Department Aerospace Engineering

Faculty of Engineering & Architectural Science

Course Code	AER 627
Section	02
Course Title	Introduction to Space Robotics
Semester/Year	Winter 2021
Instructor	John Enright
TA	Joel Moore
Project 3	Drawing Robot
Submission Date	March 26th 2021

Name	Student ID	Signature*
Jann Cristobal	500815181	Jann C.
Stephanie Chan	500819304	SC

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*
http://www.ryerson.ca/senate/policies/pol60.pdf

Abstract

In this project, the drawing robot is able to do its intended tasks sufficiently. The straight lines are slightly curved in comparison to the ideal shapes. Inverse kinematics is the method used to solve the joint angle requirements. The sequence of angles are converted to the motor angles to do the motion. The coefficients for the cubic spline are calculated to represent the joint trajectory. The actual workspace area is significantly lower in comparison to the ideal case. This is due to the physical limitations of the structure. Particularly the width of the link, reducing the overall range for both the links.

Table of Contents

Table of Contents 2
1 Introduction 4
2 Theory 4
2.1 Package A: Jann Cristobal 4
2.2 Package B: Stephanie Chan 6
3 Design Documentation 7
3.1 Package A: Jann Cristobal 7
3.1.1 Physical Design 7
3.1.2 Workspace Area 10
3.1.3 Final Design 12
3.2 Package B: Stephanie Chan 14
4 Project Operations 15
5 Conclusion 18
6 References 19
7 Appendix 20
7.1 Simulated Results 20
7.2 Actual Results 24

List of Figures

Figure 2.1: Cantilevered Beam with loading at free end [1]. 4
Figure 2.2: Two-link manipulator Workspace Area [3]. 5
Figure 3.1: TRAC3R [4]. 7
Figure 3.2: Alternate Build Geometry [5]. 7
Figure 3.3: Initial link 1 design 8
Figure 3.4: Link 1 and link 2 final design 8
Figure 3.5: Final design for the base structure 9
Figure 3.6: Joint 1 gear train 9
Figure 3.7: Joint 2 gear train 9
Figure 3.8: Homing Mechanism 10
Figure 3.9: Joint 1 maximum angle 10
Figure 3.10: Joint 2 maximum angle 11
Figure 3.11: Workspace Area 11
Figure 3.12: Isometric View 12
Figure 3.13: Front View 12
Figure 3.14: Top View 13
Figure 3.15: Right View 13
Figure 4.1: Drawing Robot operation flowchart 15
Figure 4.2: Inverse Jacobian Control 18
Figure 7.1: Cartesian Trajectory for Horizontal Line 20
Figure 7.2: Joint Trajectory for Horizontal Line 20
Figure 7.3: Cartesian Trajectory for Vertical Line 21
Figure 7.4: Joint Trajectory for Vertical Line 21
Figure 7.5: Cartesian Trajectory for Triangle 22
Figure 7.6: Joint Trajectory for Triangle 22
Figure 7.7: Cartesian Trajectory for Square 23
Figure 7.8: Joint Trajectory for Square 23
Figure 7.9: Video Demonstration Results 24

1 Introduction

In this project, the goal is to design a robot that can writing or draw the desired shape with the use of inverse kinematics and inverse Jacobians.

2 Theory

This section outlines the mathematical theories utilized in designing and programming the robot.

2.1 Package A: Jann Cristobal

For physical structure of the drawing robot, there are several design considerations that will affect the performance as well as add additional constraints to the robot's workspace.

The first consideration is the mass distribution of all the components. The overall design must consider all the moving parts are ensure that the base is robust and can withstand any moment and force produced by the links. By having a low centre of gravity and having heavy stationary components on the opposite side of the arm link will greatly improve structural stability.

Second consideration is with regards to reducing bending of the arm links. The arm links act similar to a cantilever beams with loading on the free end. Figure 2.1 shows how the arm links can be affected by loading at the free ends.

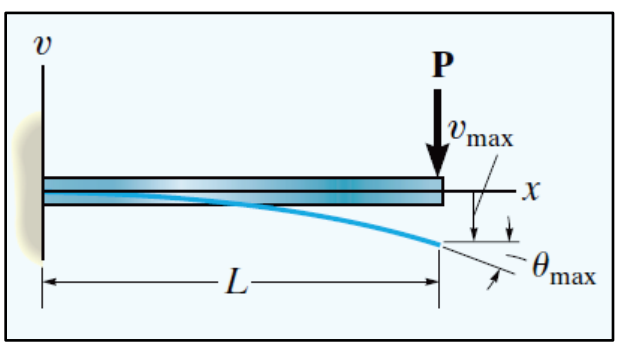

Figure 2.1: Cantilevered Beam with loading at free end [1].

$$v_{max} = \frac{-PL^2}{2EI} \tag{2.1}$$

Equation 2.1 shows the maximum deflection of the beam. The goal is to reduce this value by reducing the load, P, and link length, L, or increasing the Modulus of Elasticity, E, and Moment of Inertia I.

Another consideration is related to the gear ratio to utilize mechanical advantage. A low gear ratio amplifies the output velocity at the expense of input torque. The opposite is

true for a low gear ratio. Clearly, using the proper gear ratio will yield the maximized performance [2]. The expression for gear ratio is

$$R = \frac{\omega_A}{\omega_B} = \frac{r_B}{r_A} = \frac{N_B}{N_A} \tag{2.2}$$

Where A is the driving gear, B is the driven gear, ω is the angular velocity, r is the radius, and N is the number of teeth.

Additional consideration is with regards to the available workspace area. A good starting point is to only consider the link lengths and the area that it can cover based on the range of joint angles.

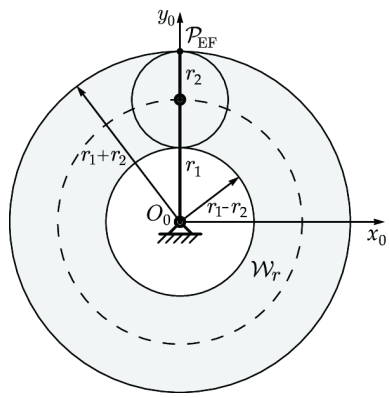

Figure 2.2: Two-link manipulator Workspace Area [3].

Figure 2.2 shows the available workspace for a full 360° range of motion for both links 1 and 2 are shaded grey area. Once the link lengths are determined, further constraints that will limit the range of motion include the width of the links, and the interaction with other components of the structures.

Lastly, both of the links must have a reference point that can be considered as the 0°. This can be achieved by having the links align along the horizontal plane which will reset or home the mechanism.

2.2 Package B: Stephanie Chan

3 Design Documentation

This section will show the design process involve of building and programming a two-link drawing robot.

3.1 Package A: Jann Cristobal

When coming up with ideas for the robot design, there are two existing designs that were used as reference. Figure 3.1 shows the TRAC3R by Christoph Stahl [4]. Figure 3.2 shows Alternate Build Geometry by John Enright [5].

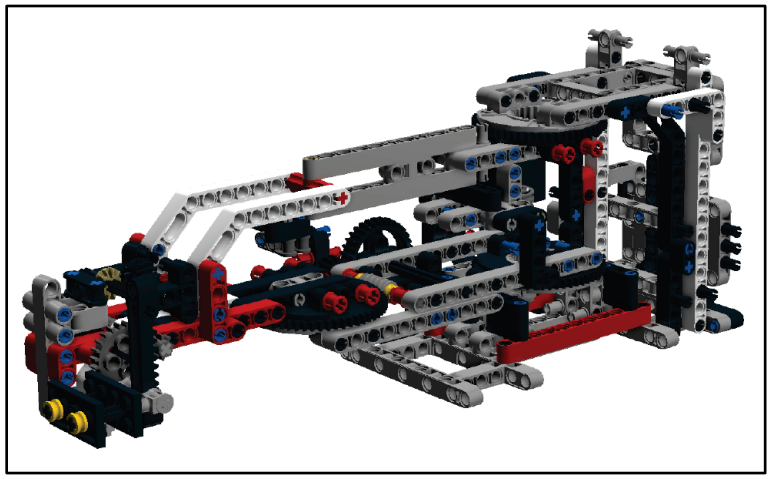

Figure 3.1: TRAC3R [4].

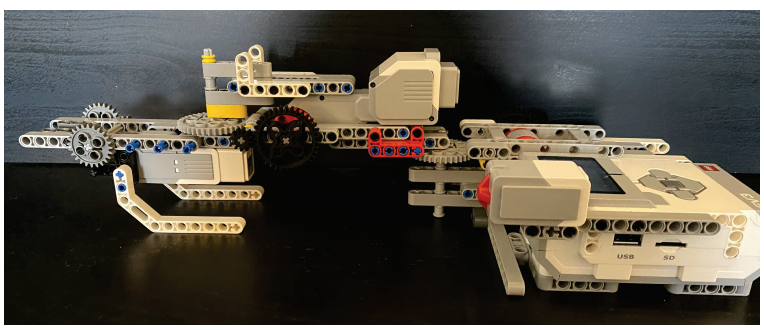

Figure 3.2: Alternate Build Geometry [5].

3.1.1 Physical Design

The most logical components to design first were the arm links for the planar RR manipulator. The ratio between the link length 1 and 2 should be between 3:2 and 2:1, respectively. By having a shorter link 2, the range of motion for the both the links will be wider at expense of smaller workspace area. Figure 3.3 shows the initial design concept for link 1.

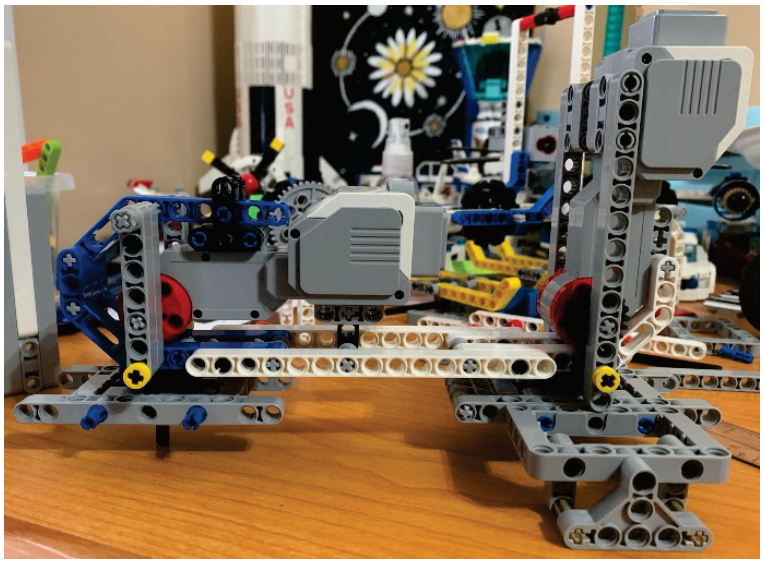

Figure 3.3: Initial link 1 design

After more revisions, the final design ended up being a combination of the two existing designs as seen on Figure 3.4

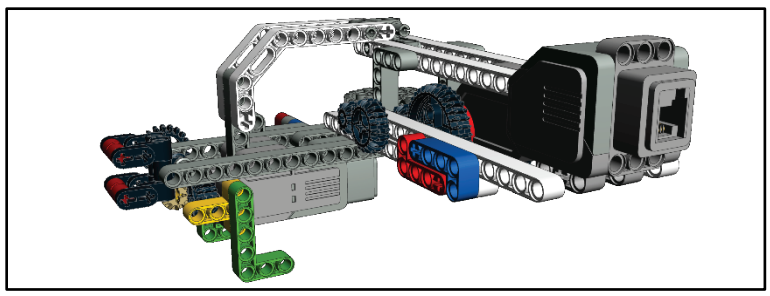

Figure 3.4: Link 1 and link 2 final design

Figure 3.4 shows the addition of supports for link 2 to reduce bending and flexing. Given the width of the links, the range for joint 1 is $0° \leq \theta_1 \leq 150°$ with joint 2 at $0°$. For joint 2, the range is $0° \leq \theta_2 \leq 135°$. With these constraints, the two-link manipulator will be right-handed with the $0°$ line along the x-axis of joint 1.

The next part is designing the base where joint 1 is placed. Keeping in mind that the two links are situated on one side of the system, the ideal place for the motor is the opposite side in order to distribute the mass more evenly. Figure 3.5 shows the construction of the base.

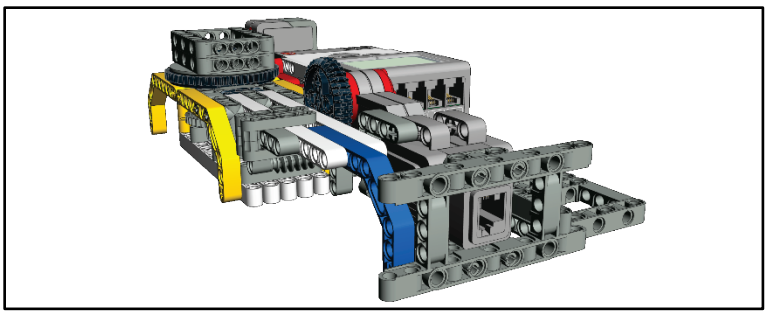

Figure 3.5: Final design for the base structure

In terms of the gear ratio, joint 1 is 3:40 as shown in Figure 3.6, while joint 2 is 1:8 as shown in Figure 3.7. The gear reduction is required in order to produce more torque for both arm links to rotate.

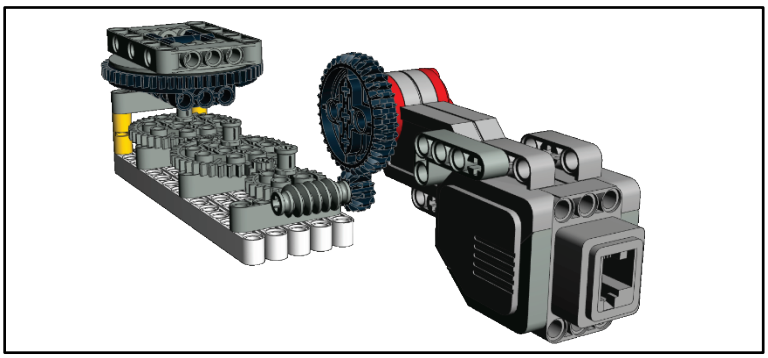

Figure 3.6: Joint 1 gear train

$$R_1 = \frac{36}{12} \times \frac{1}{24} \times \frac{24}{40} \times \frac{40}{40} = \frac{3}{40}$$

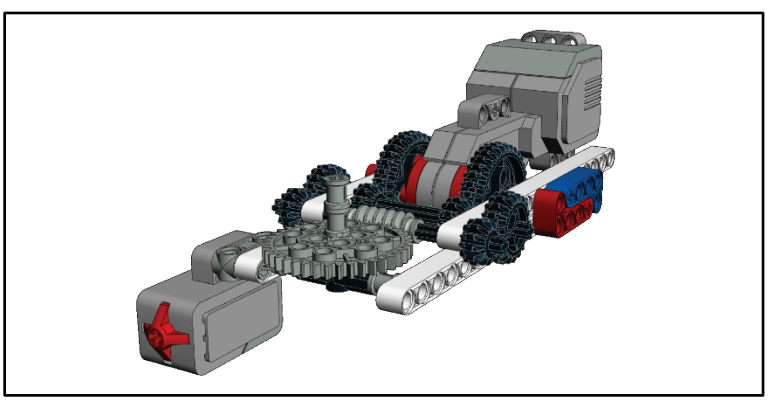

Figure 3.7: Joint 2 gear train

$$R_2 = \frac{36}{12} \times \frac{20}{12} \times \frac{1}{40} = \frac{1}{8}$$

The next part is designing the homing mechanism. This was achieved by placing a touch sensor on the base of the structure to zero the joint 1 when link 1 touches the sensor. Link 2 is homed similarly where the touch sensor is attached to link 1. In Figure 3.8 link 1 is touching the sensor therefore it is homed, while link 2 is $90°$ with respect its homed position.

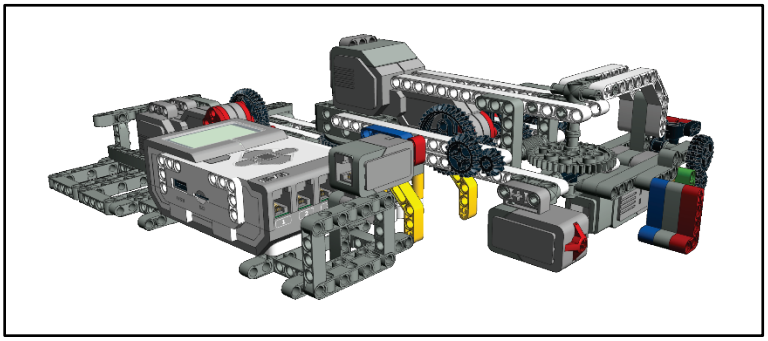

Figure 3.8: Homing Mechanism

Lastly is the design for the pen holder. Due to the limited number of touch sensors, the mechanism to home the pen must be done manually.

3.1.2 Workspace Area

To determine the workspace area, the range for each joint must first be determined. The minimum angle for both joints are $0°$. In Figure 3.9 the maximum angle for joint 1 is $150°$ when joint 2 is homed given the physical constraints of the design. In Figure 3.10 the maximum angle for joint 2 is $135°$ given the physical constraints of the design. To accommodate both limitations, the range for joint 1 must be $0° \leq \theta_1 \leq 90°$, while the range for joint 2 stays as $0° \leq \theta_1 \leq 135°$.

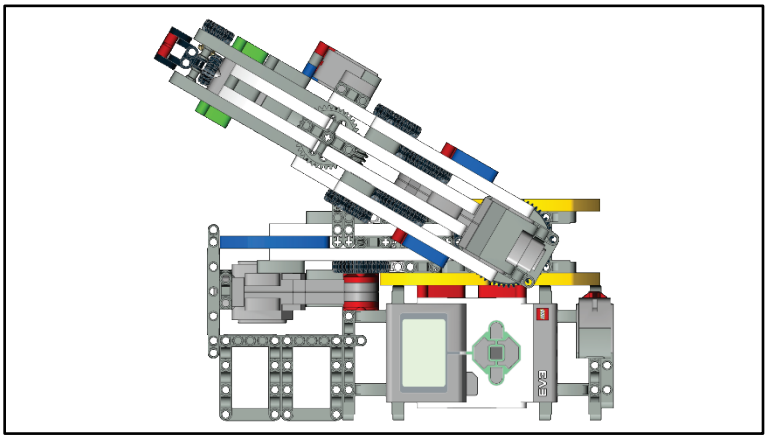

Figure 3.9: Joint 1 maximum angle

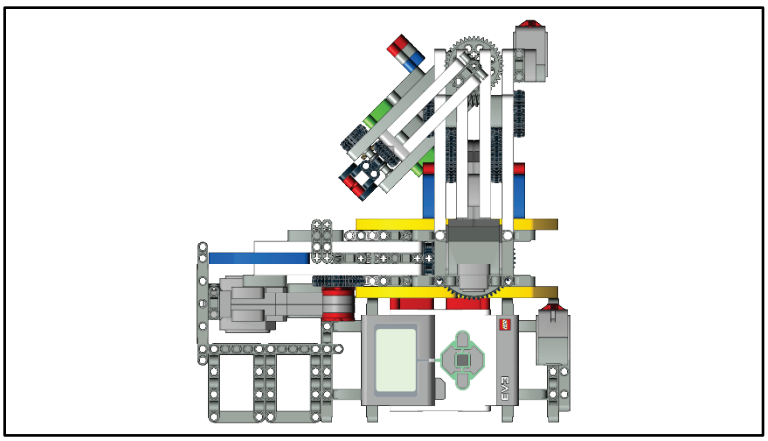

Figure 3.10: Joint 2 maximum angle

Figure 3.11 shows the ideal workspace and actual workspace of the robot. Due to external factors such as friction, overshoot, link flexibility and backlash, the actual workspace was kept within a 12cm-by-12cm area. Trying to reach the edge of the ideal workspace is difficult due to singularities. And avoiding below 10cm vertically ensures that link 1 is well within the 0–90-degree range. The actual workspace area is sufficient for the purpose of this project.

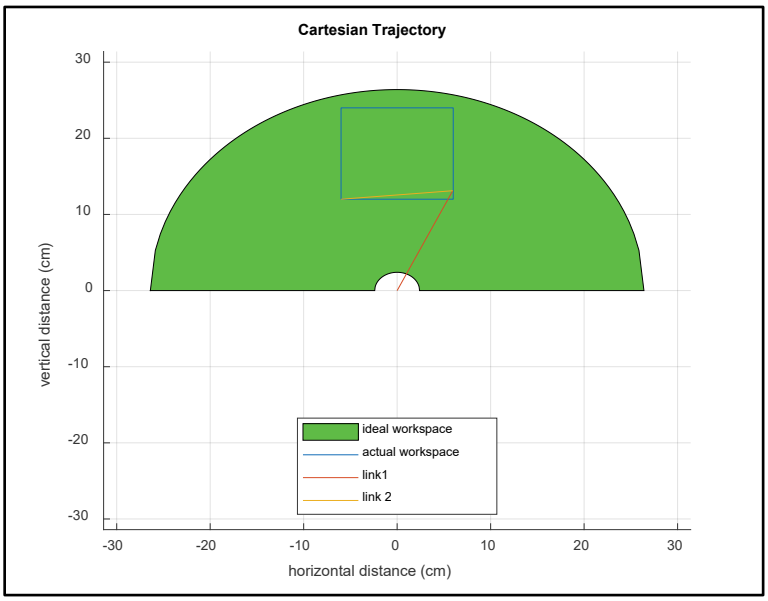

Figure 3.11: Workspace Area

3.1.3 Final Design

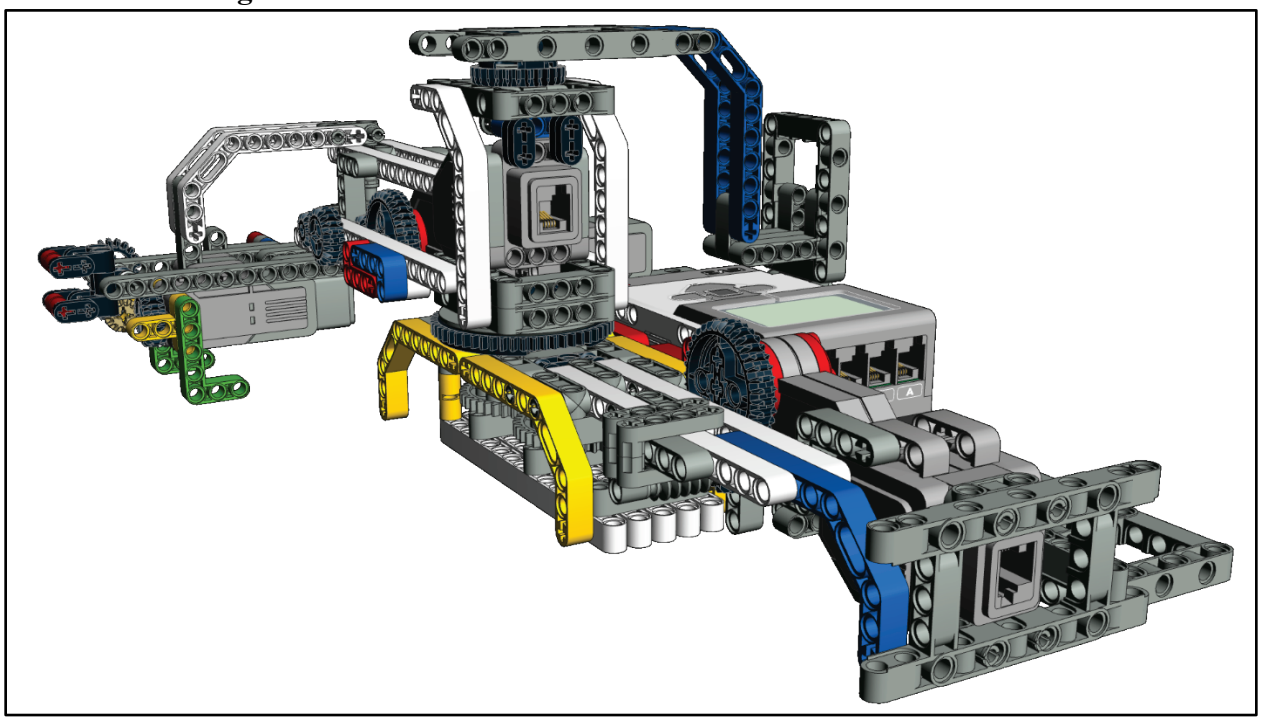

Figure 3.12: Isometric View

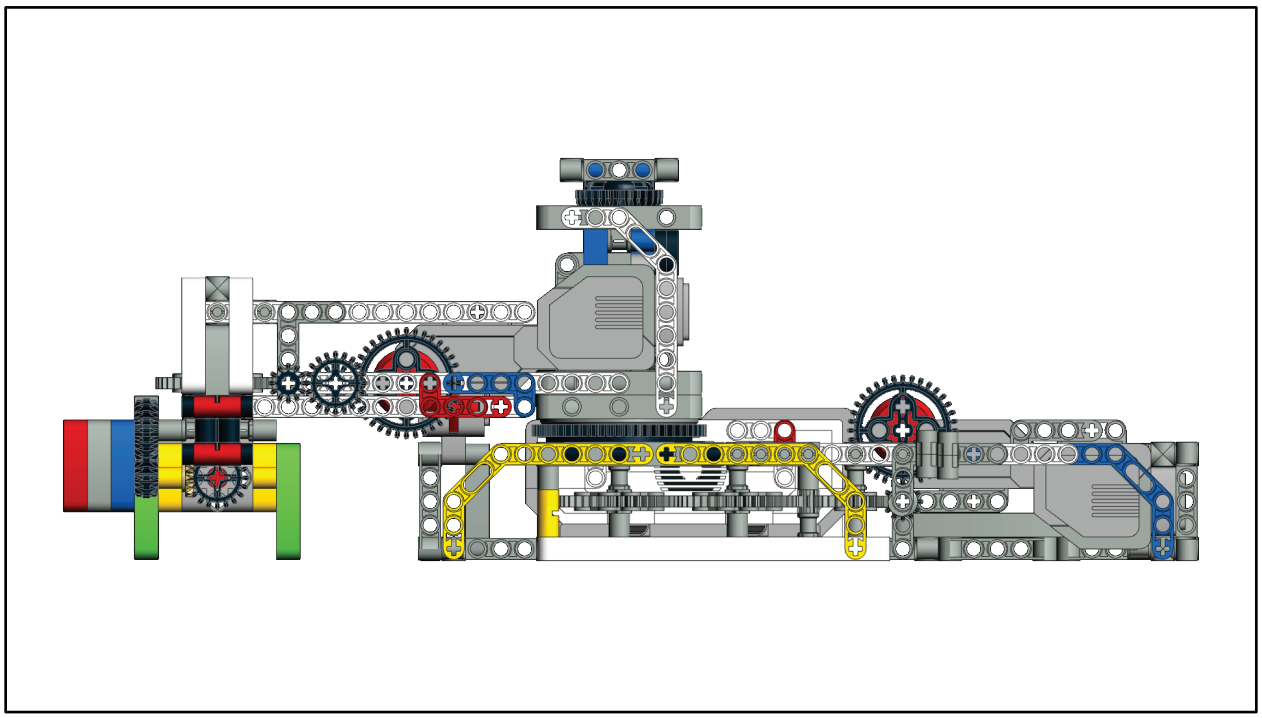

Figure 3.13: Front View

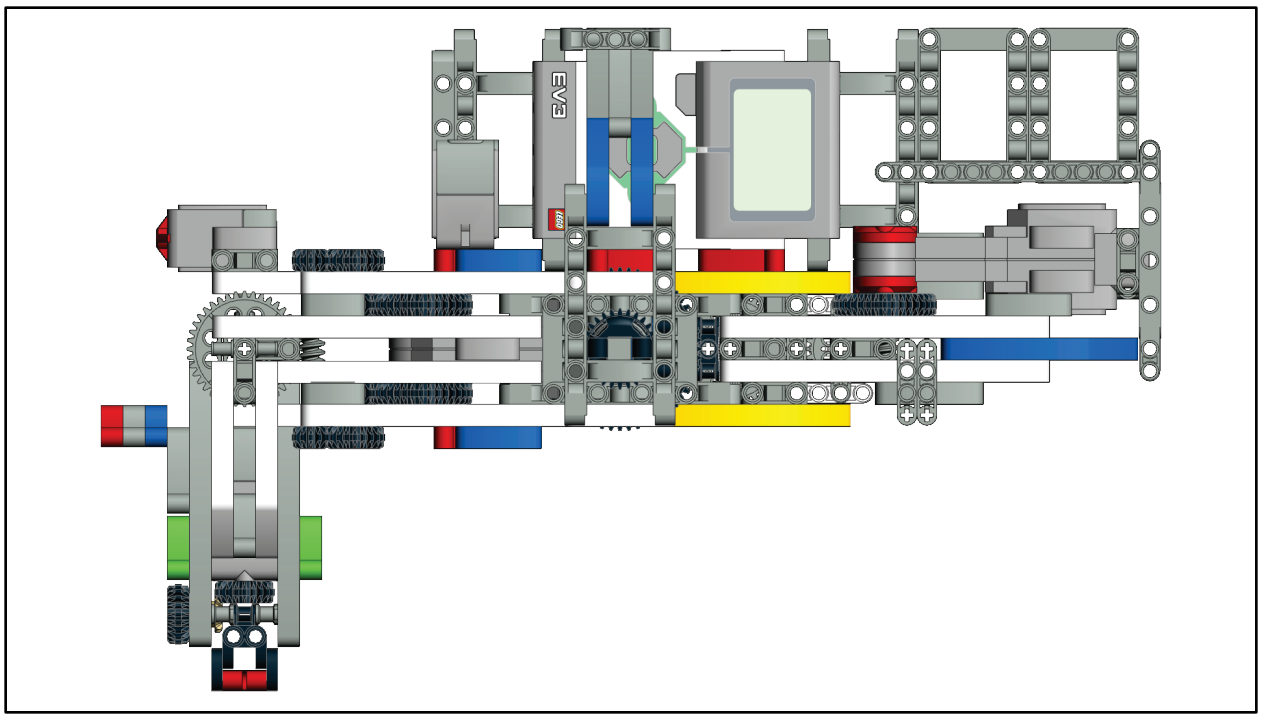

Figure 3.14: Top View

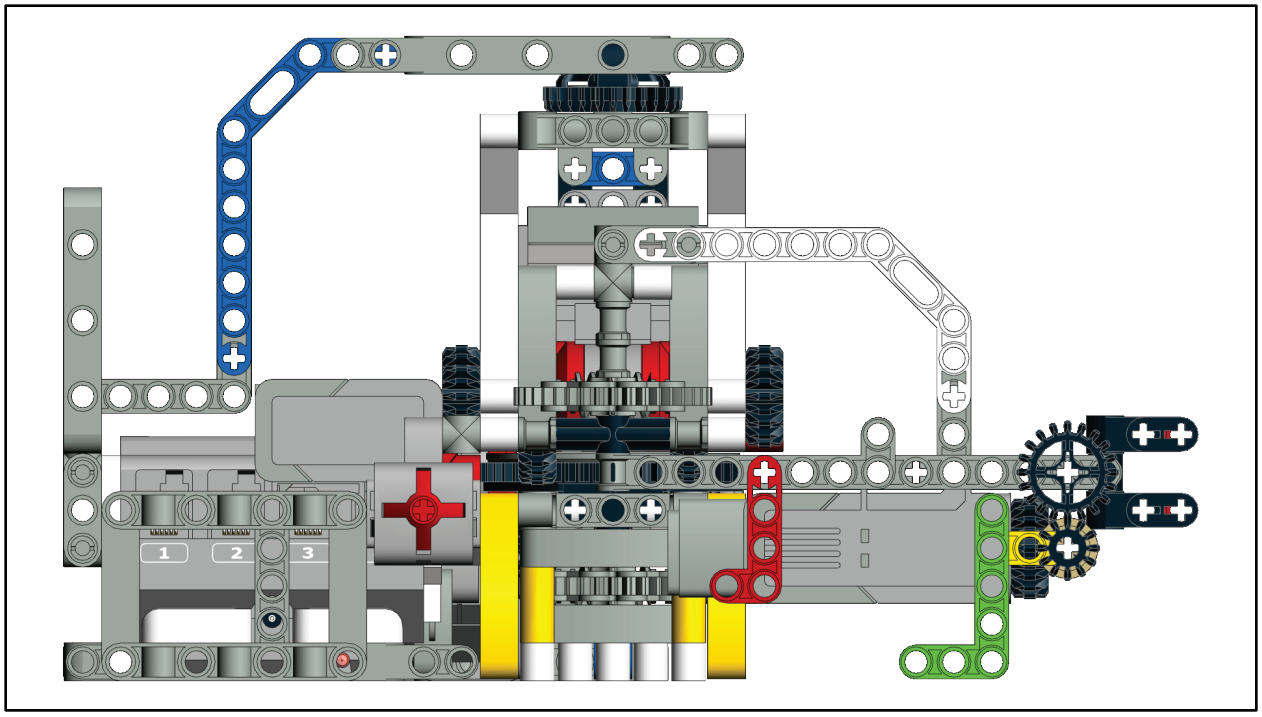

Figure 3.15: Right View

3.2 Package B: Stephanie Chan

This section will show the various steps taken in coming up with the logic for the software.

4 Project Operations

This section answer questions regarding the robot's operation and the results from all the tests when designing and implementing the various parts.

A. Complete the assembly of the robot. Test the basic operation of the system to ensure that the joint/motor relations operate correctly.

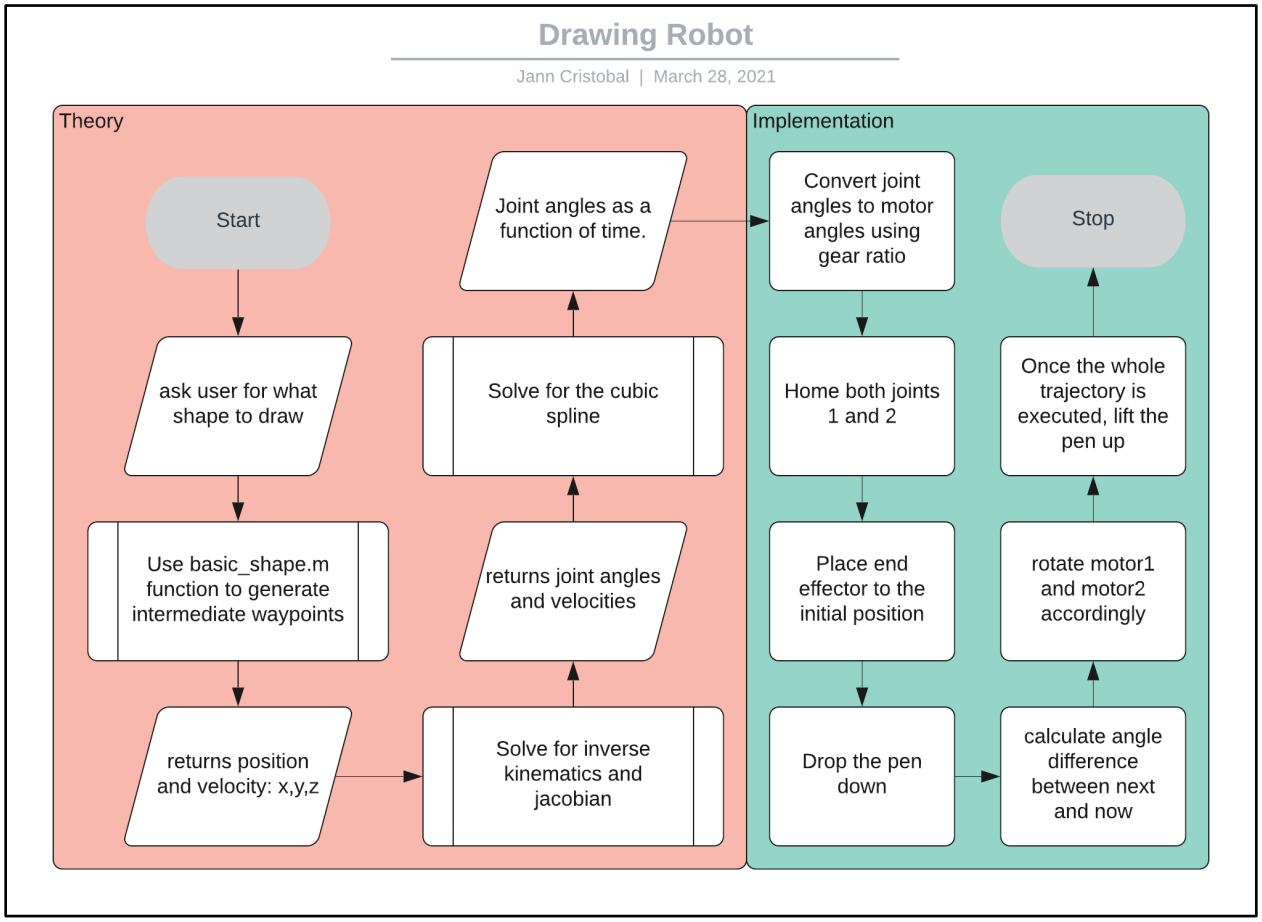

Figure 4.1: Drawing Robot operation flowchart

Once the whole robot was fully assembled and the theoretical values for the joint angles are implemented, the motor is moving in the correct direction of the desired rotation. There are still issues with the accuracy and precision of the drawing. One major problem is with how the links would turn significantly less than what the desired angles when the motor would change the direction of the rotation. A possible solution for this was to add a time delay in order to compensate for the lost angles. But since there is no accurate sensor that measures that actual angles of the links with respect to the joint frames, the precision of the compensator is poor. Overall, the logic for the robot is sufficient in doing

the basic function. But due to limitation with the frequency of the motor readings and external factors, the robot has poor accuracy and precision.

B. *Examine the parameters that affect the trajectory quality of your system. As you increase the number of interpolated trajectory positions, how does this affect manipulator speed and ability to follow a line exactly?*

There were two main parameters that affected the trajectory quality which were the number of waypoints, and motor speed. Increasing the number of waypoints improved the results to a degree, but there was a point where increasing the number of segments just made it difficult for the motor angles to be read. This created a reliability issues where in the middle of drawing, if the desired angle is not detected the system must be manually shut down. Having 10 intermediate waypoints in between two key points proved to be most effective in following the line exactly. The second parameter had significant effect on the trajectory quality was the motor speed. Given the large mass of the links, more torque is required. Even though the robot was designed to have a gear ratio that provide more torque, increasing the power was still required. The drawback with increasing motor speed was that reading its rotation accurately proved to be more difficult. The *readRotation* function would often detect the desired angle 100 degrees off which can translate to between 7.5-12.5degrees in error. Thus, it was necessary to keep the motor speed minimum to preserve reading accuracy, but high enough to provide the required torque to rotate the arm links.

C. *Demonstrate that your system can draw a variety of patterns (e.g., letters, geometric shapes, etc.) Be as creative as you want in testing, but only submit 3-4 examples. Writing on graph paper can scanning the result is probably a good way of documenting this. Comment on the manipulator performance. Are there non-ideal behaviors that manifest in the output (e.g., overshoot, link flexibility backlash, etc?*

Overall, the performance of the manipulator was adequate. It was not to capture the general figure of the shape when compared to the theoretical results. Straight lines are curved substantially. The inaccuracies can be attributed to several external factors such as friction from the supports, and backlash from the gear train. These factors causes the actual link angles to be significantly less than the desired angles. Link flexibility and bending had undesirable consequences. The penholder was not completely rigid and can also be improved. Further improvements with the software and the logic can be performed. The errors can be corrected to a certain degree by doing more tests and adjusting the manipulator behavior accordingly. Other sources of errors can be attributed to the limitation of the sensors themselves. It was not possible to directly command the motor

to rotate a set angle, but rather be determined by continuously reading the motor angles and then compared to the desired value. The indirect process is costly in terms of speed and accuracy. The longer the motor spins without resetting, the more the error accumulates. The corresponding ideal behavior for all the shapes can be referred to in Figure 7.1-7.8 in the appendix. Figure 7.9 in the appendix show the results from the video demonstration. The black line was the horizontal line, the purple was the vertical line, red was triangle, and green was square.

D. *Experiment with Inverse-Jacobian control of the manipulator. This will require an adjustment to the way your trajectories are generated: instead of solving the inverse kinematics for the whole trajectory, you will pick a starting location and then calculate differential joint motions required to create a specified relative motion of the end effector. Demonstrate some simple motions relative to both the end effector and global frames. (hint, if you ask for relative motion in the +x direction, this essentially will draw the x-axis for you)*

Inverse Jacobian solution solves for the joint velocity given the end effector velocities. The first step is to pick a reference point or initial position. At this position, the joints and end effector velocities must be 0 since there is no motion yet. Then, given the velocity at the desired location the joint velocities can then be calculated. To control the trajectory of the end effector, the current joint angles can be inputted and related to the time duration to match the desired joint velocity. To solve for the joint velocities, equations 4.1-4.3 can be used.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J_E \times \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{4.1}$$

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = J_E^{-1} \times \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{4.2}$$

$$J_E = \begin{bmatrix} -L_2 S_1 - L_2 S_{12} & -L_3 S_{12} \\ L_2 C_1 + L_2 C_{12} & L_3 C_{12} \end{bmatrix} \tag{4.3}$$

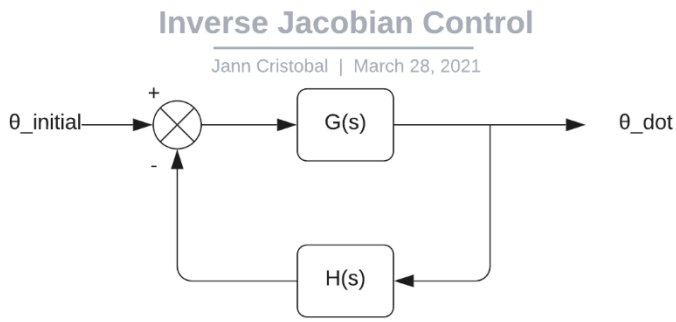

Figure 4.2: Inverse Jacobian Control

Figure 4.2 shows a simple block diagram of how this can be implemented. The open-loop transfer function, $G(s)$ is simply the cubic spline for the joint velocity shown in equation 4.5. The feedback $H(s)$ is simply the cubic spline for the joint angle shown in equation 4.4.

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \tag{4.4}$$

$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2 \tag{4.5}$$

5 Conclusion

Overall, the main goal of this project was achieved although the results can further be improved. Using a two-link manipulator to draw a set of desired shapes was achieved by solving for the inverse kinematics of the system. By knowing the cartesian position for the end effector, the joint angles were calculated. The sequence of angles were then executed in order to produce the desired shape. The actual workspace area was significantly lower in comparison to the ideal case. This was due to the physical limitations of the structure. The width of the link reduced the overall range for both of the links.

6 References

[1] R. Hibbler, Mechanics of Materials, Hoboken: Pearson, 2015.

[2] Wikipedia, "Wikipedia," 30 January 2021. [Online]. Available: https://en.wikipedia.org/wiki/Gear_train. [Accessed 4 February 2021].

[3] M. Brandstotter, Adaptable Serial Manipulator in Modular Design, ResearchGate, 2016.

[4] C. Stahl, "stahlfabrik/TRAC3R," 16 May 2014. [Online]. Available: https://github.com/stahlfabrik/TRAC3R. [Accessed 26 Mar 2021].

[5] J. Enright, "Project 3," Ryerson University, Toronto, 2021.

7 Appendix

7.1 Simulated Results

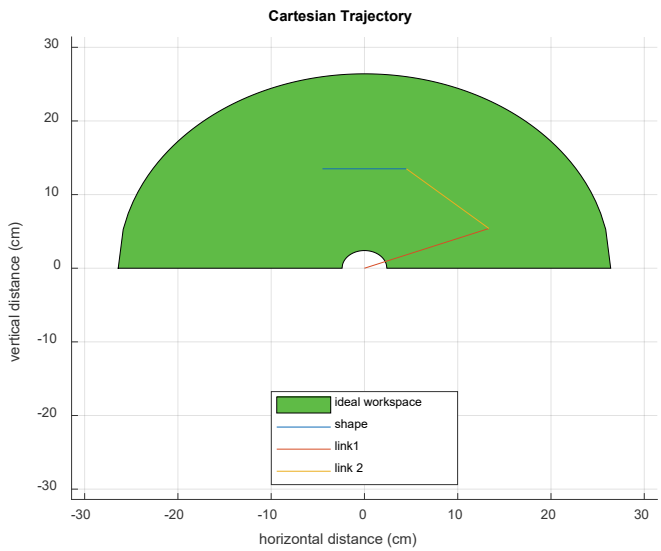

Figure 7.1: Cartesian Trajectory for Horizontal Line

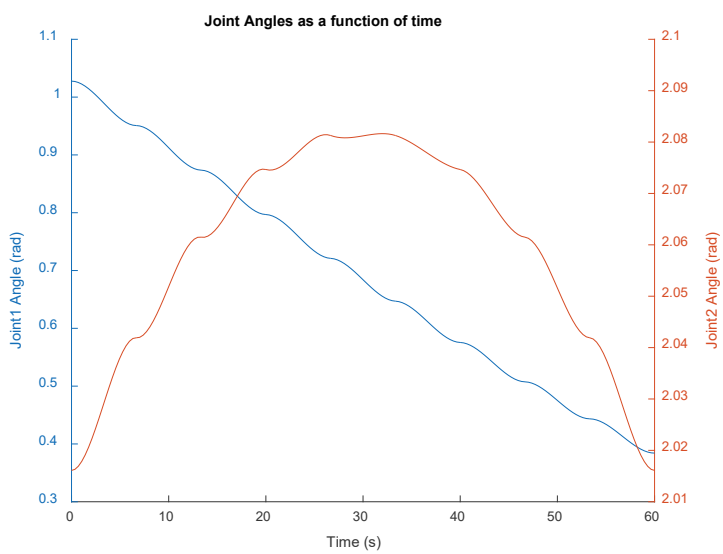

Figure 7.2: Joint Trajectory for Horizontal Line

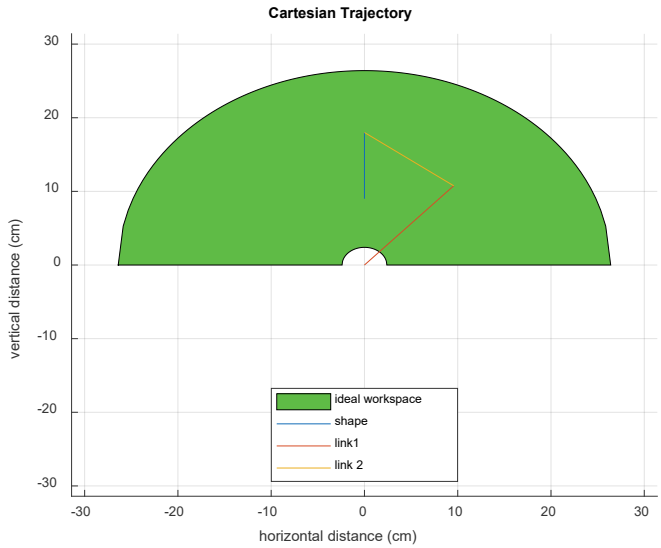

Figure 7.3: Cartesian Trajectory for Vertical Line

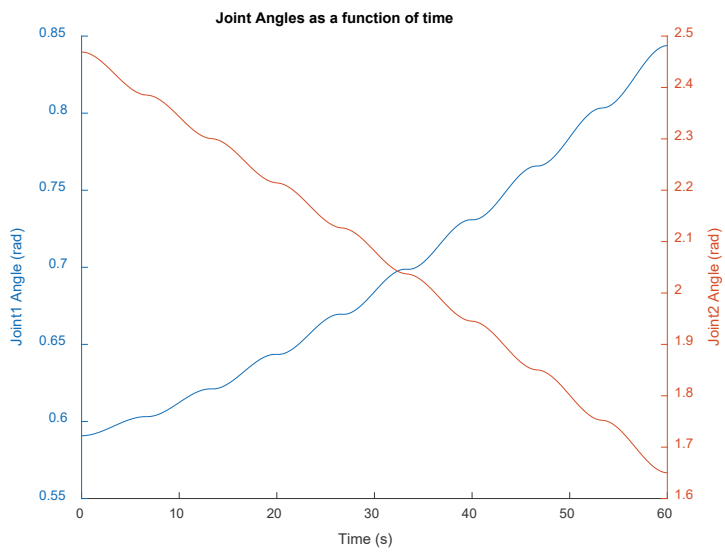

Figure 7.4: Joint Trajectory for Vertical Line

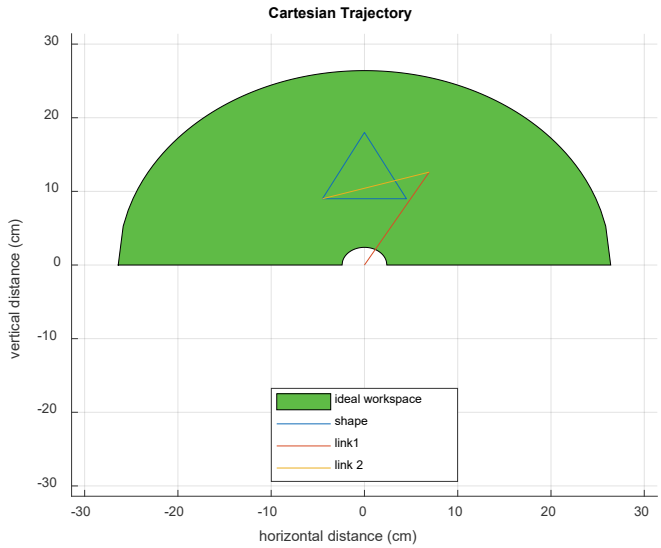

Figure 7.5: Cartesian Trajectory for Triangle

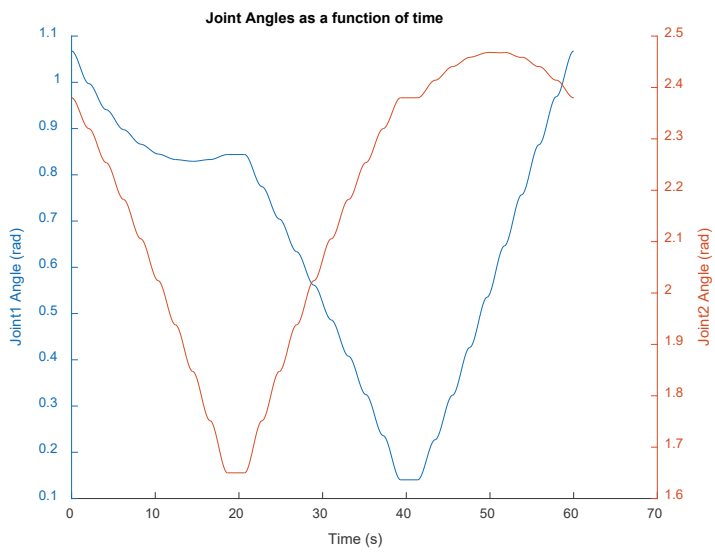

Figure 7.6: Joint Trajectory for Triangle

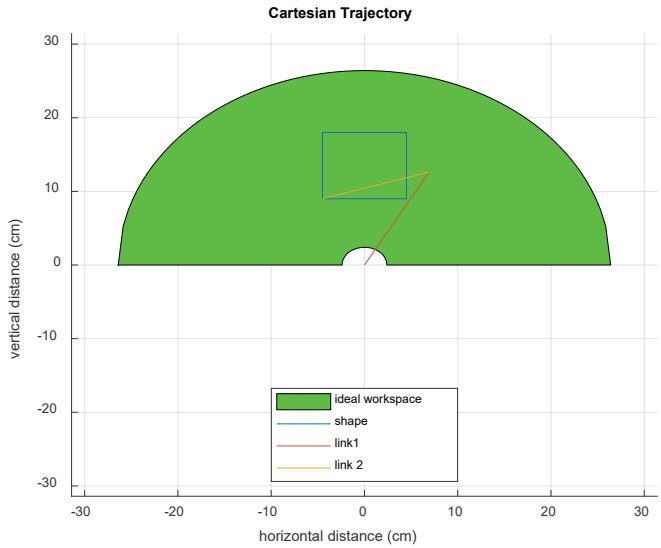

Figure 7.7: Cartesian Trajectory for Square

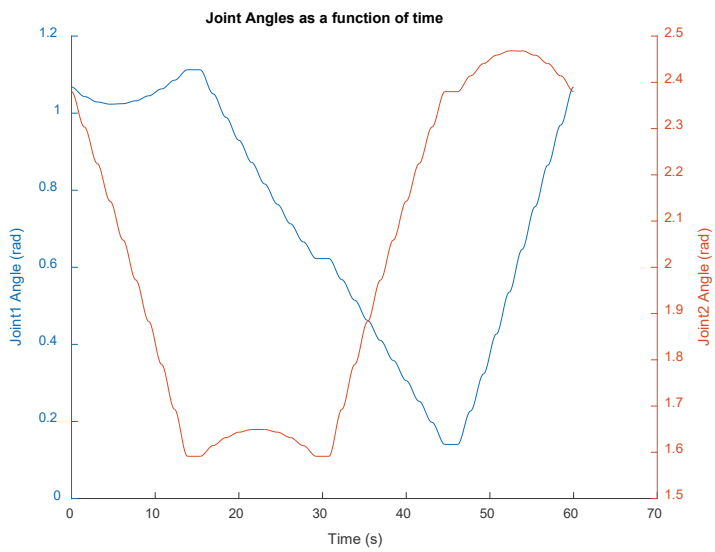

Figure 7.8: Joint Trajectory for Square

7.2 Actual Results

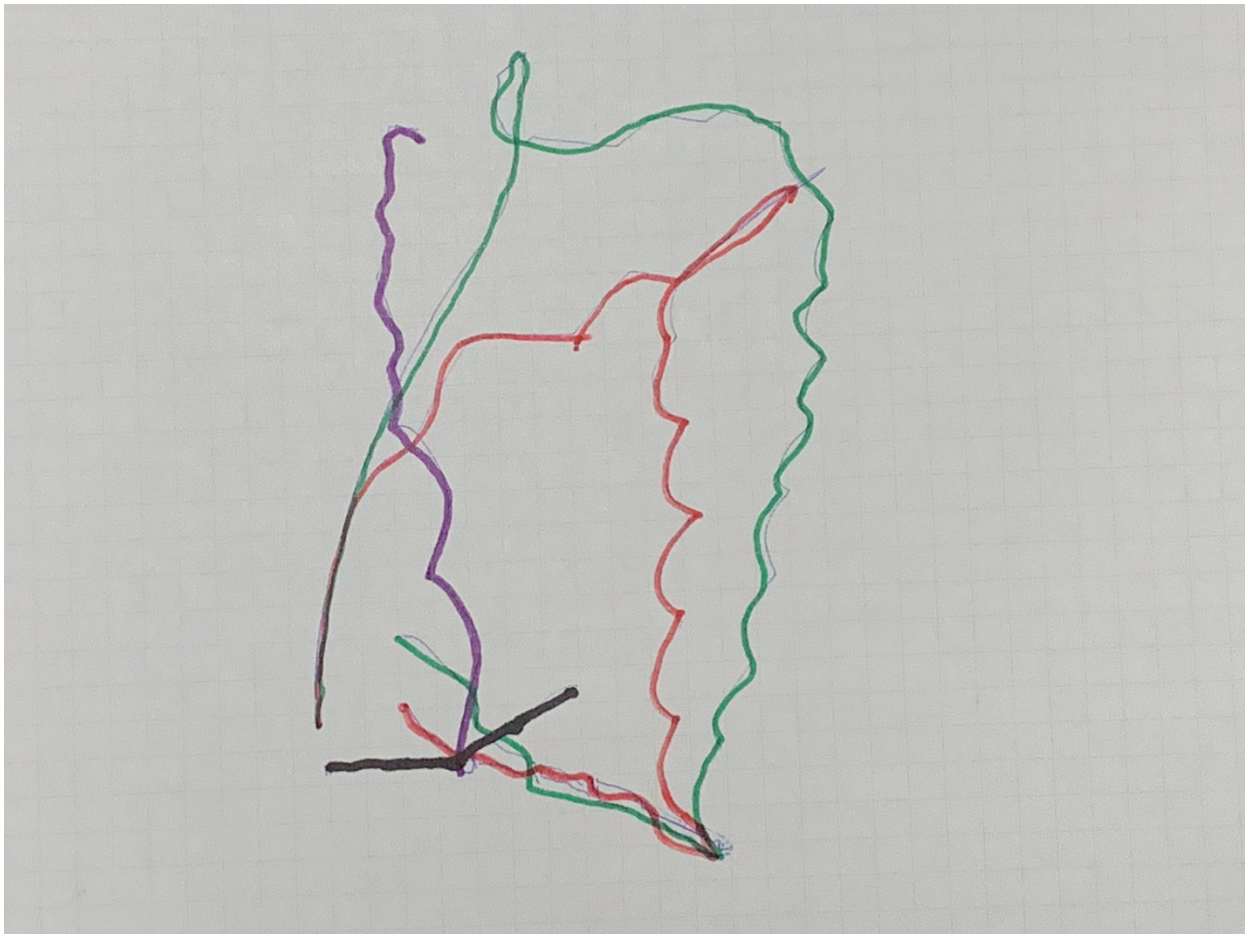

Figure 7.9: Video Demonstration Results