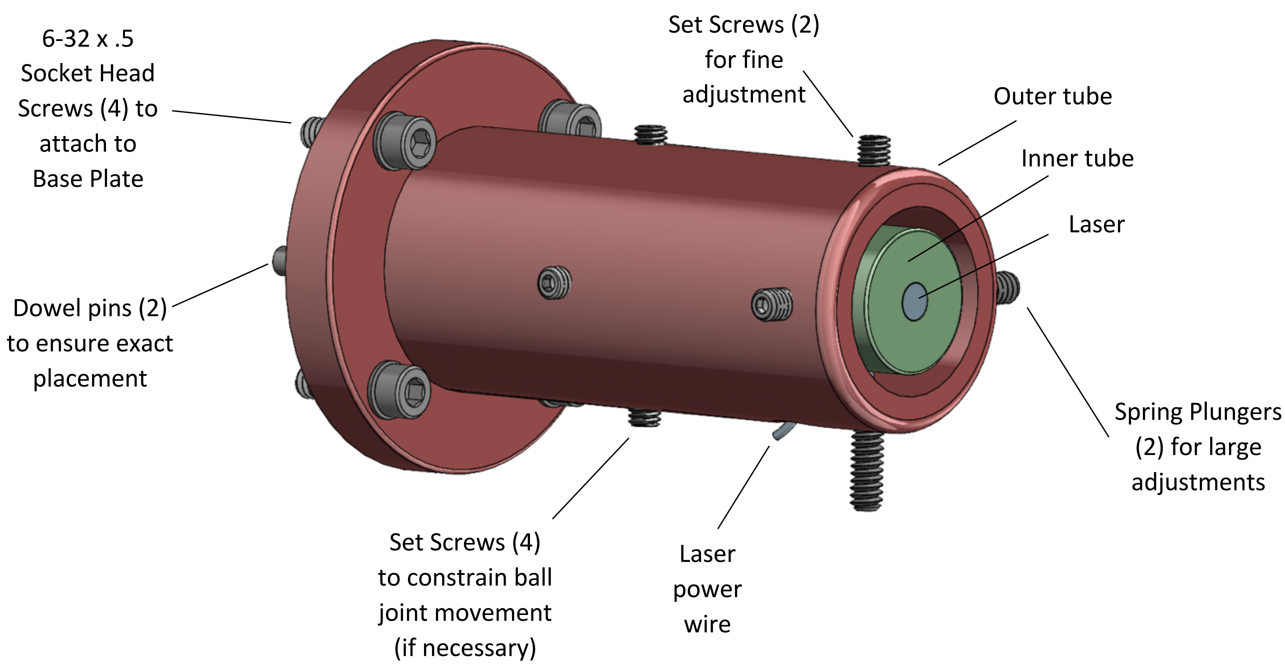


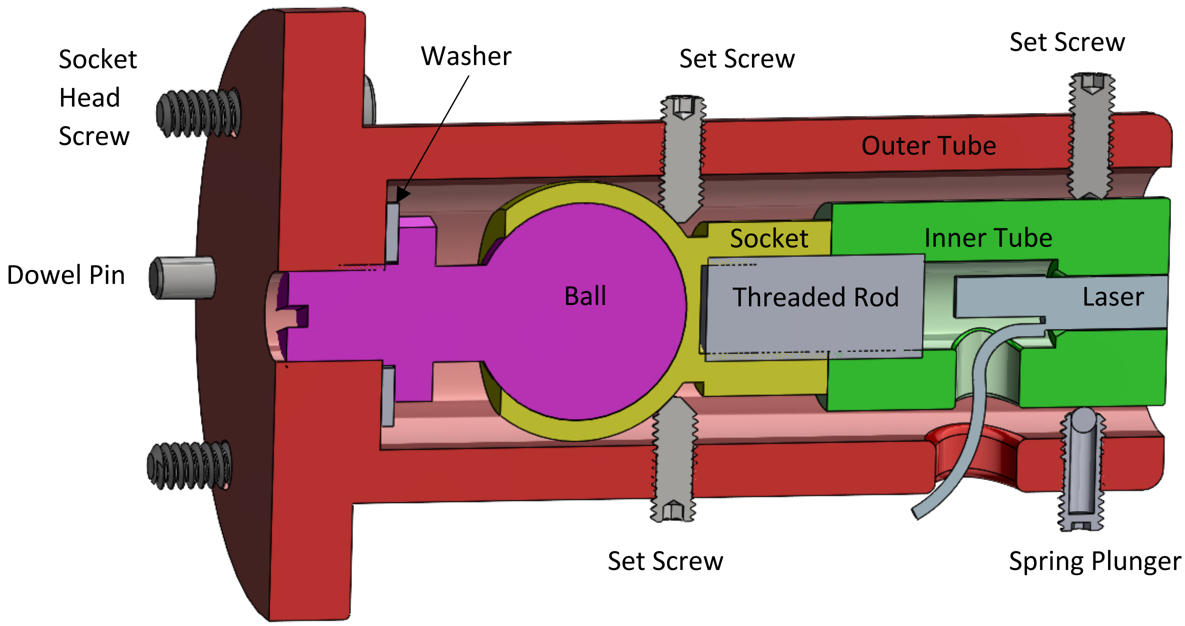
The Robot with the Head Mount Attached

Laser Holder

Goals:

- Attachment orthogonal to head base plate
- Laser aiming with precision of +/- .1 degree
- Ease of use
- Possibility of using a different laser with minor modification
- Sustained precision after extensive robot movement

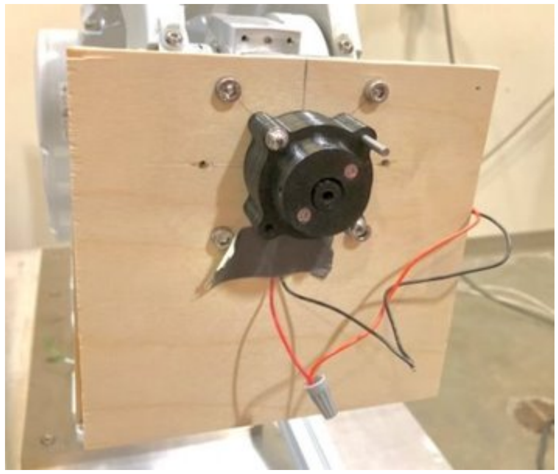




Laser holder – Section View

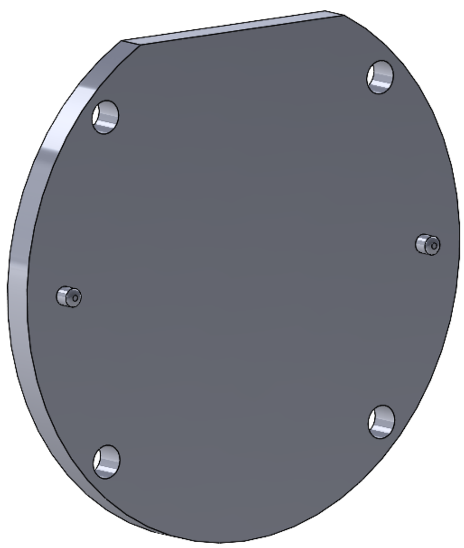
Due to manufacturing time constraints, we did not build this laser holder. We used the laser holder designed by the Eye/Head team in our calibrations. That laser holder was designed for deployment in their model's eye sockets. Because of this size constraint, their design is much smaller than ours. Due to the reduced length, it's direction is not quite as precisely adjustable as ours is predicted to be. This smaller design was meant to be milled from steel, but had to be made of 3-D printed plastic instead. Despite these compromises, this laser holder performed well enough to achieve a precision of +/- 0.05 degrees, which exceeds our target precision of 0.1 degrees.

The Eye/Head team's laser holder uses a mechanism similar to ours. We believe its success is a proof of concept for our design. Furthermore, we believe our design will work better. Milled steel will respond more smoothly and predictably to being pushed by the set screws and spring plungers. It will also have greater precision due to the longer barrel.



The Base Plate and Laser Holder we used in our calibrations

The robot was donated by Google with the wrist plate below attached. We kept the wrist plate for our apparatus because it was easier to mount to than the robot wrist without the plate.



Calibration Procedure

In order to calibrate the robot, ensure that the robot is properly zeroed by running the calibration wizard in the RC7 software. This provides a step-by-step process to align the robot using notches in the side of each joint. Once the robot is zeroed, there is a calibration procedure outlined in our documentation that should allow testing accuracy. This process goes through rotating the laser on the end effector of the robot to ensure that it is orthogonal to the wrist of the robot. For further details of the calibration process, refer to document in our supplemental materials. The following is a reference table that gives requirements for calibration measurements using various projection distances with the goal of achieving an accuracy for the laser of +/- 0.1 degrees from orthogonal to the base plate.

Accuracy Calibration Requirements

Distance to projection surface (m)	Required radius of circular tolerance (mm)
3	5.24
6	10.47
9	15.71
12	20.94

Calibration Results

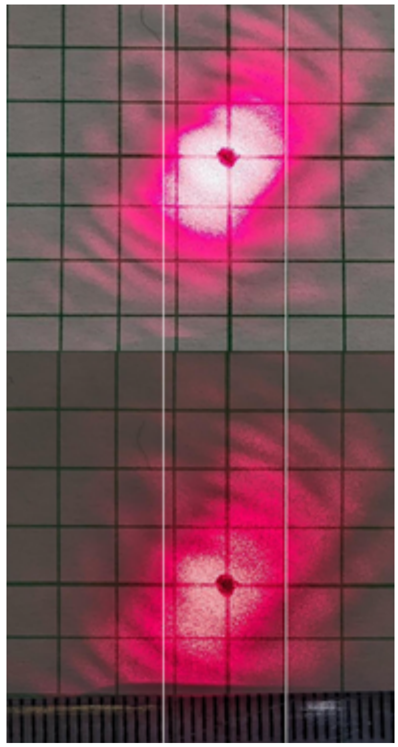
The first calibration run was executed by one person and took about 30 minutes. After this procedure, the discrepancy between where the robot was pointed and where the laser dot indicated was about 0.25 degrees.

The second calibration was more efficient as it was performed by two people. In this case the result was a precision of +/- .05 degrees of provable accuracy. This is determined by the laser dot being expanded when projected over a large distance causing it to have a maximum estimated radius of 12.7mm. The radius of the laser then would be 6.35mm meaning that this distance would be the smallest accuracy able to be accounted for which was well within the parameters set by the client of achieving 0.1 degrees of accuracy in the system. This calibration procedure is relatively simple to understand with an emphasis on being able to calibrate nicely through observation rather than rigorous measurements and adjustments. It can be performed relatively quickly with two people and performs much better than the required accuracy for the project.

Testing & Benchmarking

The point of this project was to develop a test apparatus that was not only accurate but reliable for long term testing. The robot performed exceptionally well for this spec. Two tests were performed on the system in order to check this, first a quick test was run where the robot would run its desired square pattern 1000 times causing 4000 motions in the system before it would return to the central point at which it started. This test was run on the device under test so that it could be seen if there was some sort of error after running several motions of the system. It was observed at this distance that there was no error in movement. After bringing this to the project mentor, he suggested running this test at the same distance we ran the calibration procedure to determine if errors were more noticeable in the system.

The second test was then performed where the distance between the laser and the surface it was projecting on was raised to about 6.3m. From this distance the same 1000 cycle pattern would run, but for this test the measurement process was repeated four times for a total of 4000 cycles or 16,000 movements of the system in total. After measurement and observation as shown in the “4000 cycle test” document, it was seen that even after each iteration the laser would remain in the same spot every time. This is likely due to the point-to-point system that the robot runs on, meaning that each point of the test is separately defined, and the robot simply moves to these points. In order for creep to happen, the robot would have to miss calls from the encoder within each motor. For a conservative observation, it was argued that a minimum 1mm movement was possible as this was the maximum accuracy of the test in both the x- and y-directions. This means that an error radius of sqrt(2) was possible or 1.41mm. From simple geometry it was found that the angular error from this test at most would be 0.013 degrees, well within the desired accuracy of the test rig. Below is a sample of the difference in laser location from the initial observation and one after 4000 cycles. For more information about the accuracy test pleaser reference A-5.



Discussion

To understand the need for this project, early in the process a few of the team members went into the eye team's lab and tried to calibrate to their Tobii eye tracker. It was found that most of us would have some sort of error in calibration where the place we would look and where the tracker would say we were looking would be skewed by about an inch in random directions based on who was running the test. These eye trackers are supposed to have a reliability of about 95% according to Jon, but, they have found that the reliability of these devices is closer to about 33%. The reason for the test rig is to eliminate any human error with calibration to create reliable metrics to improve the eye tracker technology for human use later.

The beginning state of the robot we were to use was not ideal. The robot was shipped to us from a Google donation and was strapped to a pallet. In addition, it was missing the end of the cable that plugged into the controller box. To get the robot operational, we had to source this cable end piece from Epson and get an electrician to install it.

From this point, the software of the robot was top priority to ensure it operated well. Epson gave our school some copies of their RC 7.0+ software to use for academic purposes and now that the robot was in place it was ready to start receiving orders. Several iterations of the square pattern program were made. The first of these implemented to the robot was implemented while the robot was oriented as shown:

While acting as a satisfactory proof of concept to run patterns on the robot, improvements were made upon advisement. The first method to improve this system was to cut out singularity points by putting the robot into an “elbow up” position. This position is commonly used with robots in industry to remove most singularities and allow for movement of the robot with less joint motion each time, improving overall accuracy.

One of the constraints on table construction was that the robot should not tip or move laterally during operation even during extremely fast movement or high acceleration. We tested this by programming the robot to move at maximum speed and acceleration/deceleration. During this test, the table did not move measurably due to the 80 kg of weights we placed on the lower shelf. The calculations used in this determination are in appendix 6.

Once the physical apparatus was completed, software became the focus. Two main programs were written to coincide with the requests of Jon from Microsoft. The first was to run the center of the face through a square pattern to run a perimeter within the screen area of the Surface Pro 7. The second was to test the rotation ability of the robot. This pattern ended up being a cross pattern where the face was centered on the screen and rotated to look up, down, left, and right. In both these tests the robot was able to perform very well. The only slight issue with the square pattern arose from the robot software itself. As the robot is programmed to specific points and not paths, if only the corners of the square were designated the robot would reach each point in any way it deemed fit. From reading software manuals, it was discovered that the robot prefers to use least joint motion to get from point to point. A result of this programming meant that the robot would run in a slight arch when moving left to right as all the joints in motion would move simultaneously only looking to rotate to point at the next designated point. This could of course be mitigated by adding in more points in between each corner point, with more accurate movement proportional to the number of intermediate points declared. For our test purposes though, this was not necessary. We only wished to prove that it could point where we wanted it to accurately.

Recommendations

Improvements to the Laser Holder

- Use the precision machined laser holder we designed. Precision machined parts with machine-tapped holes are likely to be an improvement over 3-D printed parts.
- Use spring plungers with longer pins. Adjustments should be made mostly using the set screws. Longer spring plunger pins would make this easier as they would have greater range of motion as the set screws are turned.
- Whether the existing laser holder or the more ideal design is used, the barrel of the laser holder should be made longer. Laser holder accuracy is directly proportional to the distance from the pivot point to the adjustment screws.
- Smaller pitch thread in set screw and spring plunger
- The laser itself could also be improved in later generations of the project. By running the calibration at a much longer length than testing length, we were able to get the robot to calibrate very accurately theoretically, but to make metrics to go with it, we would need to somehow negate the interference and fuzziness that was observed during this process.
- Ball tip hex wrenches for ease of adjustment of the laser.

Conclusion/Future Work

We found that the robot would provide accurate and repeatable movements consistent with human head movement for the purpose of testing eye tracker technology. The robot had virtually perfect accuracy with the largest error theorized to be 0.013 arc degrees of uncertainty for any given point of travel. To make this robot a viable testing tool for future groups, we suggest merging the eye team with the robot team to put everything together and integrate the two softwares to work together. This will require an understanding of inverse kinematics with parallax. The robot software for head positioning is relatively simple to accomplish, but the interpretation to cause the eyes to move exactly where they need to go may prove more difficult a task. Now that there is a working robotic platform however the next semester should have plenty of time to make this work.

Note that the CAD model in the files accompanying this report may not include the latest version of the Head and Eye Assembly. The latest version can be found in the Head/Eye Team's file submission.

Appendices

A-1 List of Deliverables

1. Calibration procedure
    a. Comprehensive manual that takes the user from startup of the robot to zeroing in on a particular point on the surface screen.
    b. The programs we use to perform calibration.
2. Calibration results
    a. Testing documentation and results
    b. Notes and observations on the procedure
3. Design documentation
    a. CAD, Code, Drawing, and Bill of Materials.
    b. Make it feasible for anybody to reproduce our design 10 times.
    c. Upload to the existing Microsoft project GitHub.
4. Physical Apparatus
    a. Robot Table
    b. “H” square
    c. Tablet stand
    d. Head mount
    e. Laser holder/pointer
    f. Weights
5. Miscellaneous tools
    a. Levels
    b. Allen keys
6. Electronic components
    a. Surface tablet
    b. Tobii

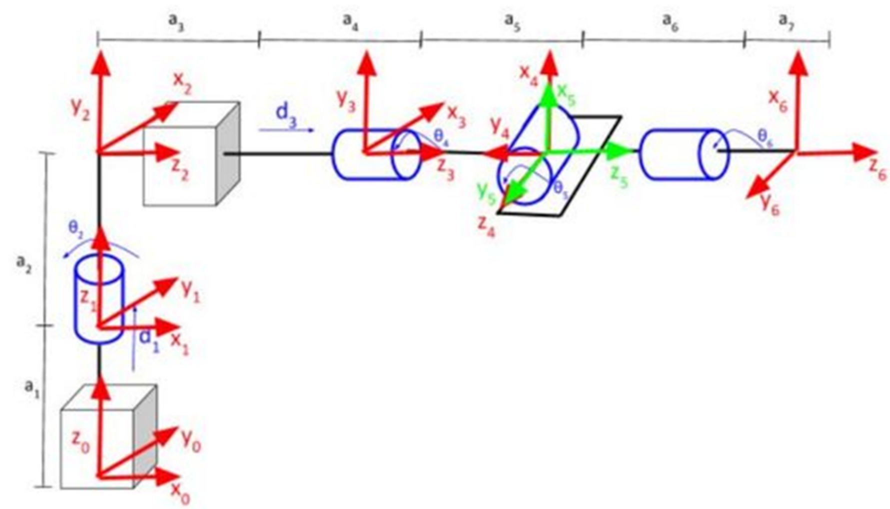
A-2 Where to find documentation

CAD, Code, BOM, can be found in the final deliverables folder in basecamp or the EyeTracker GitHub upon authorization by Jon Campbell.
https://github.com/Gleason-Institute-Org/EyeTrackingTestRig/

A-3 6 Degree of Freedom Inverse Kinematics

Inverse kinematic is used for trajectory planning and dynamic analysis of robot. The Epson C8 Robot has six angle of joints, which means the robot is able to move in six angles and allow us to estimate where the end tip of robot arm will be placed in term of X, Y, and Z coordinates. There are two type of approach which are analytical approach and numerical approach, and we used Analytical approach. The Analytical approach does not require initial guesses for the joint angle.

Analytical approach:



$$\theta_2 = tan^{-1}\left(\frac{y}{x}\right)$$

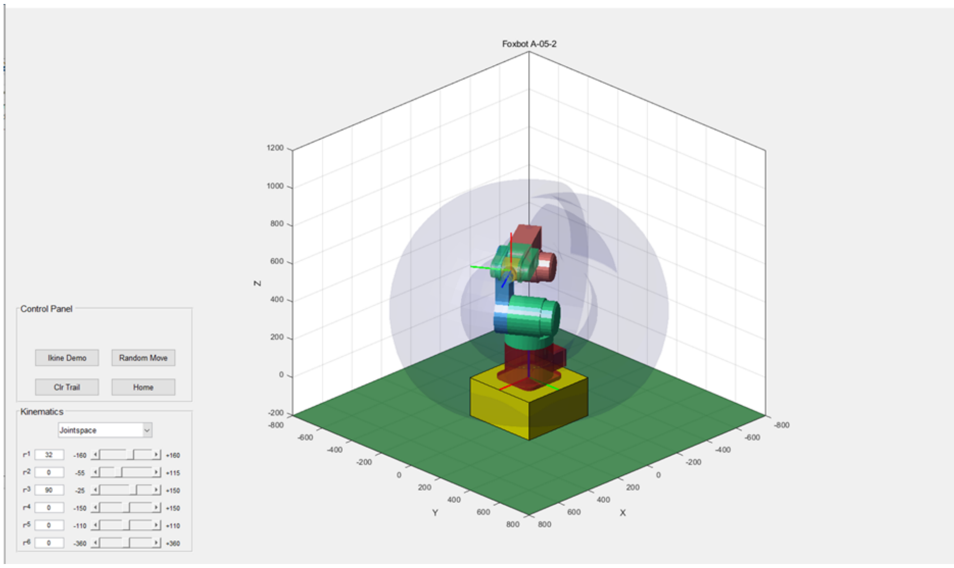
$$R_6^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_3^0 = \begin{bmatrix} -\sin\theta_2 & 0 & \cos\theta_2 \\ \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R_6^3 = (R_3^0)^{-1} R_6^0$$

$$R_6^3 = \begin{bmatrix} -\sin\theta_4\cos\theta_5\cos\theta_6 - \cos\theta_4\sin\theta_6 & \sin\theta_4\cos\theta_5\sin\theta_6 - \cos\theta_4\cos\theta_6 & -\sin\theta_4\sin\theta_5 \\ \cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6 & -\cos\theta_4\cos\theta_5\sin\theta_6 - \sin\theta_4\cos\theta_6 & \cos\theta_4\sin\theta_5 \\ -\sin\theta_5\cos\theta_6 & \sin\theta_5\sin\theta_6 & \cos\theta_5 \end{bmatrix}$$

Simulation result



For the simulation, we were able to get the matlab simulation file from (https://ww2.mathworks.cn/matlabcentral/profile/authors/6364791-ramoflaple)
This simulation is not 100% accurate in real situation because the arm length is different. However we were able to find out the basic idea of the 6 DOF inverse kinematics.

Referencese for inverse kinematic

https://ww2.mathworks.cn/matlabcentral/profile/authors/6364791-ramoflaple

A-4 Robot Programs

In the event that the recipient of this folder cannot open the program files provided here is a list of screenshots with the relevant code structure to build these codes in the same or other robot programs. Please reference the safety manual for startup procedure of the robot to ensure safety. After the controller box is turned on, ensure the USB connection is turned on in the RC7.0+ software. For future convention, it will be a good idea to name the program the same name as the function being called as the run window goes by the functions in the program not the name. i.e. for RotationCross.prg, find RotationFromOrigin in the run window to execute the parent program. Also note that functions operate system wide regardless of the parent program. i.e. InitRobot can be called by multiple programs even though it is only declared in one program.

Square Pattern:

SquarePattern.prg

```
Function SquarePattern
    Call InitRobot
    Go P2
    Call SquarePat
Fend
Function InitRobot
    Reset
    If Motor = Off Then
        Motor On
    EndIf
    Power High
    Speed 50
    Accel 25, 25
Fend
Function SquarePat
    Real i, j
    For j = 1 To 1
        For i = 3 To 6
            Go P(i)
        Next
    Next j
    Go P2
Fend
```

RotationCross:

RotationCross.prg

```
Function RotationFromOrigin
    Call InitRobot
    Go P2
    Call RotCross
Fend
Function RotCross
    Real i, j
    For j = 1 To 8
        For i = 11 To 14
            Go P(i)
            Go P2
        Next
    Next j
    Go P2
Fend
```

CalibTest2:

CalibTest2.prg

```
Function SquareCalib2
    Call InitRobot
    Go P16
    Call SquareTest2
Fend
Function SquareTest2
    Real i, j
    For j = 1 To 1000
        For i = 17 To 20
            Go P(i)
        Next
        Print j
    Next j
    Go P16
Fend
```

A-5 Sustained Accuracy Test

Abstract

This test is a proof of accuracy to a higher level within our system. This will be useful to show the limit of creep the robot provides over long term use. Test was run 4 times so that the end point can be measured each time to determine deviation from the center point. The idea was to check the creep at 1000 to 4000 cycles of the pattern. With the square motion this would mean that the robot would be moving a total of 16,000 times throughout the test. It was found that the robot performed nearly perfectly at a distance of about 20.5 ft, providing an estimated accuracy of 0.013 degrees radially after running the test.

Procedure

To run this test, the robot was programmed to run a square pattern similarly to the square pattern delivered to the client, however some slight adjustments were made namely the height the robot would operate at to hit a certain point of the wall on the opposite end of the room rather than a tablet placed in front of it. From this central point, four other points were generated and placed on the point list after this initial point. The program was written in the following way:

```
Function SquareCalib2
    Call InitRobot
    Go P16
    Call SquareTest2
Fend
```

```
Function SquareTest2
    Real i, j
    For j = 1 To 1000
        For i = 17 To 20
            Go P(i)
        Next
        Print j
    Next j
    Go P16
Fend
```

```
Function InitRobot
    Reset
    If Motor = Off Then
        Motor On
    EndIf
    Power High
    Speed 50
    Accel 25, 25
Fend
```

The program initializes the robot to be on and given maximum power. This will ensure the robot moves at an adequate speed as low power is locked to low speed. The speed and acceleration values are then defined, these are given as percentages of maximum speed with the Accel function describing both acceleration and deceleration as it approaches a point. In this case speed is set to 50% and accelerations are both set to 25%. These values were chosen to keep a good pace for testing while keeping low vibration of the system. The function of the SquareTest2 function utilizes a nested for loop where the first loop iterates how many times the test should be run and the second describing the points to run through. To keep track of which test iteration the system is on, the value of the test loop is printed every time it runs through. At the end of the test the program returns to its start position for measurement. The program operates with the following points listed:

16	CalibHigh	-23.869	615.700	560.000	0.000	21.123	-90.000	0		Righty	Above	Flip
17	SqTest2TL	-119.116	615.700	585.000	0.000	21.123	-90.000	0		Righty	Above	Flip
18	SqTest2TR	-119.116	615.700	520.000	0.000	21.123	-90.000	0		Righty	Above	Flip
19	SqTest2BR	71.381	615.700	520.000	0.000	21.123	-90.000	0		Righty	Above	Flip
20	SqTest2BL	71.381	615.700	585.000	0.000	21.123	-90.000	0		Righty	Above	Flip

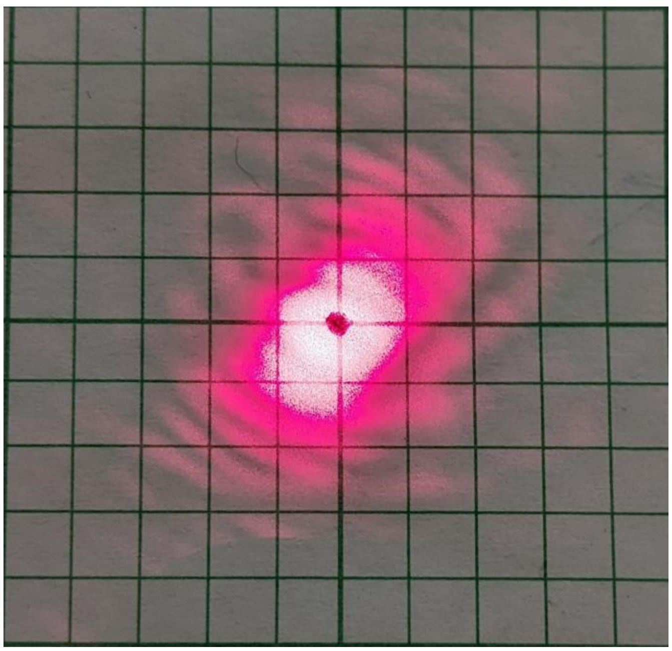
These points are similar to the points used in the square patter procedure on the device under test, though the z values are adjusted to be larger to point the robot higher, the angles always remain constant to keep each point on the same plane and orientation. The first point listed is X, where X in the robot's reference axis moves it left to right. The second one is Y which moves the robot forward and back, for the purposes of this test it will remain constant. Finally, the Z axis controls the height of the robot, meaning the top orientation should have a larger number than the bottom. The following 3 points are the respective rotations around these axes relative to the base of the robot. The reason that the y-axis is rotated to 21.123 degrees is that there is a flat spot on the end effector mount of the robot that is oriented to the negative of this angle, in order to have a nice flat spot to orient our mount to this was rotated such that this flat spot would be parallel with the floor. The next 3 boxes on any given line are the corresponding rotations and the "Righty, Above, Flip" boxes define certain parameters about the robot's interpretations of these axes. For the purposes of this test to keep movement to a minimum, these should all be set to the values described above.*

**Note: The orientation interpretation is determined by the "Righty, Above, Flip" boxes. If the "Flip" box is "No Flip", the program will take the inverse of everything and will cause the robot to try and orient the laser to face itself.*

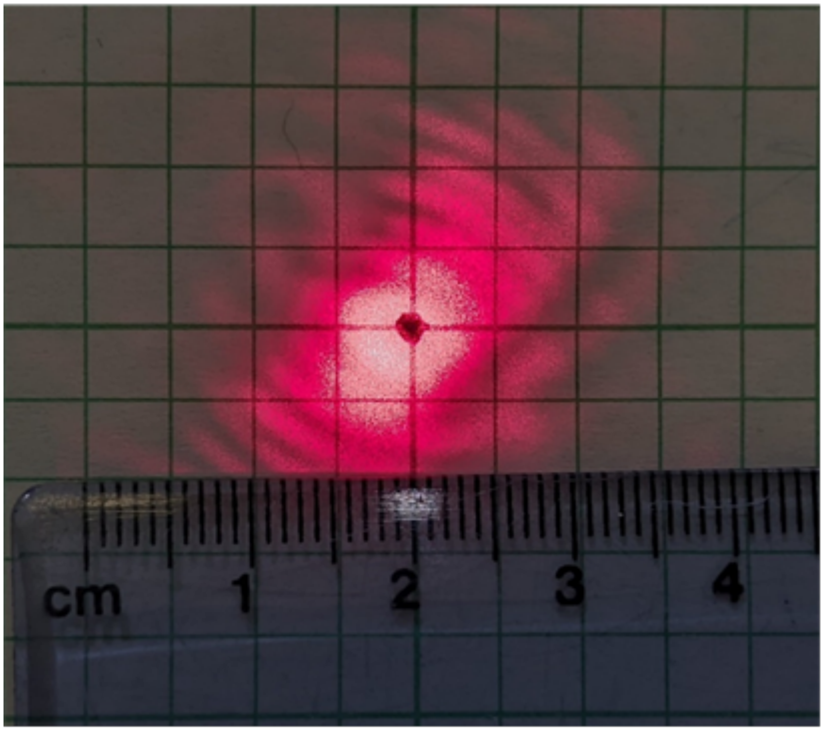
Test Results

For a baseline the laser was pointed to a specific grid intersection on the paper which was then indicated with a black dot made in pen.

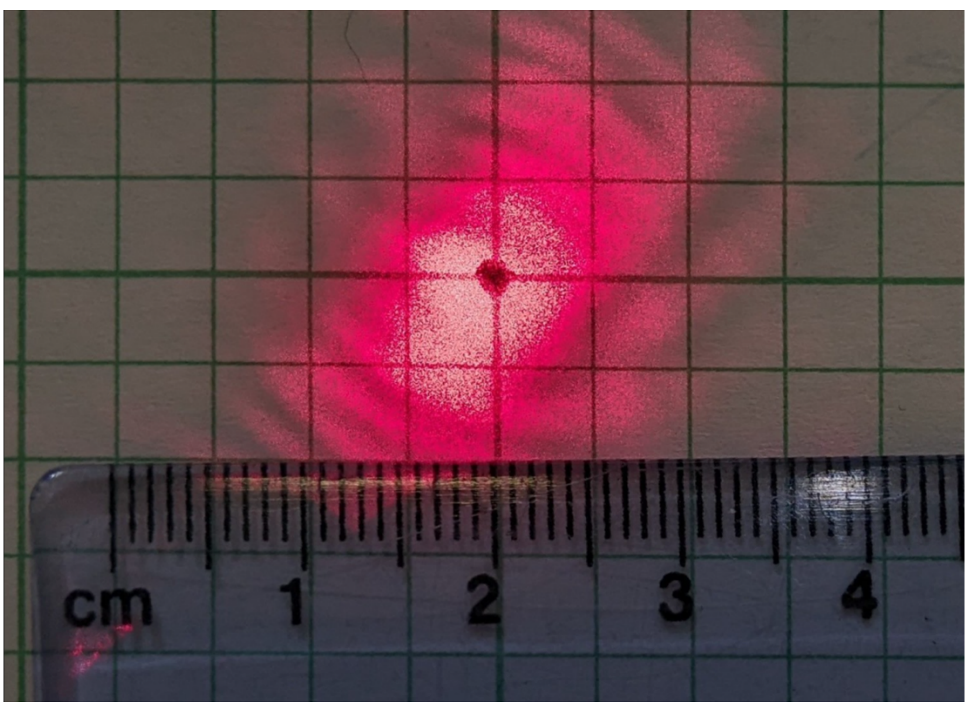
Original position:



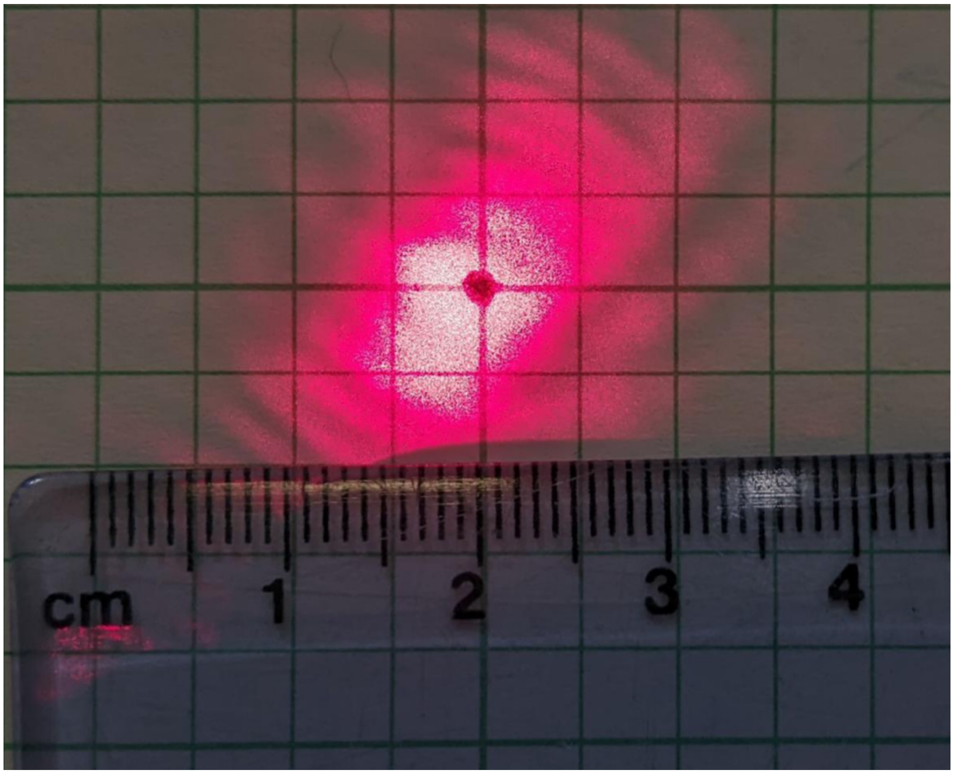
After 1000 cycles:



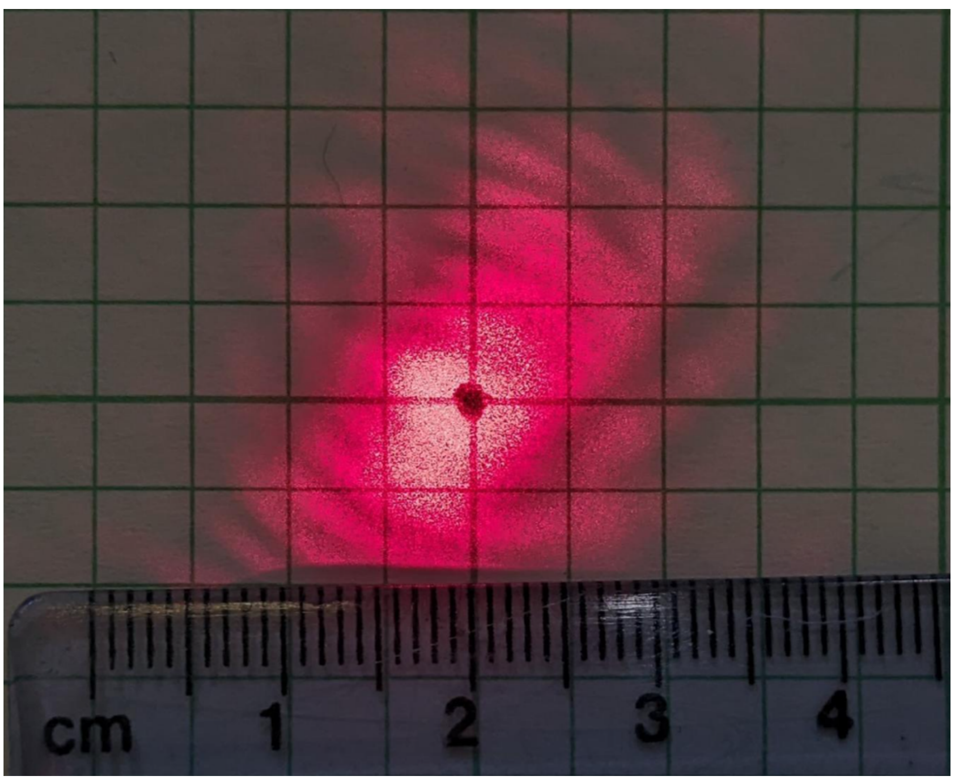
After 2000 cycles:



After 3000 cycles:

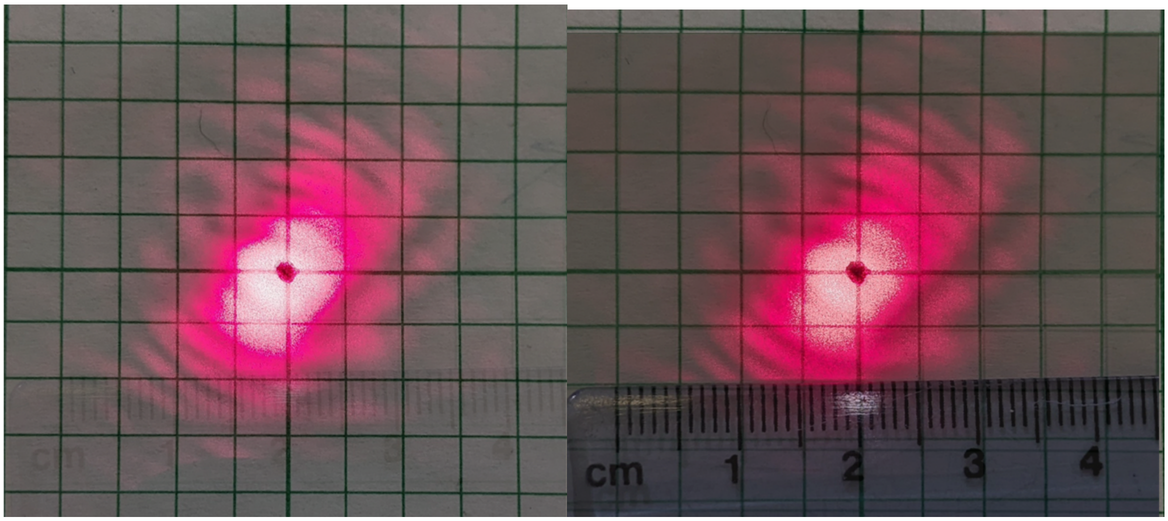


After 4000 cycles:

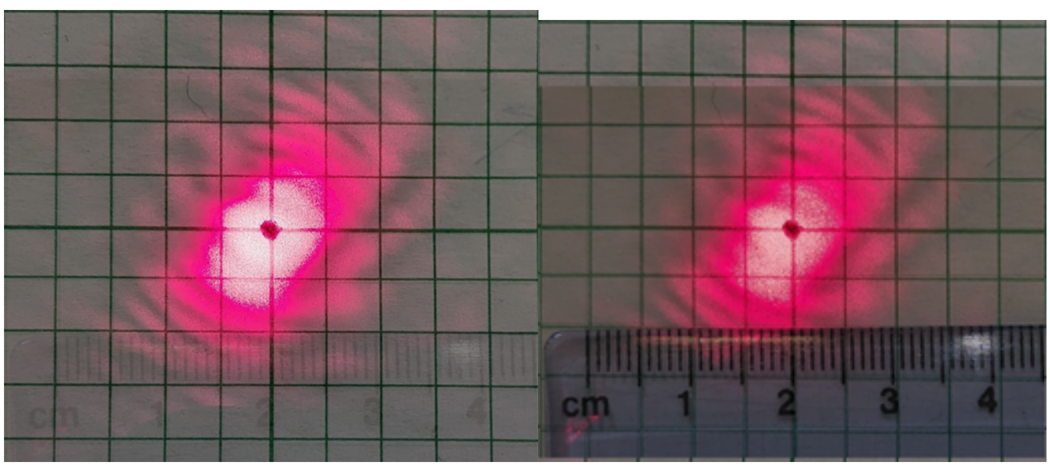


In order to compare these pictures, they were transferred to PowerPoint where they were overlayed with each other. The initial picture was placed in the back of the frame while the test images were placed on top of the initial. The test image was then made transparent, and the grid lines adjusted until they matched. Then as the transparency was adjusted the difference between the initial and final results became apparent as shown:

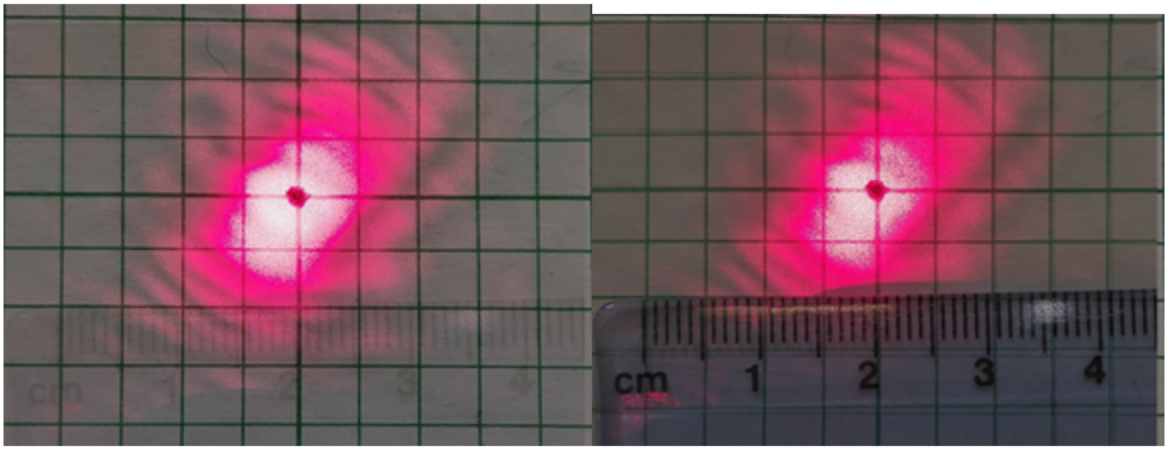
1000 cycles:



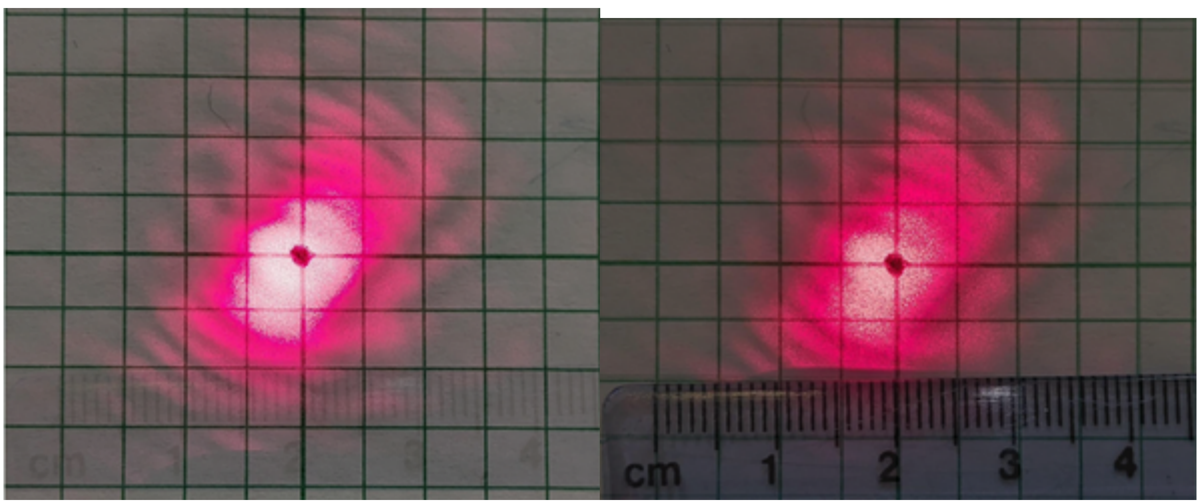
2000 cycles:



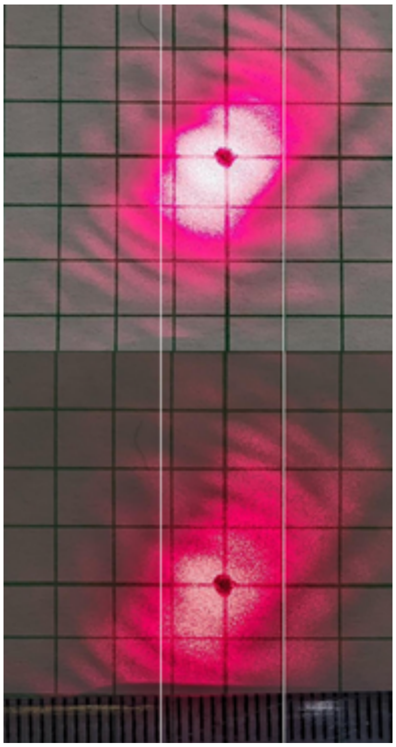
3000 cycles:



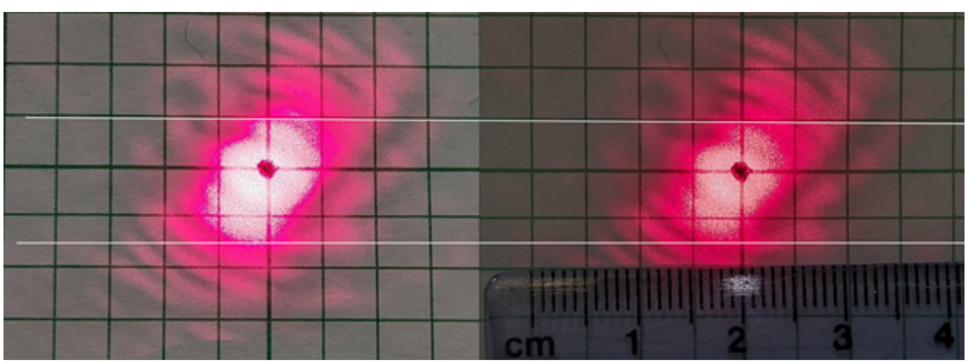
4000 cycles:



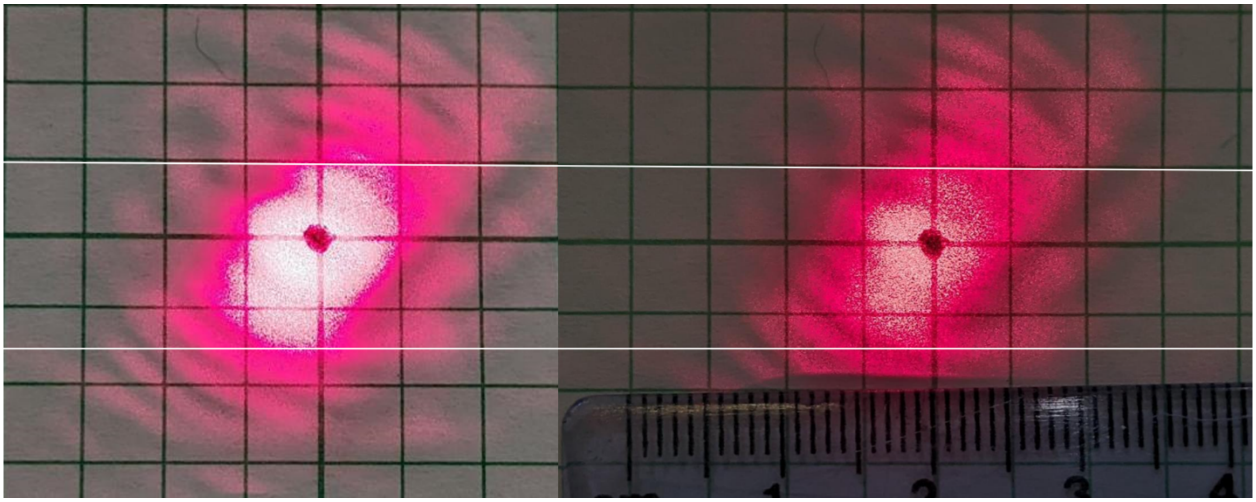
From the analysis of this picture there is no discernable error in the displacement of the laser from the original point to the final point after performing this test. Some pictures of later tests seem to make the right side of the main light of the laser become smaller, but the left side still lines up nicely with the grid on the left, making it difficult to argue that it is traveling to the left after the test. This can be shown with vertical lines superimposed on the pictures as shown for the 4000-cycle case:



The vertical offset was checked by lining up the grids in the pictures in such a way that the size and locations matched up and then drawing lines that were locked horizontally at the top and bottom of the pictures as shown for the 1000 cycle case:



A similar observation can be made for the 4000 cycles:



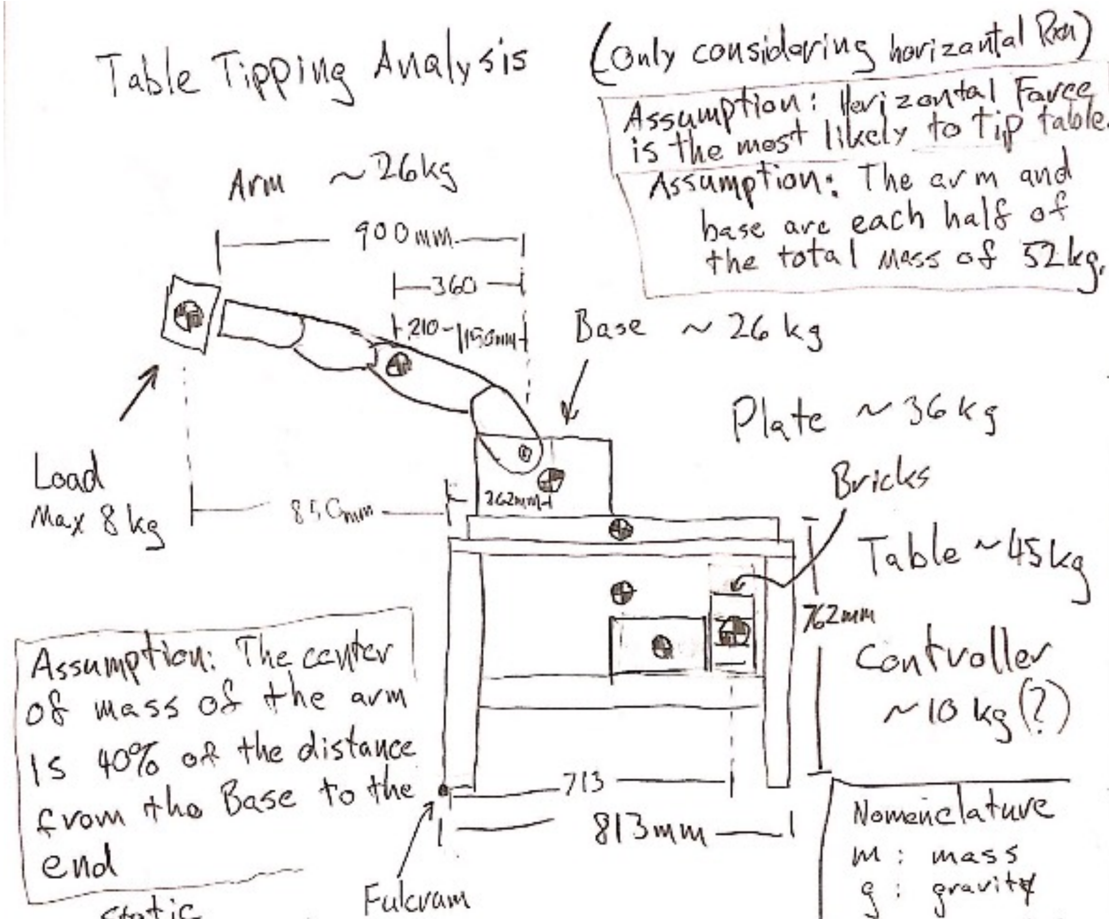
As it is shown the laser on the right only gets smaller within the bounds of the first laser. In order to argue for conservation's sake, it can be argued that the maximum deviation of the laser at this range would be 1mm or 1/5 the width of a box. For radial error then, the maximum distance traveled would the 1.41mm. This means at a distance of 20.5 ft or 6250mm the angular error of the system would be 0.013 degrees, which is well within the desired angular tolerance of 0.1 degrees for this system to run its necessary tests.

There was a noticeable error present while the robot was in motion. This could likely be due to many factors within the system including but not limited to: vibration in the system, the spring reaction of the movements, the digital encoder movement of the robot, and the least joint motion of the robot in conjunction with parallel motion of joints in order to reach the next known point. It should be noted however that even though there was error while moving, it did not ultimately affect the end test results as the laser ended up in the same position It started in and ran a consistent pattern the entirety of the test.

Conclusion

The Epson C8 robot with proper laser calibration provides a highly accurate solution for motion control within the context of the eye tracker test required by Microsoft. These robots are designed for highly accurate and precise movements as well as a spherical range of motion allowing for any head movement desired to test the capabilities of the eye tracker. After a trial of 16,000 motions or 4000 cycles the robot was still able to find its exact point at a large distance. The final proven accuracy is a factor of 7-8 higher than the desired accuracy stated by Microsoft conservatively. This is likely due to the robot using point to point motion and its good design. With point-to-point motion, each point can be independently stored in the system making it unlikely that creep would arise out of this system simply by making it run a pattern. A recommendation for future iterations of this project is to acquire a better laser for calibration measurements as the fuzziness along with the expansion of this laser made it difficult to make extremely precise measurements.

A-6 Table Tipping Analysis (from which we deduced that 80 kg of weight should be placed on the shelf at the back end of the table)



Static

$\Sigma M =$

$$m_{Load}\, g \cdot A_{Load} + m_{arm}\, g \cdot A_{arm} - m_{Base}\, g\, A_{Base} - m_{plate}\, g \cdot A_{plate} - m_{table}\, g \cdot A_{table} - m_{cont}\, g\, A_{cont}$$

$$g(8kg\,850mm + 26kg\,360mm - 26kg\,262mm - 36kg\,407mm - 45kg\,407mm - 10kg\,203mm)$$

$\Sigma M = -232\ Nm$ in the static case. (this is clockwise)

Dynamic

Maximum Horizontal reaction force is 1300 N. The 3rd law counterpart of this force, applied at the bottom of the base of the robot causes a moment of 991 N·m counterclockwise. This will lift the back legs of the table off the floor.