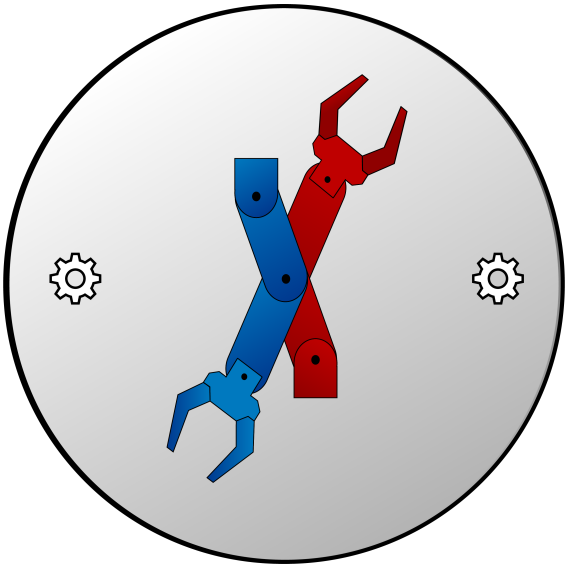
Test Document

Trey Dufrene, Zack Johnson, David Orcutt, Alan Wallingford, Ryan Warner



ME 407

Preliminary Design of Robotic Systems

Embry-Riddle Aeronautical University

EMBRY-RIDDLE AERONAUTICAL UNIVERSITY

Meiosis

Contents

List of Figures

1 Introduction

With the intention of making robotics education more accessible, The Manipulator for Educational Institutions with Open Source Integrated Systems (MEIOSIS) intended to provide robotics classes with an accurate and precise manipulator for a lower cost. The system is designed to be 3D printed by the end-user to reduce overall cost and will be modifiable to create a sustainably increased understanding of robotics. This document details the tests performed by the MEIOSIS team and the specific results obtained by each test. This is done in order to ensure that the design requirements are properly met by the finished product.

2 Specifications and Tests

Specification **1.1.a** states: **The cost for the MEIOSIS team to develop the manipulator shall cost no more than $800.** This is met if the cost for the MEIOSIS team to develop the manipulator does not exceed $800. This can be tested by looking at the parts list provided in the parts list seen in Table 1.

Table 1: MEIOSIS Bill of Materials with Costs

money	money

As can be seen in Table 1, the manipulator will cost XX and the manipulator meets specification 1.1.a.

Specification **1.2.a** states: **The system shall consist of six rotational joints connected by four links. The last three joints will create a spherical wrist.** For this specification to be met, the robot must have six rotational joints and 4 links. The last three joints must create a spherical wrist where the axis of revolution for the last three joints are intersecting with one another. To perform a test to check for six rotational joints, SolidWorks is required. A render of the manipulator with labeled joints can be seen in Figure 1.

As can be seen labeled in Figure 1, the robot consists of six rotational joints. Joint 1 controls the rotation about the base, joints 2-3 control the "shoulder" and "elbow" of the robot, and joints 4-6 control the spherical wrist. The rotational axis of joint 4 intersects with joint 5's and joint 5's also intersects with joint 6's rotational axis. To perform this test, the assembly file containing the manipulator was opened and the joints were counted. Using the move tool, each joint can be observed to be purely rotational. The data obtained is purely simulated due to the fact that the robot has not been fully fabricated. **This manipulator meets specification 1.2.a**.

Specification **1.2.b** states: **The system shall have no link offsets.** For this specification to be satisfied, there must be no link offsets present in the design of the robotic manipulator. This means that the link should connect two joints by translating in only one

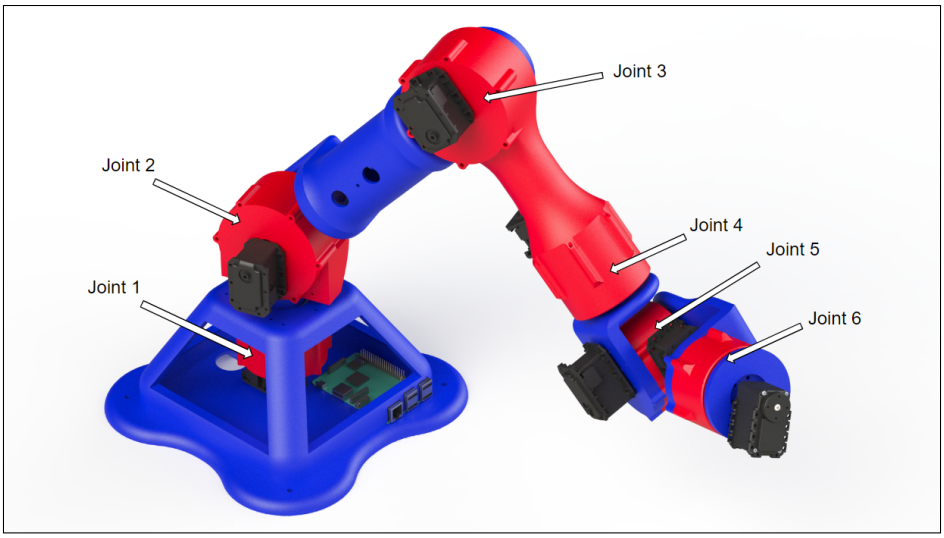


Figure 1: Labeled Image of Manipulator Render

direction and the axes of the joints attached to the link should be collinear with one another. Testing this specification requires SolidWorks' measurement tool. Using the measurement tool, measure each joint to ensure that there is only displacement along one axis. Figure 2 shows an example of a measurement done on the elbow link of the manipulator.

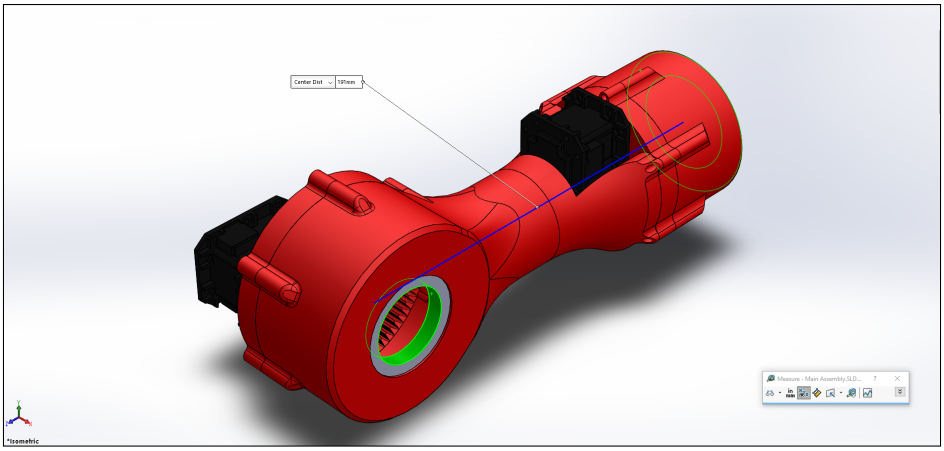


Figure 2: Measurement Test for Link Offset

Figure 2 shows a screen capture of a measurement taken and as can be seen by the blue line, the link only extends across one axis. If there were a link offset present, there would be multiple multi-colored lines to show displacement in multiple directions. These tests were done in simulation on each link of the manipulator and none contained any offset direction. Through this test, it is observed that **the manipulator meets specification 1.2.b**.

Specification **1.3.a** states: **The system shall accommodate a process in which the end user can calibrate the end effector position and orientation to within 0.5**

mm and 1 degree of the manipulator's precision. In order to fulfill specification 1.3a, an algorithm was developed for calibration. This algorithm must allow the user to manually move the links into a desired zeroed configuration, then accurately retain that zeroed configuration as a reference position during use. Figure 3 depicts the manipulator in a zeroed configuration.

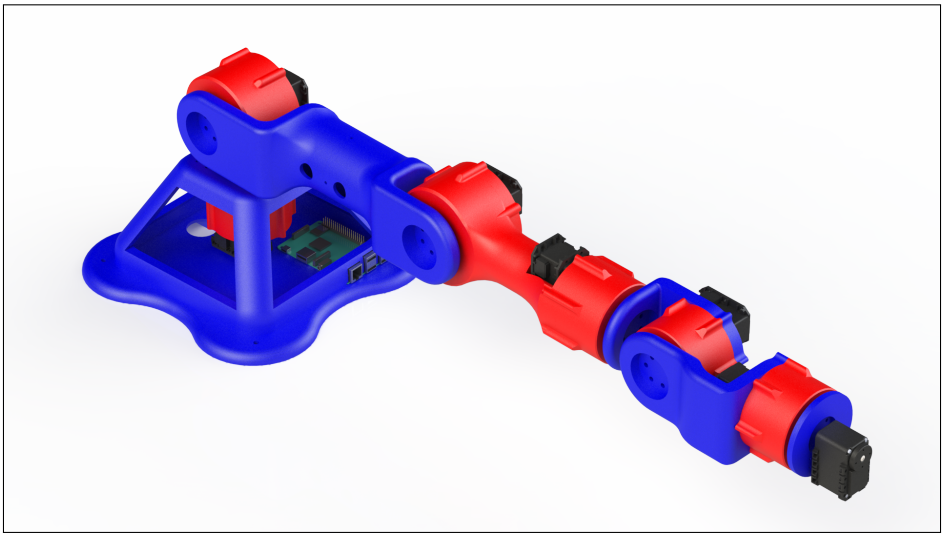


Figure 3: Manipulator in its Zeroed Configuration

The configuration shown in Figure 3 is what we have defined to be the zeroed configuration. The purpose of the calibration algorithm is to force the software to consider every joint angle to be zero when the manipulator is in this configuration, even if the servos return non-zero angular values. The algorithm developed allows the users to move the joints one by one until the robot is in the desired "zeroed" position. The program then reads the servos positions and stores those values as offset variables. These offsets are subtracted off of the desired joint angle values when commanding the servo positions.

In order to measure the calibration accuracy, the PhaseSpace Motion Capture system will be required. The calibration algorithm itself is accurate to the same value as the manipulator, however inaccuracies can arise due to the user not being able to manually put the manipulator into the correct configuration. The testing procedure will be performed as follows. The motion capture system will be fully set up, including placing the tracking markers on the manipulator's base and end-effector. Once the motion capture system is fully set up, the manipulator will be placed in an arbitrary configuration, at least 30 degrees away from zero in each joint. The manipulator will then be restarted, which will erase the system's memory of it's configuration. At this point, the motion capture system will start recording data. Using the calibration software, the manipulator will then be manually zeroed to the best of the operator's abilities. Once this is completed, error terms can be determined by comparing the tracking data to the known values of the end-effector's position and orientation in the zeroed state. Due to circumstances beyond our control, we are not able to perform this test. Because of this, adherence to specification 1.3a, and by extension **adherence to specification 1.3.a cannot be determined.**

Specification **1.4.a** states: **Joint one and two of the system shall possess an angle error of no more than 0.025 degrees.** Specification **1.4.b** states: **Joint three of the system shall possess an angle error of no more than .03 degrees.** Specification **1.4.c** states: **Joints four, five, and six shall possess an angle error of no more than .29 degrees.** In order to verify specifications 1.4.a-1.4.c, the angle accuracy of the system must be measured. To determine the accuracy, the PhaseSpace Motion Capture System is required. Three LED indicators will be placed at the end of each gearbox that is farthest from the previous gearbox in the line to indicate the farthest point directly controlled by the previous gearbox. The three LEDs will be placed at 120 degree increments around each link/gearbox so that one or more of the LEDs are visible to the motion capture system at a time. To attain accurate results, small divots will be placed on the manipulator where the LEDs will be located in order to allow precise application of the LEDs to the system. Accurate LED placement will allow for more accurate kinematic/angle calculations.

To test the system, the manipulator will be placed in the zeroed configuration and each joint will be tested individually. First, joint 1 will be tested by running from zero to 90 degrees, and then back to zero, and repeat for a total of 10 times. Each time joint 1 has reached the desired angle, the motion capture system will collect the location data of the LEDs and store it for error calculation. When the test has completed, the data can be used to calculate the angle the link actually achieved, and can be compared to the desired angle. Any error in zeroing can also be seen from the collected data. After the joint 1 has been tested, joint 2 will be tested following the same procedure, after which joints 3 through 6 will be tested in order. The worst angle error achieved by each joint will be compared to the value defined in the specifications. This test is unable to be performed and thus **the manipulator's adherence to specifications 1.4a-1.4c cannot be determined.**

Specification **1.5.a** states: **The manipulator shall have a maximum reach between 300 and 700 mm.** In order to satisfy this specification, the robot's links must have a sum length of between 300 mm and 700 mm. Using Solidworks, the total reachable workspace can be measured using the "Measure" tool. This measurement can be seen in Figure 4.

As can be seen in Figure 4, the total reachable workspace is 643 mm. This measurement is between 300 and 700 millimeters and therefore **the manipulator meets specification 1.5a.**

Specification **1.5.b** states: **The length of link one, two, three, and the wrist shall be 161.5 mm, 250 mm, 250 mm, and 143 mm respectively.** For the manipulator to meet specification 1.5.a, each link and the wrist simply need to match the lengths provided in the specification. Testing this specification requires the SolidWorks measurement tool. An example of the measurement tool being used to test this specification can be seen in Figure 5.

Once the measurement tool in SolidWorks is selected, a measurement from the first joint in a link to the last joint in a link should be taken. Figure 5 shows a measurement between joints 3 and 5, these joints make up Link 3 or the "elbow" link of the manipulator. The

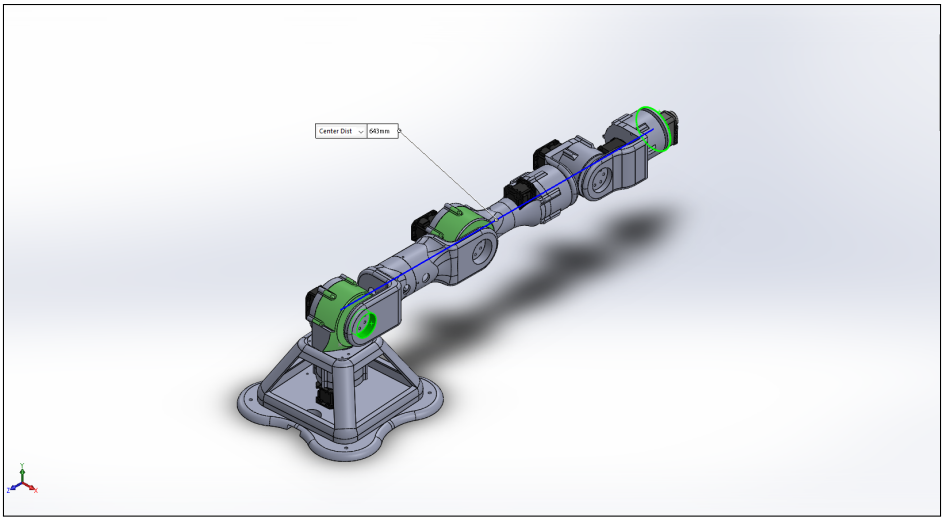


Figure 4: Solidworks Measurement of Reachable Workspace

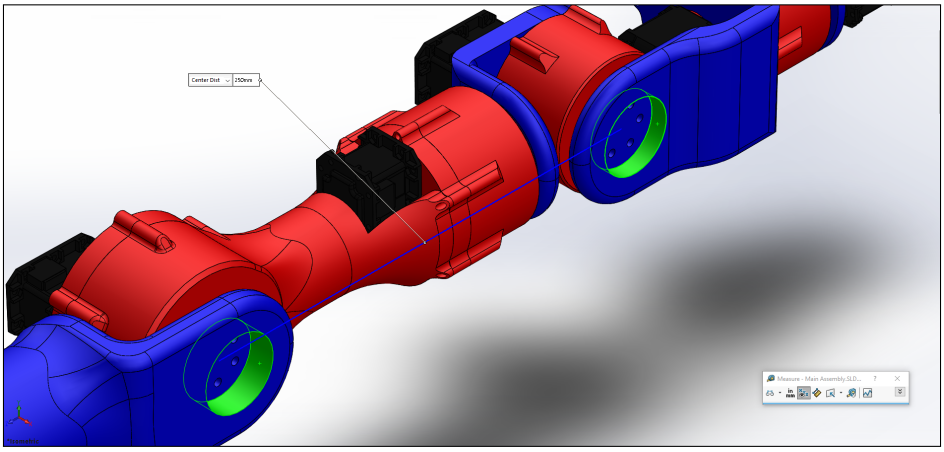


Figure 5: Test of Link Length for Specification 1.5.a.

center distance measurement came out to be 250 mm. Once a measurement is done for each link, the center distance values are to be recorded. The variable factors in these measurements are the dimensions of the parts used to make up the link lengths. The lengths of the individual links can be seen in Table 2. These dimensions are simulated in SolidWorks.

Table 2: Measurements of Manipulator Link Lengths

Portion of manipulator measured	Measurement Value
Link 1 (Base to Joint 2)	161.5 mm
Link 2 (Joint 2 to Joint 3)	250 mm
Link 3 (Joint 3 to Joint 5)	250 mm
Wrist (Joint 5 to End Effector)	143 mm

As can be seen by Table 2, the link measurements obtained in SolidWorks match the measurements provided by specification 1.5.a and the **manipulator meets the specification.**

Specification **1.6.a** states: **The system's workspace must contain a hemispherical shell which has a thickness of 280 mm where the robot is theoretically fully dexterous.** In order to test the dexterity of the manipulator and ensure that it fulfills specification 1.6.a, a program was created which tests a series of points for dexterity. These points are 1 millimeter apart to match the manipulator's precision and can be rotated around the z-axis to create a hemispherical shell. It does this by calculating the inverse kinematics for the manipulator, given its dimensions, and if any point is outside of the manipulator's workspace it is marked on a MATLAB plot, as can be seen in Figure 6.

Figure 6 shows a failed configuration for our dexterity check. After around 35 cm away from the robot it is no longer fully dexterous in this configuration. In order to fix this, changes to the design need to be made, specifically the length of the links. Another fix would be moving the dexterous workspace closer to the base of the robot, however it cannot be moved too close since the base of the robot would be in the way. Figure 7 shows a successful dexterity check.

As can be seen in Figure 7, a successful dexterity check was achieved and the requirement is met. The dexterous workspace spans from 70 mm away from the base to 350 mm away from the base and thus creates a hemispherical workspace with a thickness of 280 mm, thus **the manipulator fulfills specification 1.6.a.**

Specification **1.6.b** states: **The rotational limit of joint one, two, three, four, five, and six shall be -165° to 180°, 13.7° to 179.8°, -179.6° to -19.2°, ±180°, -180° to 0°, and ±180° respectively.** This specification is met if the links can rotate to the maximum and minimum angles provided by the specification. These angles were tested in SolidWorks using the measurement tool. The links to be measured are moved until they collide with one another and then the measurement tool is selected. Once the measurement

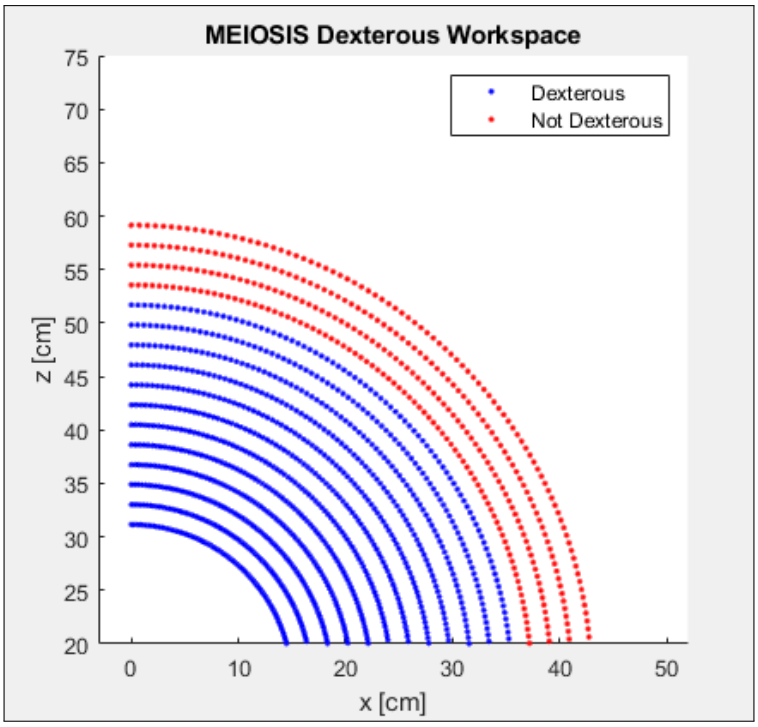


Figure 6: Matlab Plot of Invalid Dexterity Check

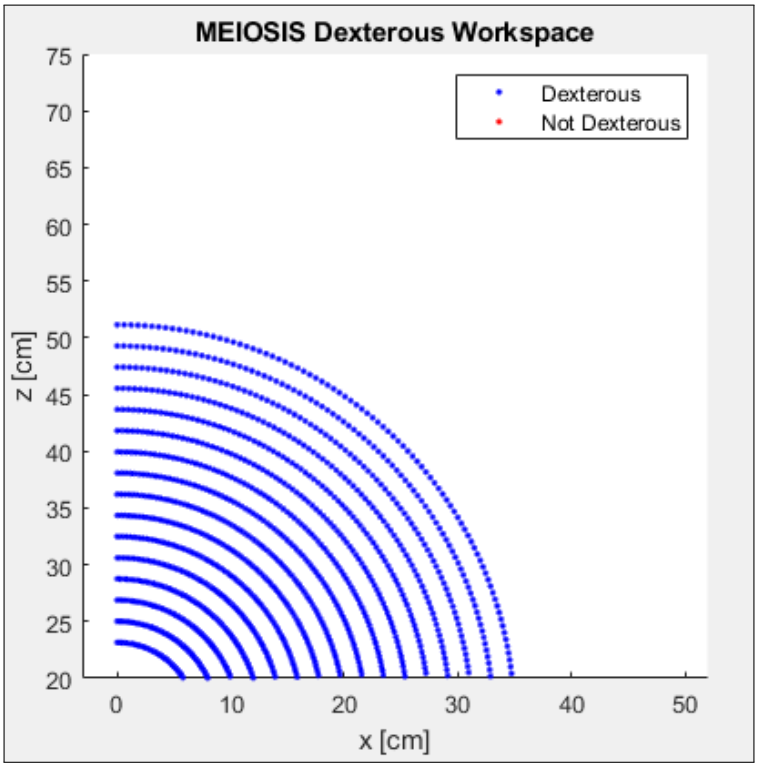


Figure 7: Matlab Plot of Successful Dexterity Check

tool is selected, the two links that form the angled are selected and the "create sensor" button is pressed. SolidWorks then provides the angle in degrees on the left sidebar. The angles for each joint are then recorded and the results of the measurements can be seen in Table 3.

Table 3: Results of Angle Measurement Test for Specification 2.1.6.a

Joint Number	Minimum Angle	Maximum Angle	Specification Met?
1	-180°	180°	✓
2	-21°	201°	✓
3	-126°	126°	✗
4	-180°	180°	✓
5	-75°	115°	✗
6	-180°	180°	✓

Table 3 shows that the measurements for joints 3 and 5 do not meet the specification's individual requirements for them, therefore **the manipulator does not meet specification 1.6.b.**

Specification **1.7a** states: **The system shall use a parallel gripper that can close to 18 mm.** To meet specification 1.7.a, a parallel gripper must be designed or purchased that can close to a minimum distance of at most 18 mm. Dynamixel offers a parallel gripper actuated by an AX-12A servo as shown in Figure 8. This can be utilized since the manipulator's end effector is designed to attach an AX-12A smart servo.

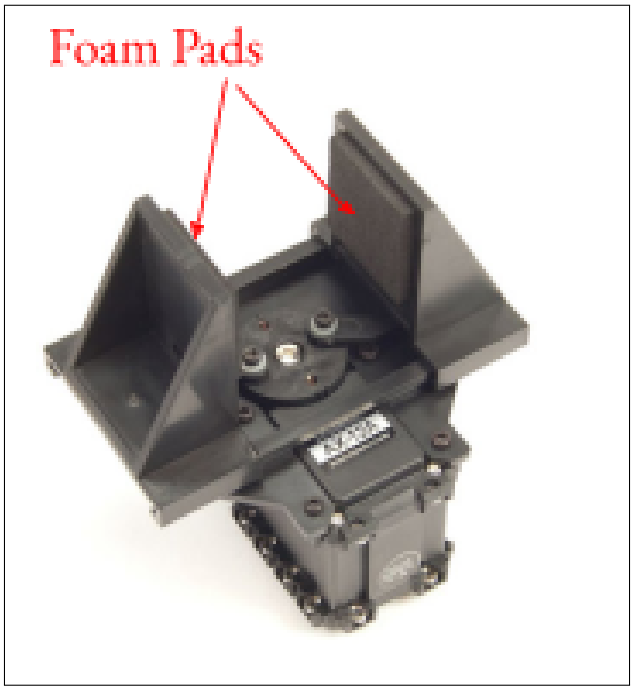


Figure 8: Dynamixel Parallel Gripper with Servo Modified from [2]

As depicted in Figure 8, the gripper is open to its widest extent. However, the gripper's fingers have removable foam pads. The foam pads allow adjustments in the gripping plasticity and the maximum opening/closure. Figure 9 shows the gripper in various configurations.

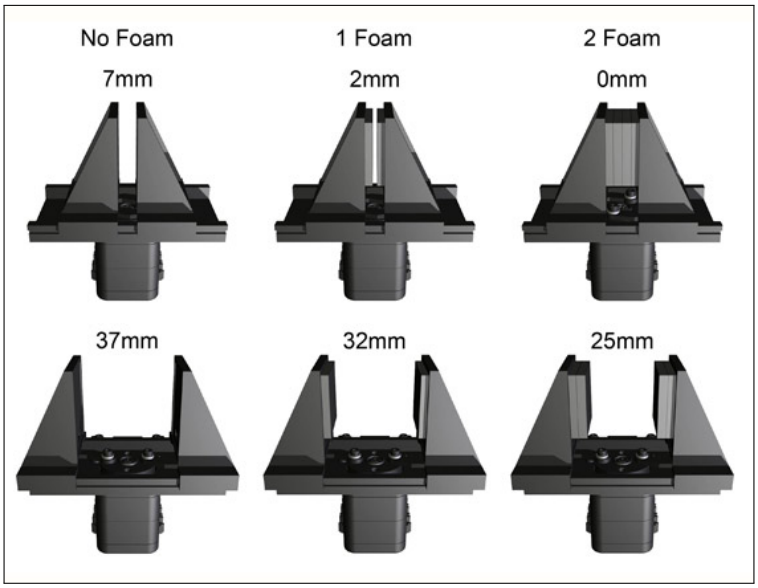


Figure 9: Dynamixel Gripper with and without Foam Padding at Minimum and Maximum Opening [2]

The top three gripper configurations, shown in Figure 9, are maximally closed and the bottom three are maximally open. From left to right, the fingers have no, one, and two pieces of foam. The maximum opening is 32 mm and the minimum amount of closure is 0mm. However, these extremes require different numbers of foam pads. So, the gripper could not open to 32 mm, then, close to 0 mm solely by the servo's actuation. While none of these configurations has 18 mm between the fingers and, consequently, cannot close to 18 mm, the distance between the fingers can be adjusted as shown in Figure 10.

By sending the servo a position, the two highlighted red half-circular bars seen in Figure 8 pivot about the servo face's edges, translating the fingers either inward or outward. Clockwise rotation would decrease the distance between the fingers, while counterclockwise rotation would increase the distance. The closure's precision would be dictated by the AX-12A's resolution. Since the precision of the 18mm closing distance is not specified, it can be assumed the precion need only be sufficient to prevent the marker from being dislodged during writing, which is addressed in 1.8a. Therefore, with any number of foam pads, the gripper could close to 18 mm. With this gripper being used as an end effector attachment, **the manipulator meets specification 1.7.a.**

Specification **2.1.a** states: **The software shall be hosted publicly on an online repository and maintain an MIT license for distribution.** To satisfy this requirement, the software used to control the robot must be open source and publicly

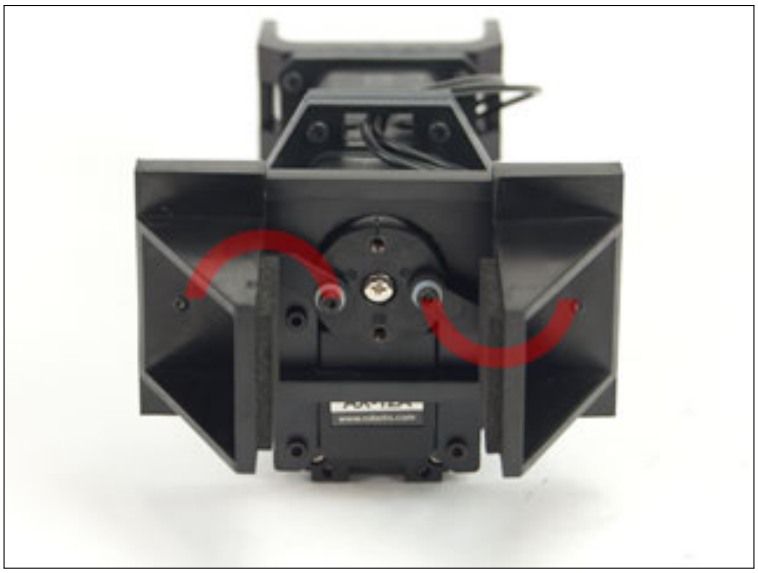


Figure 10: Dynamixel Gripper Rotational to Linear Motion Mechanism [3]

accessible; the use of the code repository GitHub allows for the former easily; by using the MIT licensing, the latter is accomplished, and possible legal implications due to misuse can be avoided. The license allows the end-user to modify, redistribute, or do with the code basically however they please: "...without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software..."[4] This makes it such that any end-user that makes a system with the manipulator, i.e. repurposes the given system for a task, has no limitations with how they wish to redistribute, share, or even sell the system. To implement the use of the license, all that must be done is the source code must contain the proper header including, shown by MIT [4], which was completed, as well as the code is hosted in a public GitHub repository. Therefore **the system fulfills specification 2.1.a.**

Specification **2.2.a** states: **The system shall have a user interface capable of accepting the end-effector's desired cartesian position and Euler angle orientation as a six element row vector.** In order to satisfy specification 2.2.a, python scripts have been methodically developed to allow for user input of desired end effector position. These scripts could potentially be used as a backend for a graphical user interface (GUI) to further simplify the process of controlling the manipulator.

In lieu of a python based GUI directly controlling the manipulator, a GUI could be created to instead control an animated simulation of the manipulator. The simulation is currently generated by a MATLAB script which draws the robot STL files into a figure window—the possibilities of controlling the simulation consist of several options, the most efficient would likely be to develop a user interface entirely with MATLAB. This avoids possible incompatibilities on systems without Python installed and avoids any communication issues that could incur with attempting to create the GUI in another language.

The best option is to have a MATLAB GUI control the MATLAB simulation, therefore a basic interface shall be created in which the user can select which control scheme is desired as well as goal positions for the end effector. A sample GUI can be seen in Figure 11.

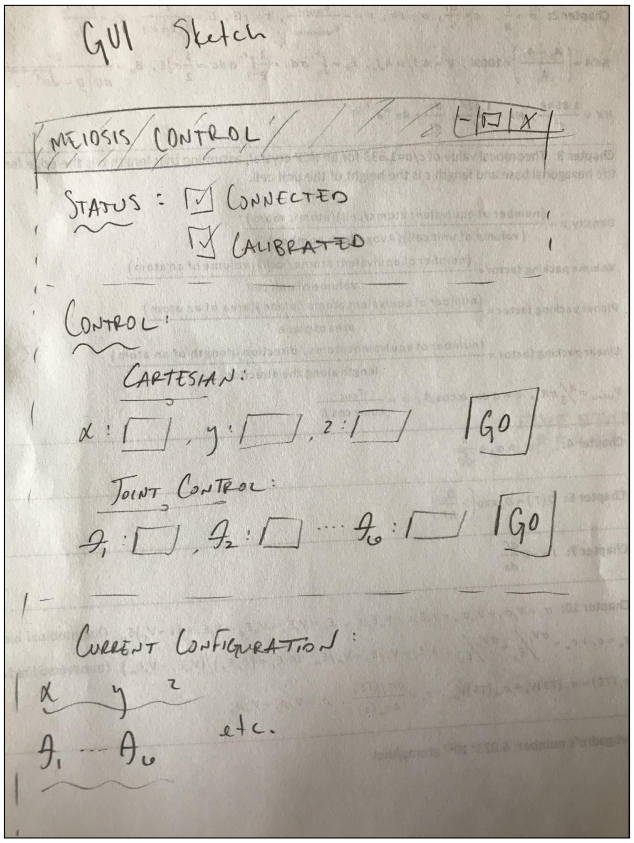


Figure 11: GUI Template

As shown in Figure 11, the GUI will have desired cartesian end effector positions, such as an X, Y, and Z for the end effector position, as well as an end effector orientation input, i.e., phi, theta, and psi. The input will closely resemble the input of a 6 element row vector. The interface will also have user selectable direct joint controls, rather than the system performing the inverse kinematic calculations for the user. The interface will also display status indicators showing whether the robot is calibrated as well as the current manipulator configuration.

In order to test this specification, several points within the workspace will be selected as well as at least 3 points outside of the workspace for the purpose of error checking. The points (in cartesian XYZ and Euler angles) are chosen randomly. These points are then given to the user interface, which will return joint angle values. The same randomly generated coordinates will be given to the verified kinematics functions, and the outputs will be compared to the output of the UI to check for validity.

Due to uncertainties in the design caused by recent events, development for this user

interface has been stopped and verification of its accuracy can not be determined. Therefore, **adherence to specification 2.2.a cannot be determined.**

Specification **2.2.b** states: **The system shall be capable of performing floating point arithmetic.** In order to fulfill specification 2.2.b, a computational system must be selected that can perform floating point arithmetic. The Raspberry Pi 3b+ includes an ARM Cortex A-53 [5], which is capable of performing high-speed floating point arithmetic [1]. Based on this fact, **the manipulator meets specification 2.2b.**

3 Conclusion

Table 4 summarizes the results of the tests performed throughout the course of this semester.

Table 4: Summary of Test Results

Specification Tested	Specification Met?
1.1.a	
1.2.a	✓
1.2.b	✓
1.3.a	Cannot be Determined
1.4.a	Cannot be Determined
1.4.b	Cannot be Determined
1.4.c	Cannot be Determined
1.5.a	✓
1.5.b	✓
1.6.a	✓
1.6.b	✗
1.7.a	✓
1.7.b	Cannot be Determined
1.8.a	Cannot be Determined
2.1.a	
2.2.a	
2.2.b	✓

Appendix