

Preliminary Design Report

Trey Dufrene Zack Johnson David Orcutt Alan Wallingford Ryan Warner

Submitted in Partial Fulfillment of the Requirements of:

ME407 Preliminary Design – Fall 2019



Meiosis

College of Engineering
Embry-Riddle Aeronautical University
Prescott, AZ

Contents

1	Introduction	1
2	Requirements	1
3	Conceptual Design	3
4	Specifications	8
5	Preliminary Design	13
5.1	Computer Aided Design	13
5.2	Actuator Analysis	15
5.3	Forward Kinematics	17
5.4	Velocity Kinematics	19
5.5	Inverse Kinematics	19
5.6	Equations of Motion	20
5.7	Open-Loop Simulation	22
5.8	Closed-Loop Simulation	24
5.9	ANSYS	27
5.10	Electrical Schematic	28
5.11	Software Flowchart	29
5.12	Project Status and Future	33
5.13	Parts List	35

List of Figures

1	Overall System Conceptual Design	3
2	Manipulator Base with Call-outs	4
3	Drawing Showing Key Features of Design	5
4	Drawing Showing Link Cross Section	5
5	Electrical System Block Diagram	6
6	Software Flowchart	7
7	Overview of Physical System	8
8	Elbow Manipulator Configuration with Link Offset	9
9	Kinematic Model Representing Zeroed Configuration	11
10	Manipulator in Zeroed Configuration with Callouts	13
11	Link Cross Section	14
12	Pulley 1-2 Center Distance	15
13	Pulley 3 Center Distance	15
14	Coordinate Systems	17
15	Open-Loop Control Simulation Animation Snapshots	23
16	Joint Angles vs Time in Open-Loop Simulation	23
17	Joint Angles vs Time in Open-Loop Simulation	24
18	Joint Angles vs Time in Open-Loop Simulation	24
19	Closed-Loop Control Simulation Animation Snapshots	26
20	Joint Angles vs Time in Closed-Loop Simulation	26
21	Joint Angles vs Time in Closed-Loop Simulation	26
22	Joint Angles vs Time in Closed-Loop Simulation	27
23	T-Bar ANSYS FEA	27
24	Electrical Schematic	28
25	General Software Algorithm Overview	29
26	Software Flowchart Subsections	29
32	Cross Section of Dexterous Workspace Quadrant	38

List of Tables

1	Manipulator Link 1 Mass Tabulations	16
2	Manipulator Link 2 Mass Tabulations	17
3	Bill of Materials with Costs	35

List Of Acronyms and Abbreviations

FK : Forward Kinematics

IK : Inverse Kinematics

PD : Proportional Derivative

Notation

$r_{\text{From Frame To}}$: Direction Vectors

$T_{\text{From To}}$: Direction Cosine (Transformation) Matrices

$c_{\theta_{nm}}$: $\cos(\theta_n + \theta_m)$

$s_{\theta_{nm}}$: $\sin(\theta_n + \theta_m)$

1 Introduction

The Manipulator for Educational Institutions with Open Source Integrated Systems (MEIOSIS) aims to increase the accessibility of robotics education to high school students and hobbyists. Contained within this document is the design of a manipulator to fulfil this end. It will address the manipulator's conceptual design with requirements defining specification therefrom. A detailed design with motor descriptions introduces the manipulator's kinematics and simulations, concluding with a description of the system's operation and plans for implementation.

2 Requirements

The requirements of the system concisely define the capabilities the system must possess in order to solve the stated problem.

2.1 Hardware

The following requirements are hardware specific and dictate the physical constraints the system must adhere to.

2.1.1 The system shall cost the end-user no more than \$1000.

2.1.2 The system shall be fully dexterous without being kinematically redundant.

To create a system with the intention of advancing education, it must be complex enough to encourage higher level problem solving, as well as be capable enough (dexterous) in a broad spectrum of tasks — in the interest of remaining useful in addition to retaining the interest of students.

2.1.3 The system end effector shall maintain a positional accuracy magnitude of ± 1 mm and an orientation accuracy of $\pm 5^\circ$ eigen angle from the base frame.

To ensure that the robot has educational value, the accuracy must be defined so that any desired positions and movements are achieved.

2.1.4 The system end effector shall maintain a pose repeatability magnitude between 0.1—1.5 mm for the position and $\pm 4^\circ$ eigen angle from the base frame for the orientation.

This is to ensure a robot that can execute the same movement commands repeatedly and have the same results every time.

2.1.5 The system's reachable workspace shall be a hemisphere with a radius of 300-700 mm.

This workspace will provide enough movement to manipulate objects in order to perform basic tasks.

2.1.6 The system's dexterous workspace shall contain a hemispherical shell within the reachable workspace with a thickness of 280 mm.

2.1.7 The system shall have a removable end effector capable of picking and placing a low-odor chisel tip Expo dry erase marker.

This creates a robot capable of performing a variety of basic tasks, which enhances its educational value.

2.1.8 The system shall be able to write with a low-odor chisel tip Expo dry erase marker.

2.2 Software

2.2.1 The system shall be open source.

This will create an easily obtainable, low cost method of distributing the system's source code, which may be modified for personal use.

2.2.2 The system shall be capable of operating given only desired end effector cartesian coordinates specified with respect to the base frame.

This simplicity makes the system of use to inexperienced users.

3 Conceptual Design

The terminator T-2000 is a science-fiction spectacle of a robot – until you see the price. Channelling the inspiration many high school students may have for robotics, MEIOSIS robotics aims to provide an affordable manipulator to educators and enthusiasts. MEIOSIS uses primarily 3-D printed components and easily accessible materials. Among these materials are a Raspberry PI, smart servos and metal tubing. These features create an open-source manipulator accessible to the public to further robotics education.

3.1 Physical System Overview

The physical design of the robotic manipulator will be shown through Figures 1, 2, 3, and 4.

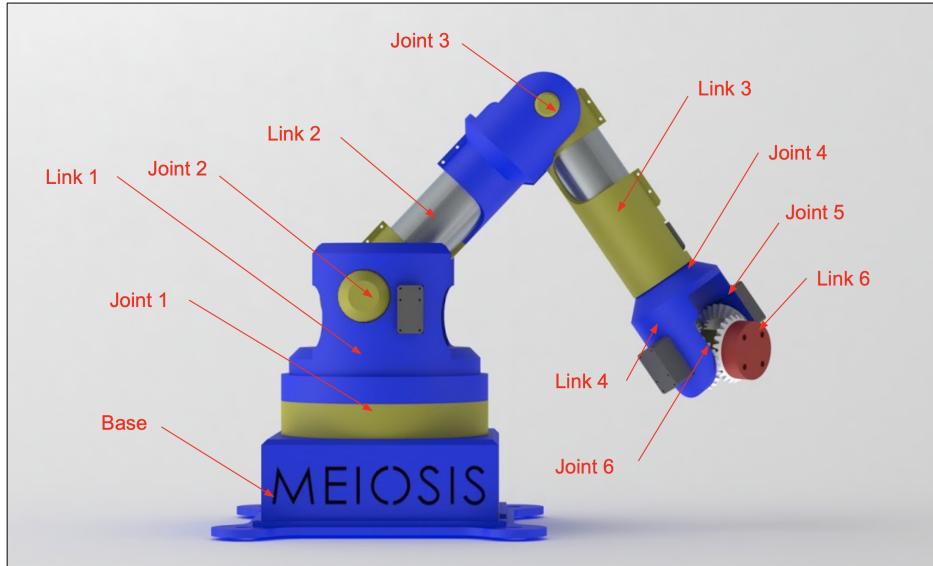


Figure 1: Overall System Conceptual Design

The colored links in *Figure 1* distinguish the different joints and links of the manipulator. The overall reach of the robot will be 582.5 mm. This length was chosen to decrease material cost and weight while still satisfying requirement 2.1.2 and 2.1.5, allowing the manipulating to pick and place objects to perform basic tasks. The base of the robot will be made to contain the Raspberry Pi and other electrical components.

3.1.1 Base

The base of the manipulator will house several of the electronic components, such as the computational system, power supply, and motor controller. A cross section of the base can be seen in *Figure 2*.

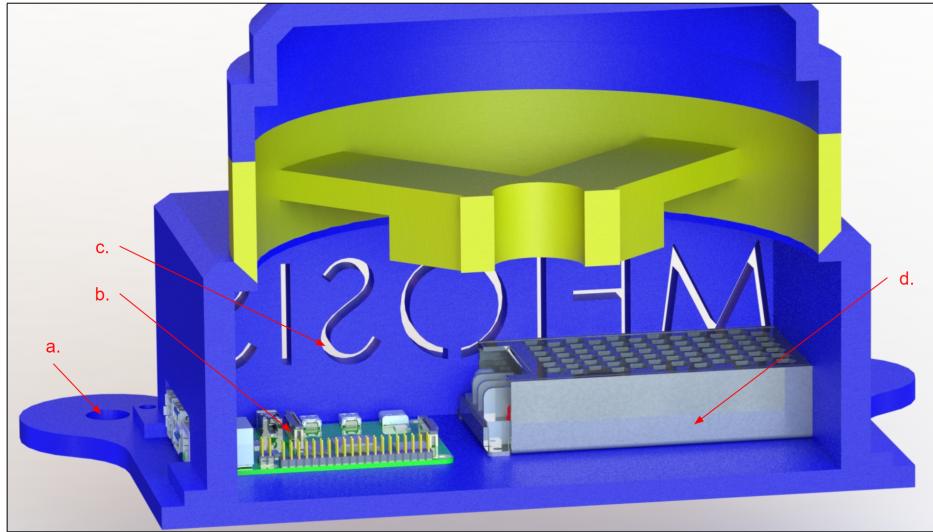


Figure 2: Manipulator Base with Call-outs

From *Figure 2*,

- a. *Base Supports*: The base supports are located at each corner of the base and will allow the base of the manipulator to be securely attached to a variety of surfaces with either standard bolt/fastener hardware or suction cups.
- b. *Computational System*: The computational system will be a Raspberry Pi; it will be housed in the base, which allows the Raspberry Pi to be more easily accessible. The primary reason for this system being chosen is to fulfill the budget requirement, 2.1.1. The Raspberry Pi will compute the manipulator's kinematics and command the motors accordingly.
- c. *Airflow Cutouts*: The side of the base will have cutouts to allow for airflow through the base; since the power supply is housed inside of the base as well as the computational system, the temperature must be regulated to prevent overheating.
- d. *Power Supply*: The power supply will be housed in the base as well; this allows the power supply to be more accessible and therefore more modifiable, so the end-user can easily expand the system to fulfill their needs.

3.1.2 Links

Figure 3 is an image of the robot that shows the links and their key features.

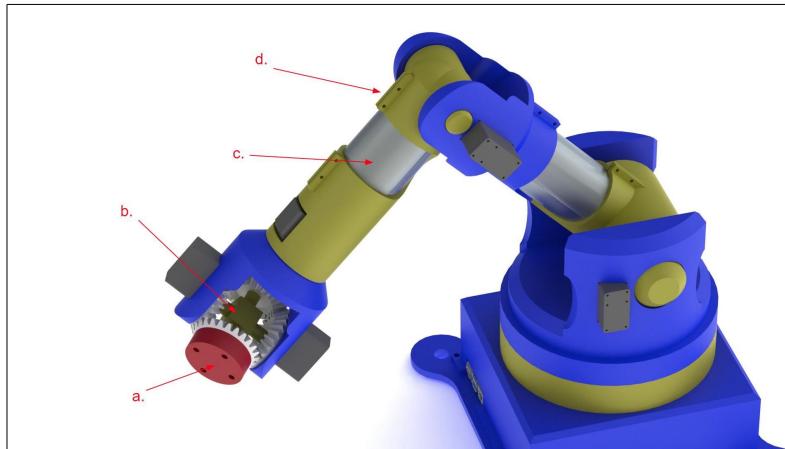


Figure 3: Drawing Showing Key Features of Design

Figure 3 highlights a few of the key features of our design. Call-out a shows the connection point for the end effector. The mountings are the standard used by the Sawyer manipulator. This may be adjusted to accommodate lower cost, more accessible end effectors. Call-out (b) shows the differential gearbox that will be used in the manipulator's wrist, saving space and weight. The manipulator will have aluminum tubing as support in the links (c) and will be attached to the 3D printed portion of the robot using clamp joints (d) tightened by screws.

Figure 4 is an image of the cross section of link 2 for the manipulator.

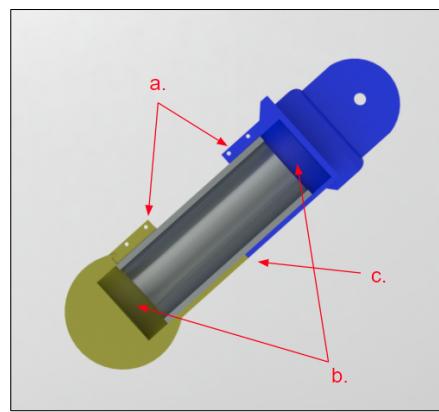


Figure 4: Drawing Showing Link Cross Section

The cross section seen in *Figure 4* shows the internal design for links two and three. It features two clamps that hold a hollow aluminum bar in place (a) and allows for gaps between the

aluminum tube and the 3D printed call-out (b). The proper length will be dictated by the 3D printed guides lining up at call-out (c). This allows for imprecision in the manufacturing of the aluminum tube.

3.2 System Functions

The system can be divided into two subsystems: the electrical and software systems. The electrical subsystem includes the wiring and hardware computational components, power system, actuators with drivers, and sensors. The software subsystem includes the algorithm for the computational system.

3.3 Electrical

Figure 5 is the block diagram for the electrical system of the manipulator.

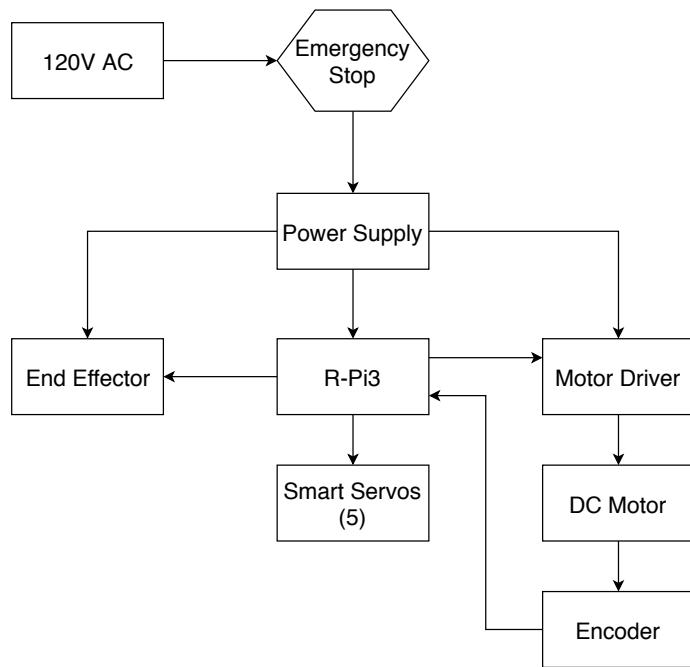


Figure 5: Electrical System Block Diagram

Figure 5 shows that the electrical systems of the manipulator will be relatively simple. Power is supplied by the 120V AC from standard wall outlets. A power supply will adapt the AC voltage to the required voltages for each component. One component is the Raspberry Pi, which will perform calculations for motor control (described below in software). It will send signals to the DC motor driver and the five smart servos. The smart servos have an on-board controller, so no feedback will be necessary. However, the first joint, between the base and the first link, will be actuated by a DC motor with an encoder to minimize cost.

3.4 Software

Figure 6 shows the software flowchart for the system.

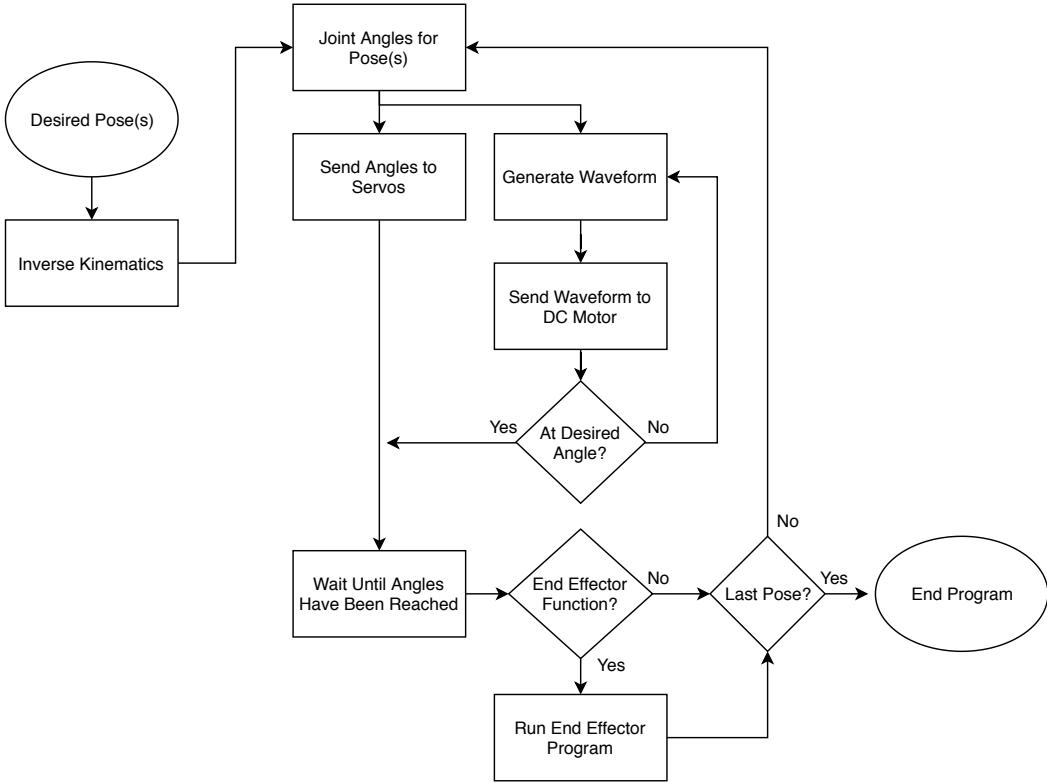


Figure 6: Software Flowchart

Similar to the electrical system, the software is also simple. *Figure 6* shows that the software will receive the desired pose or poses the user would like the manipulator to reach. Then the Raspberry Pi will use inverse kinematics to calculate the necessary joint angles. The waveforms/desired angles will be sent to the respective drivers/motors, and positional information will be sent back to the Raspberry Pi to adjust the DC motor angle. When the motors have reached their desired pose, the Raspberry Pi will actuate the end effector if it is specified by the user. The system will then check to see if there are any more poses to reach and either repeat the motor control section given the desired angles of the new pose or end program if the last pose has been reached.

4 Specifications

With the intention of making robotics education more accessible, The Manipulator for Educational Institutions with Open Source Integrated Systems (MEIOSIS) intends to provide high school educators and robot enthusiasts with a low cost manipulator. The system should be usable by novice students. It should also be modifiable to create a sustainably increased understanding of robotics. While MEIOSIS may not fully emulate industrial manipulators, it aims to provide more students with access to robotics education.

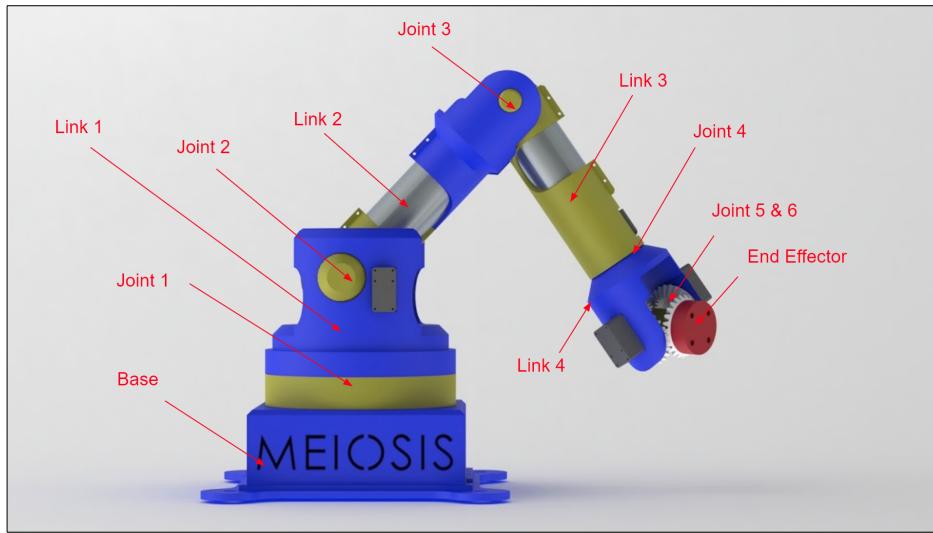


Figure 7: Overview of Physical System

The design seen in *Figure 7* is based on our conceptual design. It features four links and six joints for rotation and will be referenced throughout this document. The base of the manipulator and end-effector can also be seen in the figure.

4.1 Design Requirements

The specifications of the system are strictly based on the requirements defined previously. The requirements are divided into two primary categories, hardware and software.

4.2 Hardware

The following requirements and specifications are hardware specific and dictate the physical constraints the system must adhere to.

4.2.1 The system shall cost the end-user no more than \$1000.

4.2.1.a *The cost for the MEIOSIS team to develop the manipulator shall cost no more than \$800.*

4.2.2 The system shall be fully dexterous without being kinematically redundant.

4.2.2.a The system shall consist of six rotational joints connected by four links. The last three joints will create a spherical wrist.

As defined [1], “A manipulator having more than six DOF is referred to as a kinematically redundant manipulator (5).” A manipulator with less than six degrees of freedom will not be fully dexterous within it’s workspace. *Figure 9* (see subsection 4.2.6, p. 11) shows a six degree-of-freedom rotary manipulator with it’s coordinate frames in zeroed positions. The joint and link locations are seen in *Figure 7* (see section 4, p. 8).

4.2.2.b The system shall have no link offsets.

Link offsets as seen in *Figure 8* are commonly used to avoid singularities. However, having a link offset prevents the manipulator’s dexterous workspace from being a complete hemispherical shell.

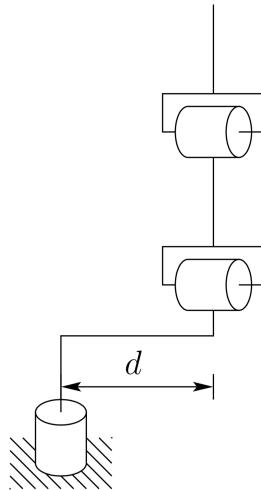


Figure 8: Elbow Manipulator Configuration with Link Offset [1]

As shown in *Figure 8*, the line directly above the first joint of the manipulator is offset such that the axes of the other joints are unable to become collinear with the base axis; this prevents singularity but causes a void in the dexterous workspace.

4.2.3 The system end effector shall maintain a positional accuracy magnitude of ± 1 mm and an orientation accuracy of $\pm 5^\circ$ eigen angle from the base frame.

To ensure that the robot has educational value, the accuracy must be defined so that any desired positions and movements are achieved.

4.2.3.a 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.

The addition of a calibration process allows the removal of any systematic errors, such as drift. The theoretical limit of the calibration process is the difference between the precision and accuracy metrics of the system.

4.2.4 The system end effector shall maintain a pose repeatability magnitude between 0.1—1.5 mm for the position and $\pm 4^\circ$ eigen angle from the base frame for the orientation.

4.2.4.a Joint one and two of the system shall possess an angle error of no more than .025 degrees.

Being that joint one and two are the first two rotational elements in the system, their error will propagate the most to the end effector's position.

4.2.4.b Joint three of the system shall possess an angle error of no more than .03 degrees.

Since joint three is closer to the end effector it's error will not propagate as severely throughout the system.

4.2.4.c Joints four, five, and six shall possess an angle error of no more than .29 degrees.

The spherical wrist is the closest to the end effector's final position and therefore has the least error propagation.

4.2.5 The system's reachable workspace shall be a hemisphere with a radius of 300-700 mm.

This workspace will provide enough movement to manipulate objects in order to perform basic tasks.

4.2.5.a The length of link one, two, three, four, and the wrist shall be 220.8 mm, 250 mm, 200 mm, 80 mm, and 52.5 mm respectively.

This results in a total height of 220.8 mm with a total reach of 582.5 mm in the zeroed configuration as shown in the configuration represented in *Figure 9*.

4.2.6 The system's dexterous workspace shall contain a hemispherical shell within the reachable workspace with a thickness of 280 mm.

This workspace will provide enough movement to manipulate objects in order to perform basic tasks. 280mm is slightly greater than the length of letter paper.

4.2.6.a The rotational limit of joint one, two, three, four, five, and six shall be $\pm 180^\circ$, -9.7°

to 177.5° , -150.6° to -19.3° , $\pm 180^\circ$, -180° to -1.6° , and $\pm 180^\circ$ respectively.

The angles stated are with respect to the kinematic model shown in *Figure 9*. To be fully dexterous within our 280 mm dexterous workspace the manipulator must have the joint angles specified above. The joint limitations were calculated by iteratively verifying the orientation about every point within the quarter hemisphere cross section seen in *Figure 32* (see Appendix, p. 38).

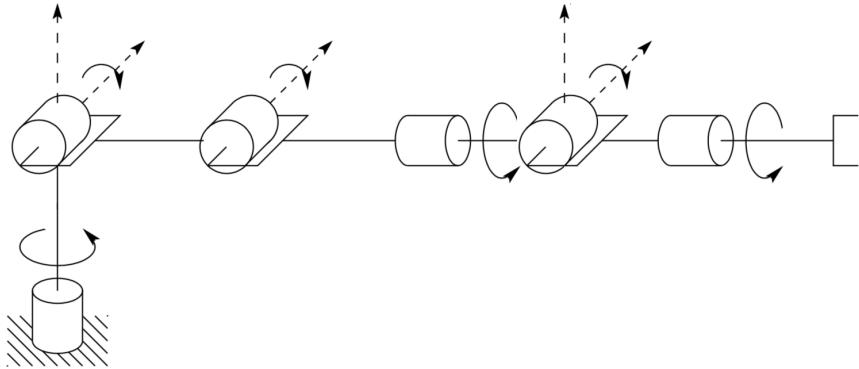


Figure 9: Kinematic Model Representing Zeroed Configuration [1]

4.2.7 The system shall have a removable end effector capable of picking and placing a low-odor chisel tip Expo dry erase marker.

This creates a robot capable of performing a variety of basic tasks, which enhances its educational value.

4.2.7.a The system shall use a parallel gripper that can close to 18mm.

The diameter of a low-odor chisel tip Expo dry erase marker is approximately 18 mm.

4.2.7.b The end effector shall attach to the manipulator using screws configured in a pattern that can accommodate a Dynamixel AX-12A servo.

It is expected that a majority of end effector styles will have to accommodate for a servo to facilitate actuation, therefore a pattern was chosen to standardize the mounting.

4.2.8 The system shall be able to write with a low-odor chisel tip Expo dry erase marker.

4.2.8.a The end effector shall be able to support 0.004 Newton meter moments about the axes normal to its gripping surfaces.

The coefficient of friction between the Expo marker and paper can be approximated and given the weight of an Expo marker the approximate grip strength of the end effector can

be calculated.

4.3 Software

The following requirements and specifications are software specific and determine the attributes of the operating system.

4.3.1 The system shall be open source.

This will create an easily obtainable, low cost method of distributing the system's source code, which may be modified for personal use.

4.3.1.a The software shall be hosted publicly on an online repository and maintain an MIT license for distribution.

This allows the end-user to freely download and modify the code without licensing. The MIT license disregards any legal obligation to code upkeep and documentation by the original author.

4.3.2 The system shall be capable of operating given only desired end effector cartesian coordinates specified with respect to the base frame.

4.3.2.a 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.

The system software interface facilitates an untrained user to operate without the advanced knowledge of the system's kinematics.

4.3.2.b The system shall be capable of performing floating point arithmetic.

The solution for the inverse kinematics requires the ability to perform high level arithmetic with little error.

5 Preliminary Design

The preliminary design section represents the most up-to-date status of the project and also encompasses information pertaining to several different specific design choices that had to be considered. The computer aided design section contains information regarding the physical design of the manipulator as well as noting key design features; the actuator analysis section contains information regarding actuator considerations and calculations being that the servo motors play an essential role in the functionality of the final system.

5.1 Computer Aided Design

The manipulator will be six degrees of freedom and will feature a spherical wrist. This design incorporates two different differential drive gearboxes that each control two degrees of freedom. This is to reduce the amount of required torques from the motors on the shoulder joint and reduce size in the wrist. In order to achieve accuracy requirements, timing belts are incorporated into the design in order to control the first three degrees of freedom. The following figure shows the joint configuration used in the design of the manipulator.

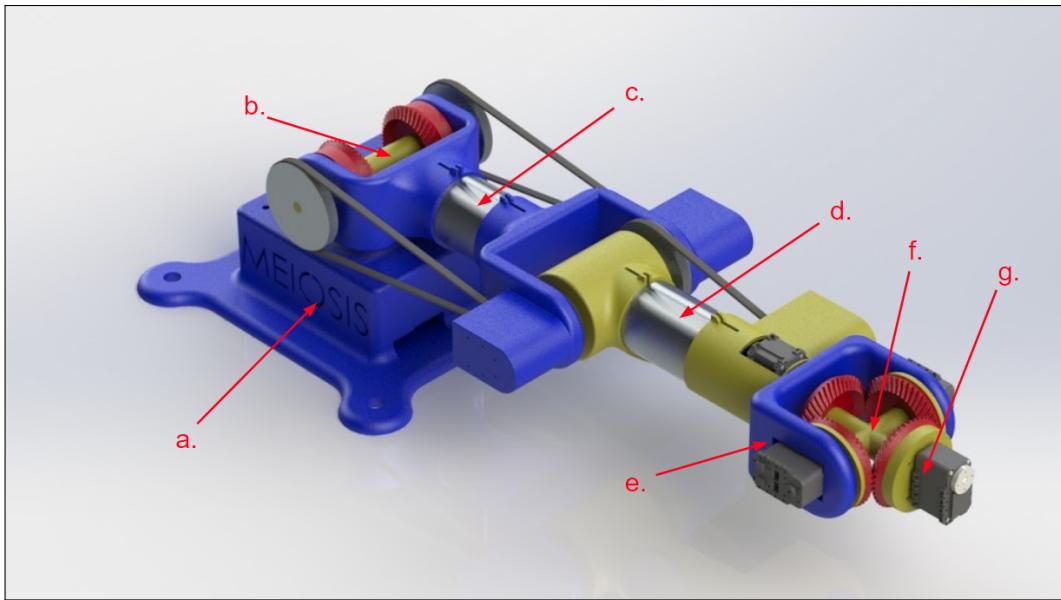


Figure 10: Manipulator in Zeroed Configuration with Callouts

Figure 10 shows the zeroed configuration for the manipulator with callouts showing the link locations. Point (a) shows the base of the manipulator. This is designed to hold the Raspberry Pi and power supply. The base features holes for airflow and I/O connections. Point (b) shows the first link of the robot and is the location of the first differential drive gearbox. Point (c) shows the second link of the robot, the servos located at the end of the link control the movement for the first two degrees of freedom at the shoulder and are connected to link 1 using timing belts. Point (d) shows the third link of the robot, there are two servos attached to the link that control the third degree of freedom (the elbow) and the fourth degree of freedom which is the first degree of freedom for the wrist. Point (e) shows

the fourth link of the manipulator which houses the servos used to control the fifth and sixth degree of freedom, which are both in the wrist's differential drive gearbox. Point (f) shows the fifth and final link of the robot. This link is support for the wrist's differential drive gearbox. Finally, point (g) shows the end effector which features connection points to hold a Dynamixel AX-12A smart servo. The next figure shows key features that are unique to links two and three.

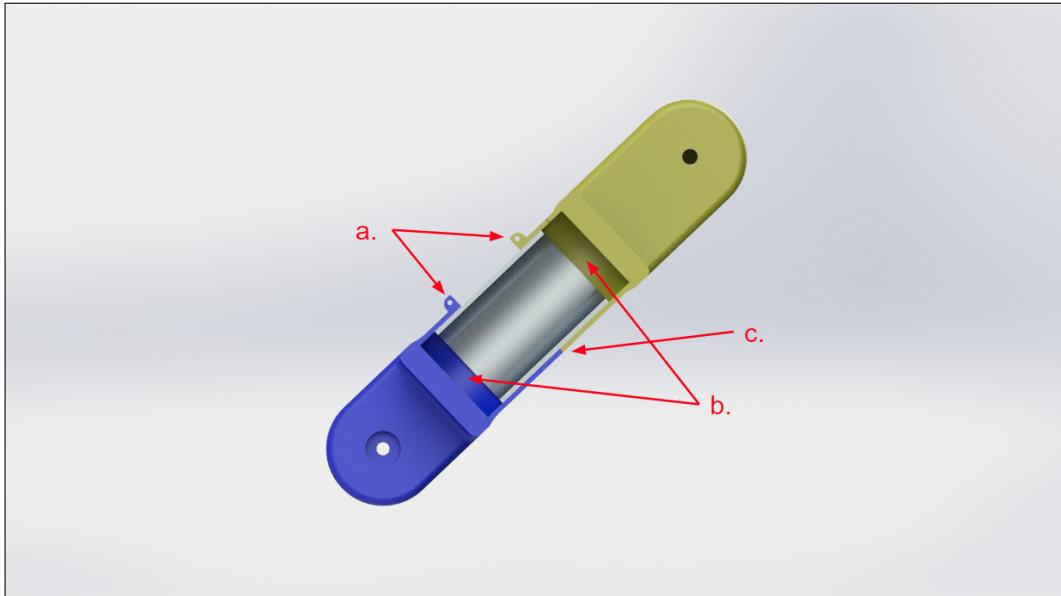


Figure 11: Link Cross Section

Figure 11 shows the cross-section for link two with callouts for key features in the design. Point (a) shows two clamping mechanisms that will be used to connect the aluminum pipe to the 3-D printed parts. This will allow for an easy way to assemble the manipulator while allowing for a large amount of tolerance in the length of the pipe when manufacturing the part. Point (b) shows an example of the gaps between the aluminum pipe and the 3-D printed parts. Point (c) shows where the 3-D printed parts will line up to ensure that the link is the correct length and orientation.

5.1.1 Pulleys

A 12 tooth and 120 tooth pulleys provide a 10:1 gear ratio and use a 0.25 in wide MXL belt. The center distance between the pulleys must be at least 5.43 in to have 5 teeth meshing. Because of the high gear ratio, greater distances do not increase the number of teeth in mesh. *Figure 12* shows the distance between the manipulator's first two pulleys is 235.185 mm or 9.259 in, which corresponds to a belt with 300 and pitch diameter of 24 in.

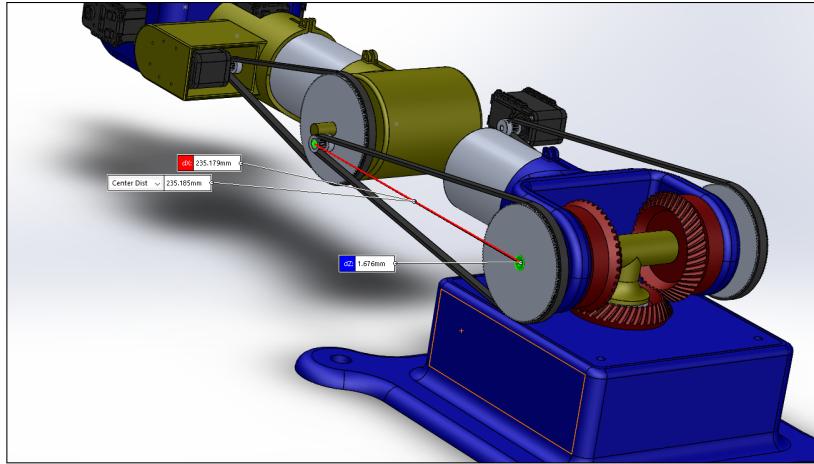


Figure 12: Pulley 1-2 Center Distance

Figure 13 shows the center distance between the second pulleys is 139.960 mm or 5.510 in. The corresponding belt has 208 teeth with a 16.6 in pitch diameter.

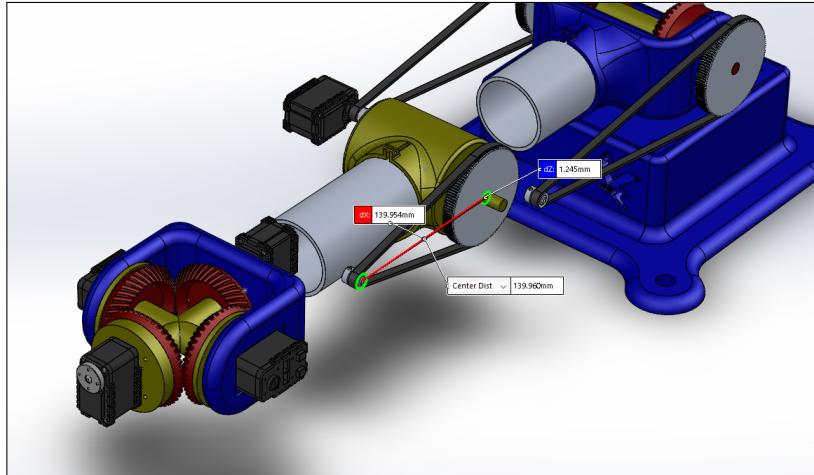


Figure 13: Pulley 3 Center Distance

5.2 Actuator Analysis

Dynamixel does not provide a stall torque for the MX-12W servo, however it does provide stall torque data for two similar servos. The MX-12W's stall torque can be found from the stall torque data since the motor are the same. The stall torque of the MX-12W should be comparable to the AX-12W's stall torque. The MX-12W and AX-12W operate at the same 12 V voltage, 32:1 gear ratio, and 470 rpm no load output speed. The AX-12A also operates at 12 V, but at with a 254:1 gear ratio and 59 rpm no load output speed. Torque and current are inversely related. If the AX-12A's and AX-12W's motors are the same, then when operating at the same voltage, the output speed of the AX-12W should equal to the output speed of the AX-12A multiplied by the ratio of the servos' gear ratios and the torque

of the AX-12W should be the AX-12A's torque multiplied by the inverse of the gear ratios ratio:

$$59 \cdot \frac{254}{32} = 468 \cong 470$$

The AX-12A's and MX-12W's stall torques are 1.5Nm and 0.21Nm, respectively [2]:

$$1.5 \cdot \frac{32}{254} = 0.189 \cong 0.21\text{Nm}$$

Therefore, the AX-12A and MX-12W use the same motor. Additionally, if operating at the same voltage, the stall torque is defined by the no load output speed and gear ratio. Since the MX-12W and AX-12A operate at the same 12 V voltage with the same 32:1 gear ratio and 470 rpm no load output speed, their stall torques should be the same. More specifically, it is between the AX-12W's and the gear reduced AX-12A's stall torques of 0.21Nm and 0.189Nm, respectively. Therefore, the stall MX-12W's stall torque is within 0.01Nm of 0.2Nm. The manipulator is designed for the worst case scenario: 0.19 Nm.

The first differential has its greatest torque when the manipulator is in the zeroed configuration. In the zeroed configuration, the arm is outstretched with gravity acting perpendicular to the arm's length. Since the manipulator must only support a chisel tip dry erase Expo marker, the maximum torque may be approximated as the manipulator's arm's mass acting near the centroid. More mass from the servos and longer aluminum support in link 3 place the center of mass closer to the end-effector than the base. Therefore, the point at which the arm's mass acts is assumed to be 273 mm from the base. These calculations will be further revised with mass parameters from the STL files generating the open and closed loop simulations. *Table 1* estimates the manipulator's arm's mass.

Table 1: Manipulator Link 1 Mass Tabulations

Item	Specifications	Weight (g)	Notes
3-D Printed Materials	30% infill	353	100% infill: 1177g
Servos	7 low torque	385	55 g/servo
Aluminum Support	.75" OD .125" thick 1' long	287	44.23 g/in ³
Bearings	5 bearings	60	12 g/bearing
End Effector		50	Approx.
Safety Factor		205	
Total		1340	

The total mass of 1340g as seen in *Table 1* yields a moment of 0.361 Nm or 3.685 kgm. To generate smooth motions, Dynamixel recommends operating at one fifth of the stall torque. Therefore, the actual maximum required torque is 1.805 Nm. An MX-12W provides 0.19 Nm of torque, necessitating a 10:1 gear ratio. While the AX-12A provides a greater torque of 1.5 Nm, it would also require gearing. It is only precise to 0.8 degrees, while the specifications require at least 0.29 degrees of precision for joints one and two. Further, the AX-12A only receives position feedback for 300 degrees of its rotation and cannot track multiple rotations, making it untenable for the base motor, which must rotate more than 360 degrees.

The third joint's torque requirements can be approximated in the same manner. However, links three, four, and five's masses are assumed to act 200 mm from the third joint 3. *Table 2* estimates the links' masses.

Table 2: Manipulator Link 2 Mass Tabulations

Item	Specifications	Weight (g)	Notes
3-D Printed Materials	30% infill	180	100% infill: 1177g
Servos	4 low torque	220	55 g/servo
Aluminum Support	.75" OD .125" thick 1' long	140	44.23 g/in ³
Bearings	5 bearings	36	12 g/bearing
End Effector		50	Approx.
Safety		205	
Total		831	

The resulting moment is 1.630 Nm, which can be satisfied by the same motor and gear ratio used on the base.

5.3 Forward Kinematics

The forward kinematics of the manipulator are described by the equations below, where the reference coordinate frames are given by *Figure 14*.

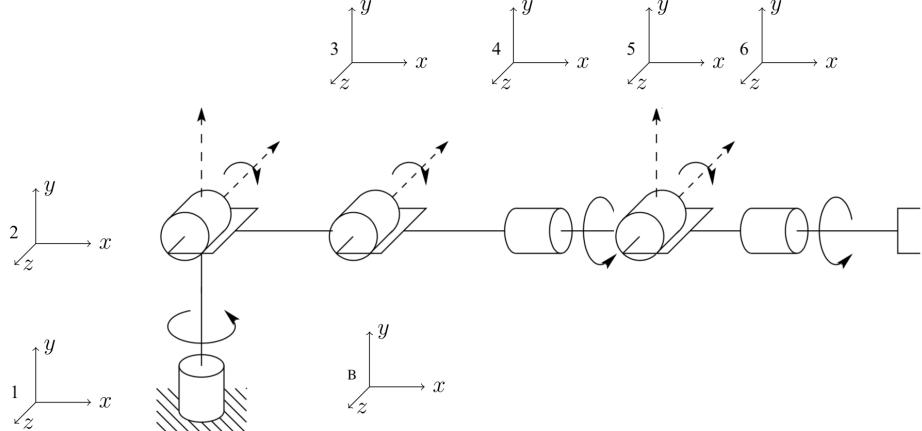


Figure 14: Coordinate Systems

Given the lengths of each of the manipulator links,

$${}^I_B r_1 = \begin{bmatrix} 0 \\ 0 \\ \ell_b \end{bmatrix} \quad {}^1_1 r_2 = \begin{bmatrix} 0 \\ 0 \\ \ell_1 \end{bmatrix} \quad {}^2_2 r_3 = \begin{bmatrix} 0 \\ \ell_2 \\ 0 \end{bmatrix} \quad {}^3_3 r_4 = \begin{bmatrix} 0 \\ \ell_3 \\ 0 \end{bmatrix} \quad {}^4_4 r_5 = \begin{bmatrix} 0 \\ \ell_4 \\ 0 \end{bmatrix} \quad {}^5_5 r_6 = \begin{bmatrix} 0 \\ \ell_5 \\ 0 \end{bmatrix}$$

The position of each link relative to the inertial frame is given as:

$$\begin{aligned} {}^I_B r_1 &= {}^B r_1 & {}^I_B r_2 &= r_1 + {}^I T_{11} {}^1 r_2 & {}^I_B r_3 &= r_2 + {}^I T_{22} {}^2 r_3 \\ {}^I_B r_4 &= r_3 + {}^I T_{33} {}^3 r_4 & {}^I_B r_5 &= r_4 + {}^I T_{44} {}^4 r_5 & {}^I_B r_6 &= r_5 + {}^I T_{55} {}^5 r_6 \end{aligned}$$

$$\begin{aligned}
{}^I_B r_1 &= \begin{bmatrix} 0 \\ 0 \\ \ell_b \end{bmatrix} & {}^I_B r_2 &= \begin{bmatrix} 0 \\ 0 \\ \ell_b + \ell_1 \end{bmatrix} & {}^I_B r_3 &= \begin{bmatrix} -\ell_2 c_{\theta_2} s_{\theta_1} \\ \ell_2 c_{\theta_{12}} \\ \ell_b + \ell_1 + \ell_2 s_{\theta_2} \end{bmatrix} & {}^I_B r_4 &= \begin{bmatrix} -s_{\theta_1} (\ell_3 c_{\theta_{23}} + \ell_2 c_{\theta_2}) \\ c_{\theta_1} (\ell_3 c_{\theta_{23}} + \ell_2 c_{\theta_2}) \\ \ell_1 + \ell_b + \ell_3 s_{\theta_{23}} + \ell_2 s_{\theta_2} \end{bmatrix} \\
{}^I_B r_5 &= \begin{bmatrix} -s_{\theta_1} (\ell_3 c_{\theta_{23}} + \ell_4 c_{\theta_{23}} + \ell_2 c_{\theta_2}) \\ c_{\theta_1} (\ell_3 c_{\theta_{23}} + \ell_4 c_{\theta_{23}} + \ell_2 c_{\theta_2}) \\ \ell_1 + \ell_b + \ell_3 s_{\theta_{23}} + \ell_4 s_{\theta_{23}} + \ell_2 s_{\theta_2} \end{bmatrix} \\
{}^I_B r_6 &= \begin{bmatrix} \ell_5 c_{\theta_1} s_{\theta_4} s_{\theta_5} - \ell_4 c_{\theta_{23}} s_{\theta_1} - \ell_2 c_{\theta_2} s_{\theta_1} - \ell_5 c_{\theta_{23}} c_{\theta_5} s_{\theta_1} - \ell_3 c_{\theta_{23}} s_{\theta_1} \\ + \ell_5 c_{\theta_2} c_{\theta_4} s_{\theta_1} s_{\theta_3} s_{\theta_5} + \ell_5 c_{\theta_3} c_{\theta_4} s_{\theta_1} s_{\theta_2} s_{\theta_5} \\ \ell_5 (s_{\theta_5} (s_{\theta_1} s_{\theta_4} - c_{\theta_4} (c_{\theta_1} c_{\theta_2} s_{\theta_3} + c_{\theta_1} c_{\theta_3} s_{\theta_2})) - c_{\theta_5} (c_{\theta_1} s_{\theta_2} s_{\theta_3} - c_{\theta_1} c_{\theta_2} c_{\theta_3})) \\ - \ell_3 (c_{\theta_1} s_{\theta_2} s_{\theta_3} - c_{\theta_1} c_{\theta_2} c_{\theta_3}) - \ell_4 (c_{\theta_1} s_{\theta_2} s_{\theta_3} - c_{\theta_1} c_{\theta_2} c_{\theta_3}) + \ell_2 c_{\theta_1} c_{\theta_2} \\ \ell_1 + \ell_b + \ell_3 s_{\theta_{23}} + \ell_4 s_{\theta_{23}} + \ell_2 s_{\theta_2} + \frac{\ell_5 c_{\theta_{23}} s_{\theta_{45}}}{2} + \ell_5 s_{\theta_{23}} c_{\theta_5} - \frac{\ell_5 s_{\theta_4 - \theta_5} c_{\theta_{23}}}{2} \end{bmatrix}
\end{aligned}$$

Given the direction cosine matrices,

$$\begin{aligned}
rotx(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, & roty(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \\
rotz(\theta) &= \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \end{bmatrix}
\end{aligned}$$

The orientation of each link with respect to the inertial frame is given as:

$$\begin{aligned}
{}^I T_1 &= rotz(\theta_1) \\
{}^I T_2 &= rotz(\theta_1) rotx(\theta_2) \\
{}^I T_3 &= rotz(\theta_1) rotx(\theta_2) rotx(\theta_3) \\
{}^I T_4 &= rotz(\theta_1) rotx(\theta_2) rotx(\theta_3) roty(\theta_4) \\
{}^I T_5 &= rotz(\theta_1) rotx(\theta_2) rotx(\theta_3) roty(\theta_4) rotx(\theta_5) \\
{}^I T_6 &= rotz(\theta_1) rotx(\theta_2) rotx(\theta_3) roty(\theta_4) rotx(\theta_5) roty(\theta_6) \\
{}^I T_1 &= \begin{bmatrix} c_{\theta_1} & -s_{\theta_1} & 0 \\ s_{\theta_1} & c_{\theta_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} & {}^I T_2 &= \begin{bmatrix} c_{\theta_1} & -c_{\theta_2} s_{\theta_1} & s_{\theta_1} s_{\theta_2} \\ s_{\theta_1} & c_{\theta_1} c_{\theta_2} & -c_{\theta_1} s_{\theta_2} \\ 0 & s_{\theta_2} & c_{\theta_2} \end{bmatrix} & {}^I T_3 &= \begin{bmatrix} c_{\theta_1} & -c_{\theta_{23}} s_{\theta_1} & s_{\theta_{23}} s_{\theta_1} \\ s_{\theta_1} & c_{\theta_{23}} c_{\theta_1} & -s_{\theta_{23}} c_{\theta_1} \\ 0 & s_{\theta_{23}} & c_{\theta_{23}} \end{bmatrix} \\
{}^I T_4 &= \begin{bmatrix} c_{\theta_1} c_{\theta_4} - s_{\theta_{23}} s_{\theta_1} s_{\theta_4} & -c_{\theta_{23}} s_{\theta_1} & c_{\theta_1} s_{\theta_4} + s_{\theta_{23}} c_{\theta_4} s_{\theta_1} \\ c_{\theta_4} s_{\theta_1} + s_{\theta_{23}} c_{\theta_1} s_{\theta_4} & c_{\theta_{23}} c_{\theta_1} & s_{\theta_1} s_{\theta_4} - s_{\theta_{23}} c_{\theta_1} c_{\theta_4} \\ -c_{\theta_{23}} s_{\theta_4} & s_{\theta_{23}} & c_{\theta_{23}} c_{\theta_4} \end{bmatrix} \\
{}^I T_5 &= \begin{bmatrix} c_{\theta_1} c_{\theta_4} - s_{\theta_{23}} s_{\theta_1} s_{\theta_4} & s_{\theta_5} (c_{\theta_1} s_{\theta_4} + s_{\theta_{23}} c_{\theta_4} s_{\theta_1}) - c_{\theta_{23}} c_{\theta_5} s_{\theta_1} & c_{\theta_5} (c_{\theta_1} s_{\theta_4} + s_{\theta_{23}} c_{\theta_4} s_{\theta_1}) + c_{\theta_{23}} s_{\theta_1} s_{\theta_5} \\ c_{\theta_4} s_{\theta_1} + s_{\theta_{23}} c_{\theta_1} s_{\theta_4} & s_{\theta_5} (s_{\theta_1} s_{\theta_4} - s_{\theta_{23}} c_{\theta_1} c_{\theta_4}) + c_{\theta_{23}} c_{\theta_1} c_{\theta_5} & c_{\theta_5} (s_{\theta_1} s_{\theta_4} - s_{\theta_{23}} c_{\theta_1} c_{\theta_4}) - c_{\theta_{23}} c_{\theta_1} s_{\theta_5} \\ -c_{\theta_{23}} s_{\theta_4} & s_{\theta_{23}} c_{\theta_5} + c_{\theta_{23}} c_{\theta_4} s_{\theta_5} & c_{\theta_{23}} c_{\theta_4} c_{\theta_5} - s_{\theta_{23}} s_{\theta_5} \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
{}^I T_6 &= \begin{bmatrix} {}^I T_{6(1,1)} & {}^I T_{6(1,2)} & {}^I T_{6(1,3)} \\ {}^I T_{6(2,1)} & {}^I T_{6(2,2)} & {}^I T_{6(2,3)} \\ {}^I T_{6(3,1)} & {}^I T_{6(3,2)} & {}^I T_{6(3,3)} \end{bmatrix} \\
{}^I T_{6(1,1)} &= c_{\theta_6}(c_{\theta_1}c_{\theta_4} - s_{\theta_{23}}s_{\theta_1}s_{\theta_4}) - s_{\theta_6}(c_{\theta_5}(c_{\theta_1}s_{\theta_4} + s_{\theta_{23}}c_{\theta_4}s_{\theta_1}) + c_{\theta_{23}}s_{\theta_1}s_{\theta_5}) \\
{}^I T_{6(1,2)} &= s_{\theta_5}(c_{\theta_1}s_{\theta_4} + s_{\theta_{23}}c_{\theta_4}s_{\theta_1}) - c_{\theta_{23}}c_{\theta_5}s_{\theta_1} \\
{}^I T_{6(1,3)} &= c_{\theta_6}(c_{\theta_5}(c_{\theta_1}s_{\theta_4} + s_{\theta_{23}}c_{\theta_4}s_{\theta_1}) + c_{\theta_{23}}s_{\theta_1}s_{\theta_5}) + s_{\theta_6}(c_{\theta_1}c_{\theta_4} - s_{\theta_{23}}s_{\theta_1}s_{\theta_4}) \\
{}^I T_{6(2,1)} &= c_{\theta_6}(c_{\theta_4}s_{\theta_1} + s_{\theta_{23}}c_{\theta_1}s_{\theta_4}) - s_{\theta_6}(c_{\theta_5}(s_{\theta_1}s_{\theta_4} - s_{\theta_{23}}c_{\theta_1}c_{\theta_4}) - c_{\theta_{23}}c_{\theta_1}s_{\theta_5}) \\
{}^I T_{6(2,2)} &= s_{\theta_5}(s_{\theta_1}s_{\theta_4} - s_{\theta_{23}}c_{\theta_1}c_{\theta_4}) + c_{\theta_{23}}c_{\theta_1}c_{\theta_5} \\
{}^I T_{6(2,3)} &= s_{\theta_6}(c_{\theta_4}s_{\theta_1} + s_{\theta_{23}}c_{\theta_1}s_{\theta_4}) + c_{\theta_6}(c_{\theta_5}(s_{\theta_1}s_{\theta_4} - s_{\theta_{23}}c_{\theta_1}c_{\theta_4}) - c_{\theta_{23}}c_{\theta_1}s_{\theta_5}) \\
{}^I T_{6(3,1)} &= s_{\theta_6}(s_{\theta_{23}}s_{\theta_5} - c_{\theta_{23}}c_{\theta_4}c_{\theta_5}) - c_{\theta_{23}}c_{\theta_6}s_{\theta_4} \\
{}^I T_{6(3,2)} &= s_{\theta_{23}}c_{\theta_5} + c_{\theta_{23}}c_{\theta_4}s_{\theta_5} \\
{}^I T_{6(3,3)} &= c_{\theta_6}(s_{\theta_{23}}s_{\theta_5} - c_{\theta_{23}}c_{\theta_4}c_{\theta_5}) - c_{\theta_{23}}s_{\theta_4}s_{\theta_6}
\end{aligned}$$

5.4 Velocity Kinematics

The translational and rotational velocities (\dot{r} & ω) can be found given the geometric jacobian of the body and transformation matrix corresponding to it.

$$\begin{bmatrix} {}^B_B \omega_I \\ {}^I_B \dot{r}_B \end{bmatrix} = J_B = \begin{bmatrix} {}^I T_B(:, 3)^T \cdot \frac{\partial}{\partial \gamma} {}^I T_B(:, 2) \\ {}^I T_B(:, 1)^T \cdot \frac{\partial}{\partial \gamma} {}^I T_B(:, 3) \\ {}^I T_B(:, 2)^T \cdot \frac{\partial}{\partial \gamma} {}^I T_B(:, 1) \end{bmatrix}$$

5.5 Inverse Kinematics

The inverse kinematics can be calculated given desired position and orientation vectors, o and R , respectively.

$$\begin{bmatrix} x_c & y_c & z_c \end{bmatrix} = o, \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Inverse Position:

$$\begin{aligned}\theta_1 &= \text{atan2}(x_c, y_c) - \pi/2 \\ \theta_2 &= \text{atan2}\left(z_c - \ell_1, \sqrt{x_c^2 + y_c^2}\right) - \text{atan2}(\ell_3 s_3, \ell_2 + \ell_3 c_3) \\ \theta_3 &= \text{atan2}(-\sqrt{1 - D^2}, D) \\ \text{where } D &\equiv \frac{x_c^2 + y_c^2 + (z_c - \ell_1)^2 - \ell_2^2 - \ell_3^2}{2\ell_2\ell_3}\end{aligned}$$

Inverse Orientation :

$$\begin{aligned}{}^I T_3 &= \text{rotz}(\theta_1) \text{rotx}(\theta_2) \text{rotx}(\theta_3) \\ {}^3 T_6 &= {}^I T_3^T R \\ \theta_4 &= \text{atan2}\left({}^3 T_{6(1,2)}, {}^3 T_{6(3,2)}\right) \\ \theta_5 &= \text{atan2}\left({}^3 T_{6(3,2)}/c_4, {}^3 T_{6(2,2)}\right) \\ \theta_6 &= \text{atan2}\left({}^3 T_{6(2,1)}, -{}^3 T_{6(2,3)}\right)\end{aligned}$$

5.6 Equations of Motion

Given robot dynamics described by $H(\gamma)\ddot{\gamma} + d(\gamma, \dot{\gamma}) + G(\gamma) = F_\gamma$, the equations of motion for the manipulator can be determined. Solving this equation for $\ddot{\gamma}$ gives:

$$\ddot{\gamma} = H(\gamma)^{-1} (F_\gamma - d(\gamma, \dot{\gamma}) - G(\gamma)) \quad (1)$$

Where F_γ is the vector of generalized forces on the joints and

$$\begin{aligned}H(\gamma) &= \sum_B^N J_B(\gamma)^T \begin{bmatrix} {}^B J & \mathring{S}({}^B \Gamma)^I T_B^T \\ {}^I T_B \mathring{S}({}^B \Gamma)^T & m_B I \end{bmatrix} J_B(\gamma), \quad {}^B \Gamma = {}^B r_{cm} m_b, \quad \mathring{S}(\omega)r = (\omega \times r) \\ d(\gamma, \dot{\gamma}) &= \sum_B^N J_B(\gamma)^T \begin{bmatrix} {}^B J & \mathring{S}({}^B \Gamma)^I T_B^T \\ {}^I T_B \mathring{S}({}^B \Gamma)^T & m_B I \end{bmatrix} \dot{J}_B(\gamma, \dot{\gamma}) + J_B(\gamma)^T \begin{bmatrix} {}^B \omega_I \times {}^B J_B^B \omega_I \\ {}^I T_B \left({}^B \omega_I \times ({}^B \omega_I \times {}^B \Gamma) \right) \end{bmatrix} \\ G(\gamma) &= \left(\frac{\partial U({}^I r(\gamma))}{\partial \gamma} \right)^T, \quad U_B = [0 \ 0 \ g] \left({}^I_B r_B m_B + {}^I T_B {}^B \Gamma \right)\end{aligned}$$

5.6.1 Actuator Dynamics

Given robot dynamics described by $H(\gamma)\ddot{\gamma} + n(\gamma, \dot{\gamma}) = \tau$, the torque, τ , provided by the servo motors is necessary to solve the closed loop dynamics of the system. Assuming the servo is driven by a D.C. motor with proportional derivative control,

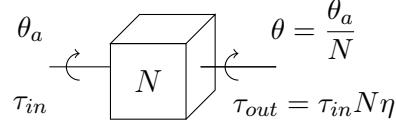
$$\tau_a = Ki_a = J_a \ddot{\theta}_a + b_a \dot{\theta}_a + \tau_L \quad (2)$$

Where τ_a is the actuator torque, K is the back-EMF constant, i_a is the motor current, J_a is the armature inertia, θ_a , $\dot{\theta}_a$, $\ddot{\theta}_a$ is the motor position and it's first and second time

derivatives respectively, b_a is the viscous friction coefficient, and τ_L is the torque available for the actuator to do work. The basic equation for a motor is known to be:

$$V_a = i_a R_a + K \dot{\theta}_a \quad (3)$$

Where V_a is the voltage applied to the actuator and R_a is the armature resistance. Given a gearbox with in/out ratio N and efficiency η ,



The motor equation (2) can be expressed in the output coordinates:

$$Ki_a = J_a N \ddot{\theta} + b_a N \dot{\theta} + \frac{\tau}{N\eta}$$

Substituting into equation (3) and solving for i_a :

$$\begin{aligned} i_a &= \frac{J_a N}{K} \ddot{\theta} + b_a N \dot{\theta} + \frac{\tau}{N\eta} \\ V_a &= \frac{R_a J_a N}{K} \ddot{\theta} + \frac{R_a b_a N}{K} \dot{\theta} + \frac{R_a}{KN\eta} \tau + KN \dot{\theta} \end{aligned} \quad (4)$$

Assuming PD control, $V_a = K_p(\theta - \theta_d) + K_d \dot{\theta}$, where θ_d is the desired orientation of the actuator, the following solution is found by setting the PD solution equal to (4). After collecting like terms:

$$\frac{R_a J_a N}{K} \ddot{\theta} + \left(\frac{R_a J_a N}{K} - K_d + KN \right) \dot{\theta} - K_p \theta = -K_p \theta_d - \frac{R_a}{KN\eta} \tau \quad (5)$$

The following parameters of the system can be obtained by applying a step input to the system with $\tau = 0$ and measuring the characteristics of it's response. Denoting ζ as the damping ratio and ω_n as the natural frequency of the system,

$$\% \text{ Overshoot} = \left(\frac{\theta_{max} - \theta_{ss}}{\theta_{ss}} \right) \times 100, \quad \zeta = \frac{-\ln(\% \text{OS}/100)}{\sqrt{\pi^2 + \ln^2(\% \text{OS}/100)}}, \quad \omega_n = \frac{\pi}{T_p \sqrt{1 - \zeta^2}}$$

Given θ_{max} , θ_{ss} , and T_p as measured parameters of the system's max output, steady state, and time to peak, respectively.

Refactoring equation (5) and equating with the general solution for a second order system given by $\ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2\theta = \omega_n^2\theta_d$, the following solutions are found:

$$2\zeta\omega_n = \frac{b_a}{J_a} - \frac{KK_d}{R_a J_a N} + \frac{K^2}{R_a J_a} \quad (6) \qquad \omega_n^2 = \frac{-KK_p}{R_a J_a N} \quad (7)$$

Performing a similar experiment as previously described, except with a known inertial load $\tau = J_m \ddot{\theta}$, the following parameters can be found:

$$\alpha_m \equiv 2\zeta\omega_n = \frac{R_a b_a N^2 \eta - KK_d N \eta + K^2 N^2 \eta}{R_a J_a N^2 \eta + R_a J_m}, \quad \beta_m \equiv \omega_n = -\frac{KK_p N \eta}{R_a J_a N^2 \eta + R_a J_m}$$

$$\begin{bmatrix} 1 & -(\alpha_1 J_1 + \beta_1 J_1) \\ 1 & -(\alpha_2 J_2 + \beta_2 J_2) \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \frac{R_a b_a N^2 \eta - KK_d N \eta + K^2 N^2 \eta - KK_p N \eta}{R_a J_a N^2 \eta} \\ \frac{1}{J_a N^2 \eta} \end{bmatrix} = \begin{bmatrix} \alpha_1 + \beta_1 \\ \alpha_2 + \beta_2 \\ \vdots \end{bmatrix} \quad (8)$$

With multiple datasets (varying inertial loads, J_m), the solutions of (8) can be found using the least-squares method, yielding

$$\frac{R_a b_a N - KK_d + K^2 N \eta - KK_p}{R_a J_a N} \quad (9) \qquad \frac{1}{J_a N^2 \eta} \quad (10)$$

Finally, the coefficients of the second order system (11) are known:

$$\underbrace{\left(J_a N^2 \eta \right)}_{1/(10)} \ddot{\theta} + \underbrace{\left(\frac{R_a b_a N^2 \eta - KK_d N \eta + K^2 N^2 \eta}{R_a} \right)}_{(6)/(10)} \dot{\theta} - \underbrace{\left(\frac{KK_p N \eta}{R_a} \right)}_{(7)/(10)} \theta + \underbrace{\left(\frac{KK_p N \eta}{R_a} \right)}_{(7)/(10)} \theta_d = -\tau \quad (11)$$

The MATLAB code implementing this process can be found in the Appendix (see section 5.13, p. 38, *Listing 1*). The torque provided by the servo can now be solved for, given the current position (θ), velocity ($\dot{\theta}$), angular acceleration ($\ddot{\theta}$), and desired position (θ_d) are known.

Given the equation of motion for the dynamical response of the system (1), substituting in the solution obtained for the motor dynamics and solving for the acceleration,

$$\left(H + J_a N^2 \eta \right)^{-1} \left[\left(B - \frac{R_a b_a N^2 \eta - KK_d N \eta + K^2 N^2 \eta}{R_a} \right) \dot{\gamma} - \left(\frac{KK_p N \eta}{R_a} \right) (\gamma_d - \gamma) - n \right] = \ddot{\gamma} \quad (12)$$

Where

$$n(\gamma, \dot{\gamma}) = d(\gamma, \dot{\gamma}) + G(\gamma) + C \text{sgn}(\dot{\gamma})$$

5.7 Open-Loop Simulation

The equations of motion described in equation (1) can be integrated to simulate the motion of the system. For the open loop control, no input torque is supplied, meaning the system

responds only to gravity. Due to the complexity of the equations of motion, they will be integrated numerically using a 4th Order Runge Kutta algorithm. Since no input force is supplied, the equations of motion reduce down to the relation described in equation (13).

$$\ddot{\gamma} = H(\gamma)^{-1}(-d(\gamma, \dot{\gamma}) - G(\gamma)) \quad (13)$$

The resulting simulation shows the manipulator assembly starting in the zeroed configuration, (*Figure 15a*) then "falling" due to the forces of gravity acting on the links (*Figure 15b*). The manipulator continues to oscillate analogous to a simple pendulum, being that there are no frictional forces accounted for.

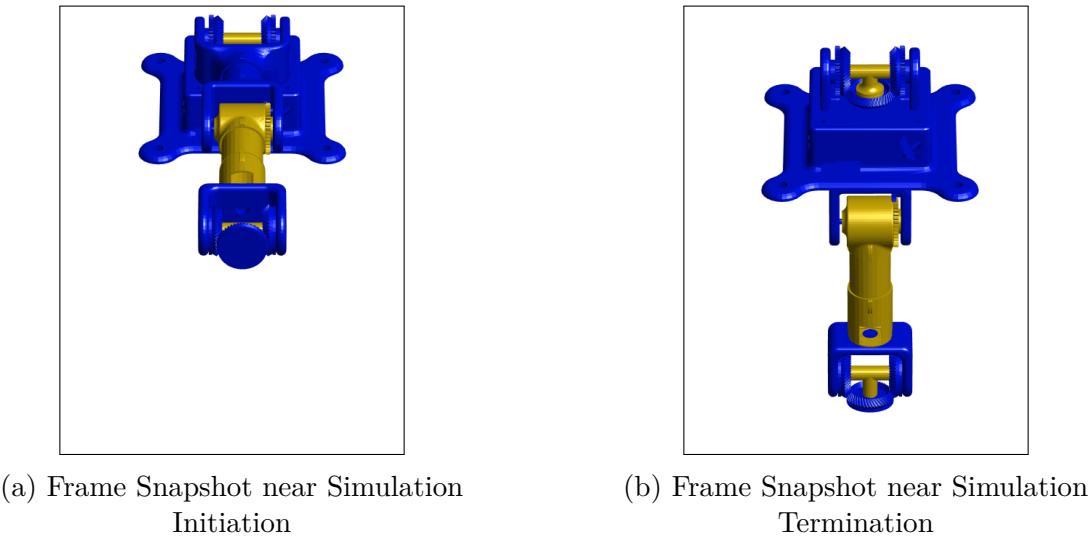


Figure 15: Open-Loop Control Simulation Animation Snapshots

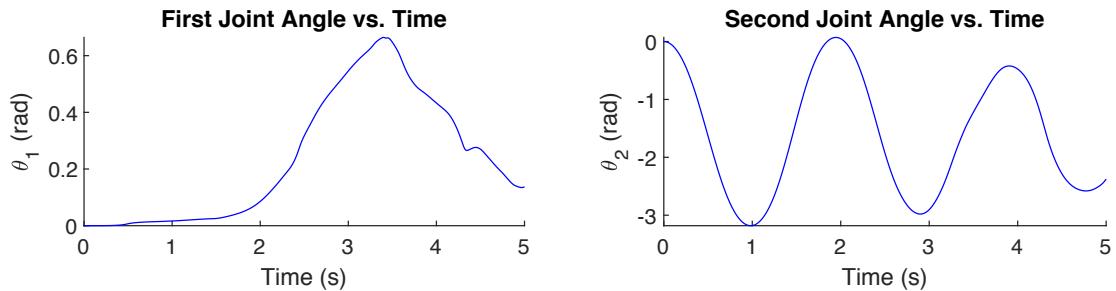


Figure 16: Joint Angles vs Time in Open-Loop Simulation

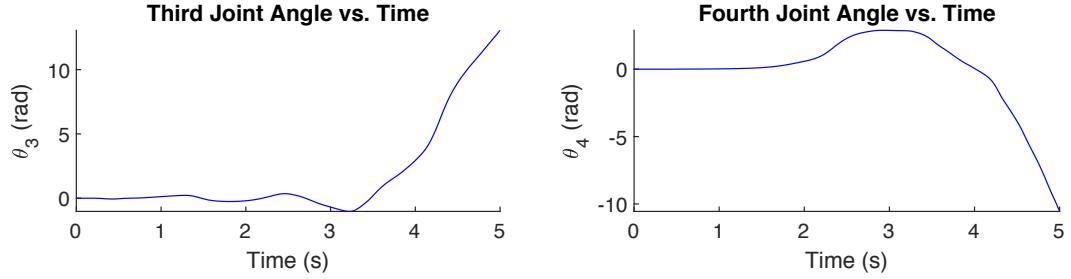


Figure 17: Joint Angles vs Time in Open-Loop Simulation

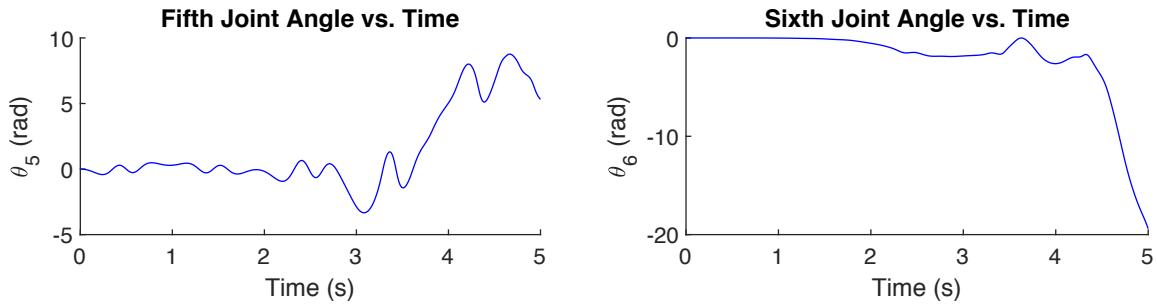


Figure 18: Joint Angles vs Time in Open-Loop Simulation

5.8 Closed-Loop Simulation

The motor equation (11) gives an expression for the motor torques, however the system dynamics are defined in terms of geometric joint angles. The inclusion of differential drive systems means that the joint angles, γ do not directly correspond to motor rotations, θ . However, there is a linear relation between them, described in equation (14).

$$\gamma = A\theta \quad \text{where} \quad A = \begin{bmatrix} 1/(2N) & 1/(2N) & 0 & 0 & 0 & 0 \\ 1/(2N) & -1/(2N) & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/N & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix} \quad (14)$$

Equation (14) can be used to map the joint angles to the motor angles. The gear ratio of 1:10 is represented by the variable N. Similarly, the motor angles can be determined by multiplying both sides of equation (14) by the inverse of matrix A, giving the following relation.

$$\theta = A^{-1}\gamma \quad (15)$$

It is important to note that the virtual work done by the joint torques (F_γ) and the virtual work done by the motor torques (F_θ) are equal. Using equation (15), a linear relation between

the joint torques and motor torques can be determined.

$$\begin{aligned}
\delta W &= F_\theta^T \delta \theta = F_\gamma^T \delta \gamma, \text{ where } \delta \gamma = A \delta \theta \\
F_\theta^T \delta \theta &= F_\gamma^T (A \delta \theta) \\
F_\theta^T &= F_\gamma^T A \\
(F_\theta^T)^T &= (F_\gamma^T A)^T \\
F_\theta &= A^T F_\gamma \Leftrightarrow F_\gamma = A^{-T} F_\theta
\end{aligned}$$

Using this equation, a relation can be determined between the motor dynamics and the system dynamics given in equation (11) and equation (1) respectively.

$$\begin{aligned}
H(\gamma) \ddot{\gamma} + d(\gamma, \dot{\gamma}) + G(\gamma) &= -A^{-T} (C_1 A^{-1} \ddot{\gamma} + C_2 A^{-1} \dot{\gamma} + C_3 \theta_d - C_3 A^{-1} \gamma) \\
\ddot{\gamma} &= H(\gamma)^{-1} (-A^{-T} (C_1 A^{-1} \ddot{\gamma} + C_2 A^{-1} \dot{\gamma} + C_3 \theta_d - C_3 A^{-1} \gamma) - d(\gamma, \dot{\gamma}) - G(\gamma)) \quad (16)
\end{aligned}$$

Because this equation includes the motor model, which in turn includes an internal PD controller, this equation can be integrated to solve for the system response given a desired motor angle input, θ_d . However, doing so will not result in the desired system response. This control scheme does not have any compensation for the inertia of the links, and it is also lacking gravity compensation. This can be remedied by modifying the input to the motors, θ_d . A new input, u , is defined such that gravity can be compensated. Thus, the motor input term in equation (16) must include both compensation for gravity and the desired motor angle.

$$\begin{aligned}
A^{-T} C_3 u &= G(\gamma) + d(\gamma, \dot{\gamma}) + A^{-T} C_3 \theta_d \\
u &= (A^{-T} C_3)^{-1} G(\gamma) + \theta_d \quad (17)
\end{aligned}$$

With this new motor input, the closed loop control system equations of motion are given as:

$$\ddot{\gamma} = H(\gamma)^{-1} (-A^{-T} (C_1 A^{-1} \ddot{\gamma} + C_2 A^{-1} \dot{\gamma} + C_3 u - C_3 A^{-1} \gamma) - d(\gamma, \dot{\gamma}) - G(\gamma)) \quad (18)$$

Equation (18) can then be integrated to solve for the system response given desired motor angles.

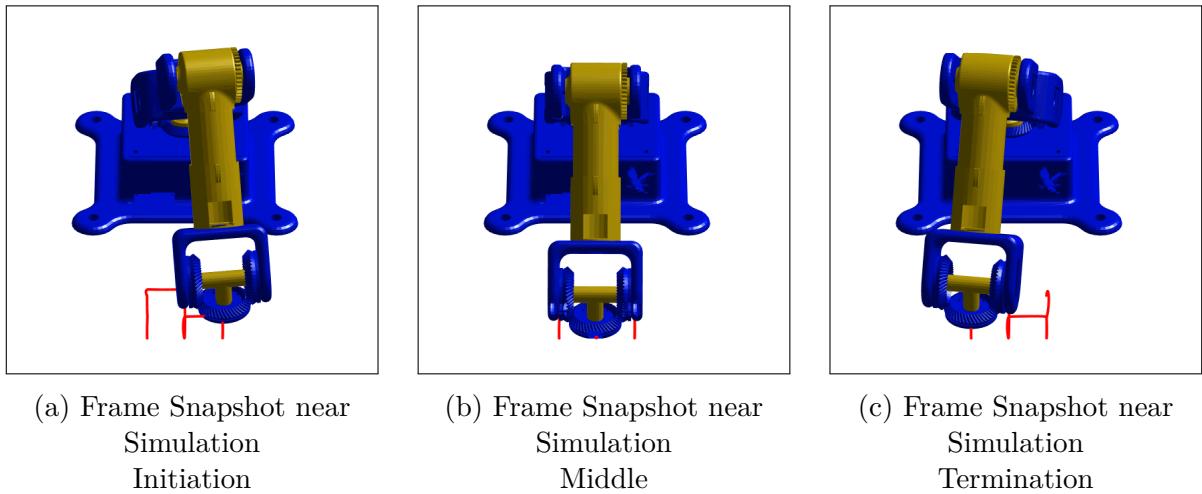


Figure 19: Closed-Loop Control Simulation Animation Snapshots

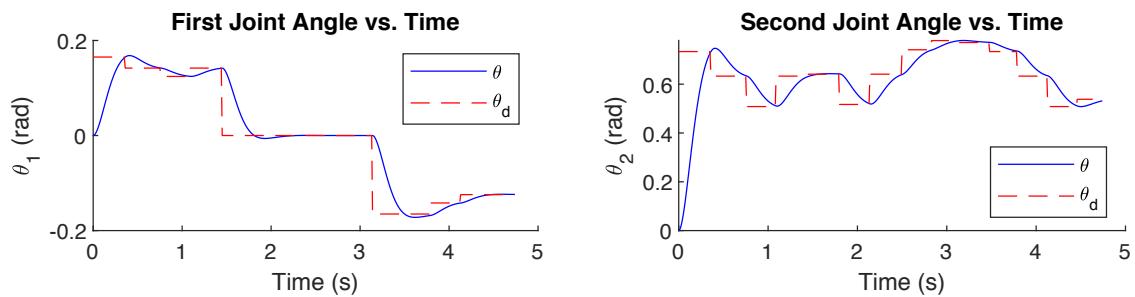


Figure 20: Joint Angles vs Time in Closed-Loop Simulation

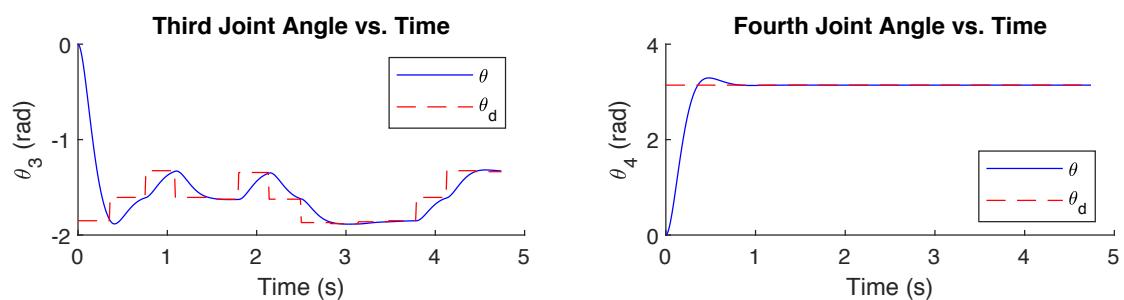


Figure 21: Joint Angles vs Time in Closed-Loop Simulation

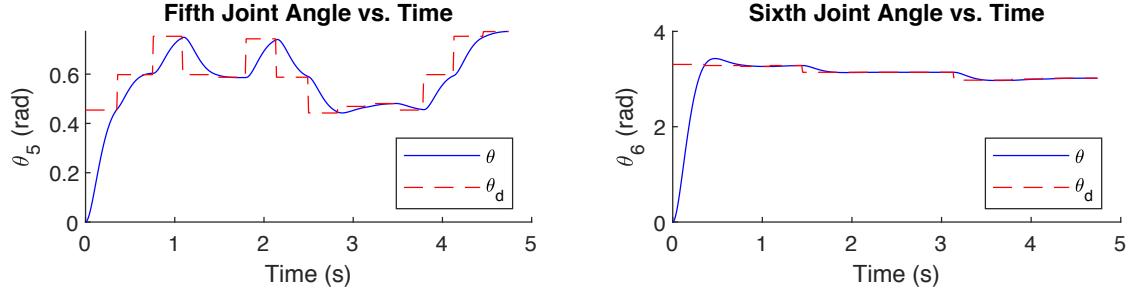


Figure 22: Joint Angles vs Time in Closed-Loop Simulation

5.9 ANSYS

With 100% infill, 3D printed PLA has a maximum shear stress of 13.6 kpsi. The manipulator applies a load of 13N in the negative y-direction. Without gears in the base differential, the differential support would bear the load on its bearing mounts. *Figure 23* shows the manipulator's differential support could experience up to 97 kPa or 14 kpsi of shear stress, which is less than the maximum shear stress of PLA with 100% infill. Since some of the manipulator's mass is supported by the gears, the actual shear experienced by the differential support will be less.

Further ANSYS simulations on repetitive loadings are in progress.

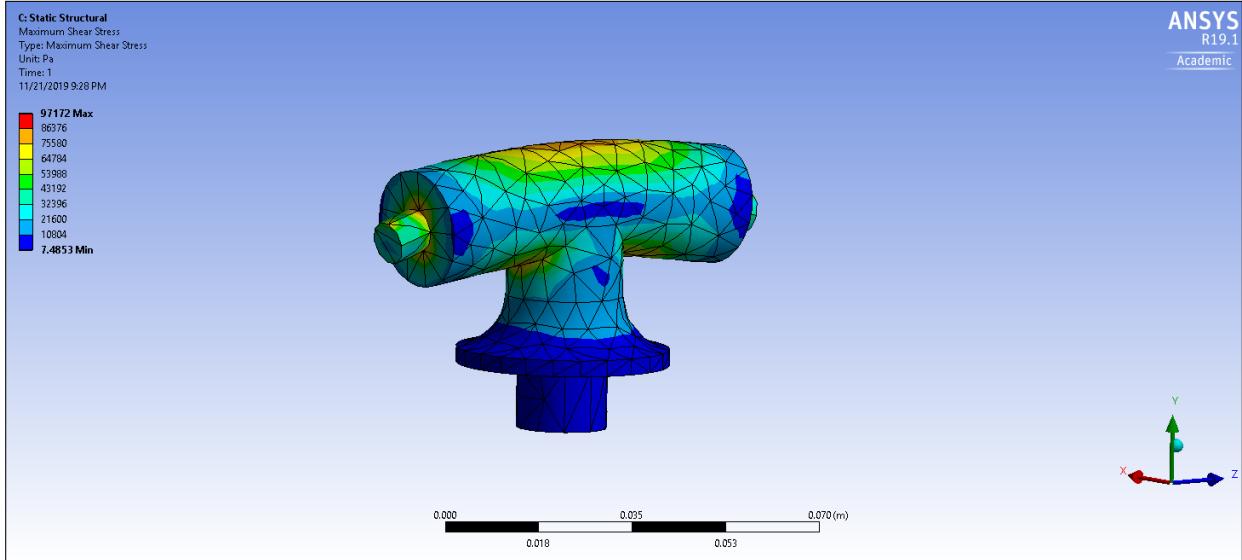


Figure 23: T-Bar ANSYS FEA

5.10 Electrical Schematic

The electrical schematic shown in *Figure 24* consists of a power supply that receives 120V AC that is standardly supplied from wall outlets. The Power supply has two outputs, a 5V terminal that has a max current output of 3A, and a 12V terminal that has a max output of 7A. The Raspberry Pi 3B is powered by the 5V terminal, and since the max current draw from the Pi is 1.2A the maximum current draw from the terminal is not surpassed. The six MX-12W servos used in the manipulator, as well as the AX-12A servo used in the end effector, are connected in parallel across the 12V terminal from the power supply. Since the MX-12W has a stall current of .6A and the AX-12A has a stall current of 1.5A, the maximum current to be pulled from the 12V terminal would be 5.1A, less than the 7A the supply can output. In order to communicate to the servos, a U2D2 communication converter is connected to the Raspberry Pi which creates a serial connection used by the servos, allowing the servos to be daisy-chained together. The Raspberry Pi also connects to an external PC so that the user can communicate with the Pi using a mouse, keyboard, and monitor.

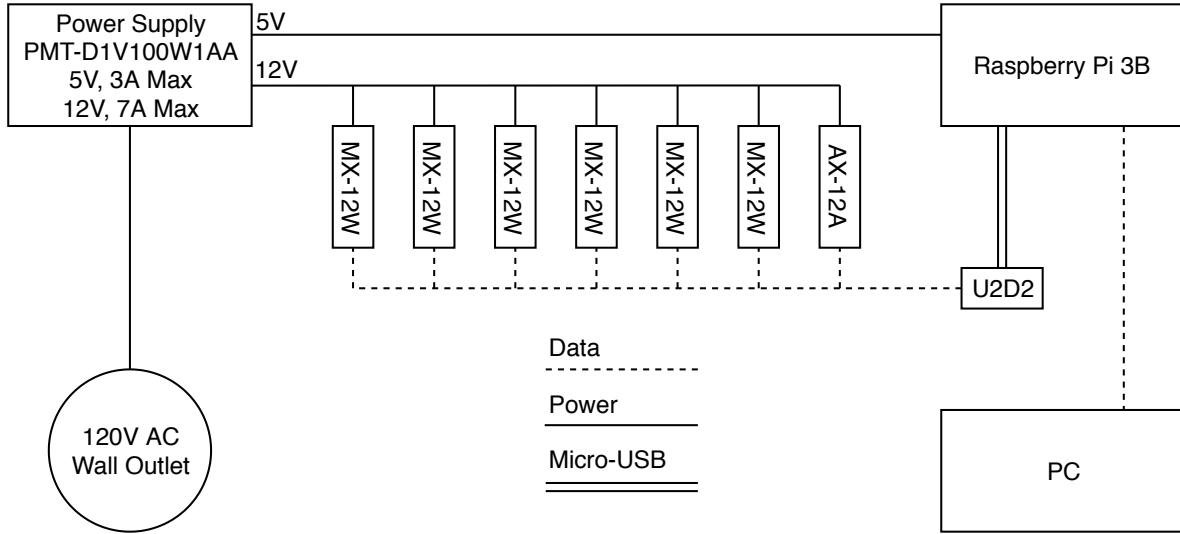


Figure 24: Electrical Schematic

5.11 Software Flowchart

The software the system uses takes in a row vector of position, orientation, path type, and end effector function information for all points to be traveled through and runs the manipulator through the desired points following the specified paths requested. The general overview of the code is shown in *Figure 25*. *Figure 25* shows that the software for the manipulator is

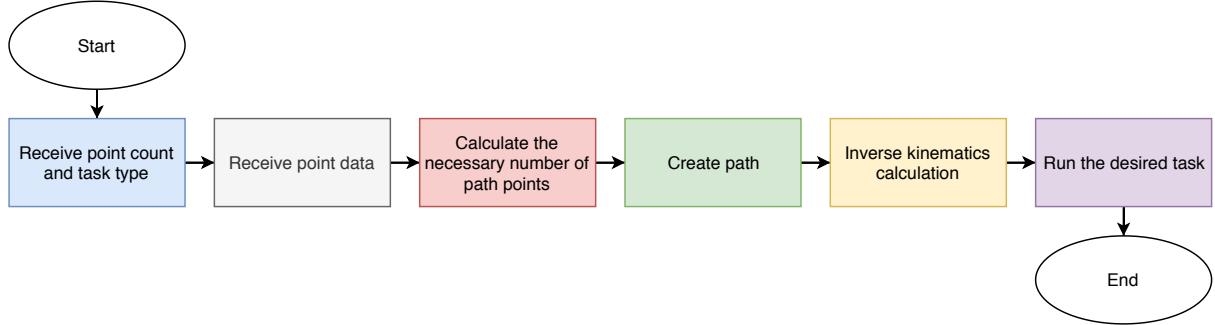


Figure 25: General Software Algorithm Overview

broken up into six subsections, two sections that receive data, three that do calculations, and one that runs the specified task.

The first subsection of the software works to receive the number of points the user is inputting as well as the general task the user is completing, shown in *Figure 26*. *Figure 26* specifies

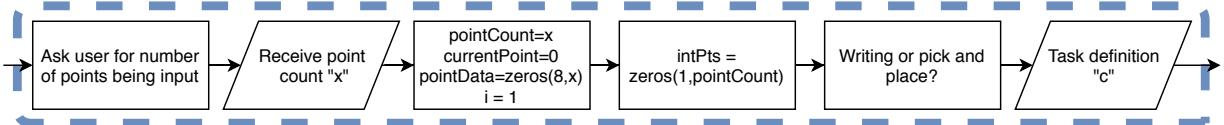


Figure 26: Software Flowchart Subsection 1

that the software prompts the user for the number of points that the manipulator will travel through and stores the input as a variable, in this case 'x'. The 'x' variable is only used to help preallocate data vectors so that the size of the vector does not change with each input. The software also receives the task specification as either a 0 for cartesian straight line pathing or a 1 for a straight line in the joint space and stores this value in the variable 'c'.

The second general block in the software flowchart works to receive and store the necessary data for the points the user is inputting depending on the path type as seen in *Figure 27*.

Figure 27 shows that the path type variable 'c' is used to determine what information is necessary to collect. If the user is doing a writing task, the software only collects the x, y,

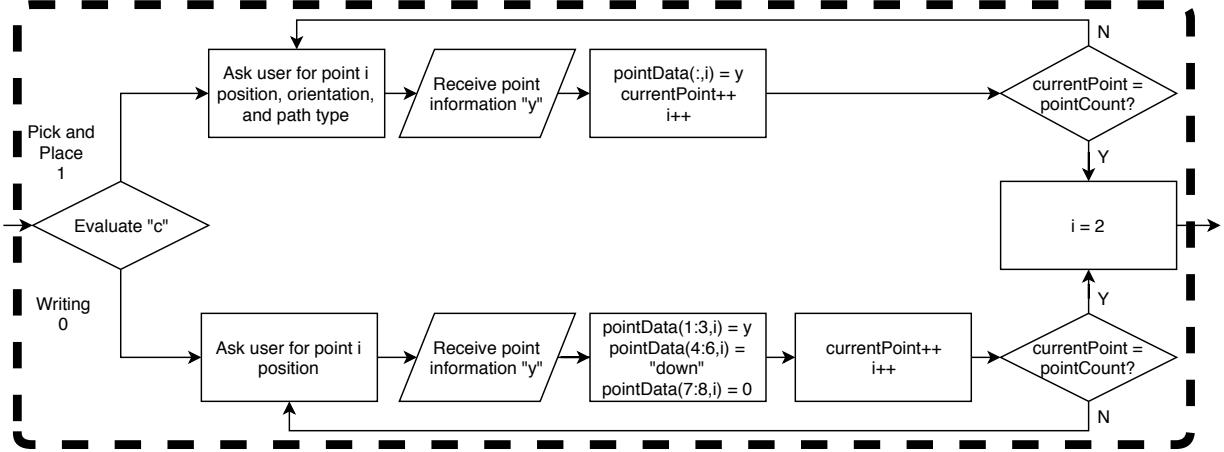


Figure 27: Software Flowchart Subsection 2

and z distances for the point and assumes that the end effector orientation will be facing down so that the marker is vertical. If the user is doing pick and place, the software prompts the user for the x, y, z, phi, theta, psi, path type, and end effector data. The software loops until all the points have been input.

The next block in the software flowchart calculates the total points necessary to complete the task. The overview for this section can be seen in *Figure 28*. As seen in *Figure 28*, the

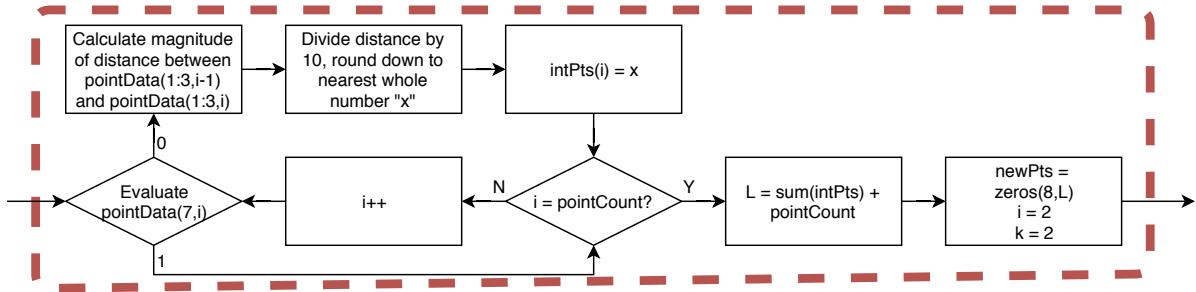


Figure 28: Software Flowchart Subsection 3

section of software calculates the distance between the current point and the previous point if the path type is cartesian straight line and divides the distance by ten to find the number of centimeters between the two points. This value is stored as the necessary number of intermediate points, and the software will loop through until every point has been checked. The section of code also stores the total number of points that will be used as the variable level for later use.

The fourth code block in the flowchart creates and stores the necessary intermediate points along the desired path, shown in *Figure 29*. The code shown in *Figure 29* creates points every centimeter if the path type is cartesian straight line using the number of path points

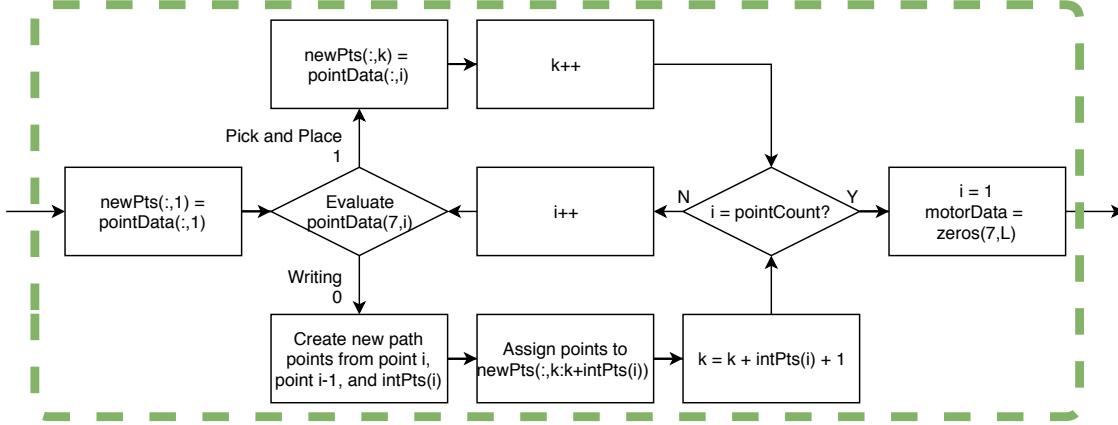


Figure 29: Software Flowchart Subsection 4

stored for each point from the previous block of software. This ensures that a straight line will be followed between the two user input points. If the path type is a straight line in the joint space, the software does not add any intermediate points since the path seen in the cartesian space does not matter.

The fifth code block in the flowchart calculates inverse kinematics of the points defined in the previous block of code and stores the angles as counts that can be used by the servos. The overview of this section can be seen in *Figure 30*. *Figure 30* shows that the new points

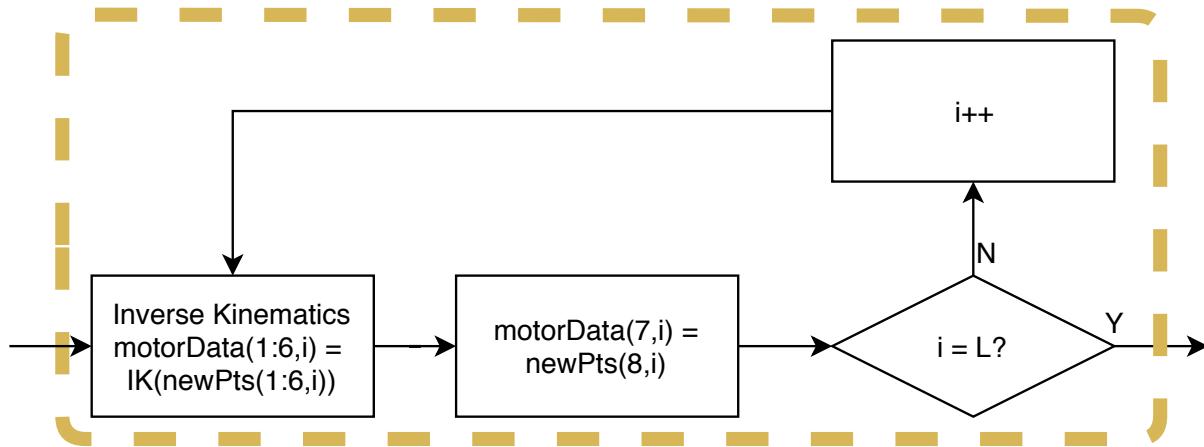


Figure 30: Software Flowchart Subsection 5

found in the prior section of code are run through an inverse kinematics function that will output the necessary counts the servos can utilize. The code iterates through each point until the inverse kinematics have been calculated for all points.

The final block in the software diagram runs the manipulator through the desired task, with this section of code requiring user input at certain stages depending on the path type, seen

in *Figure 31*. The code in *Figure 31* prompts the user to press space to close the end effector

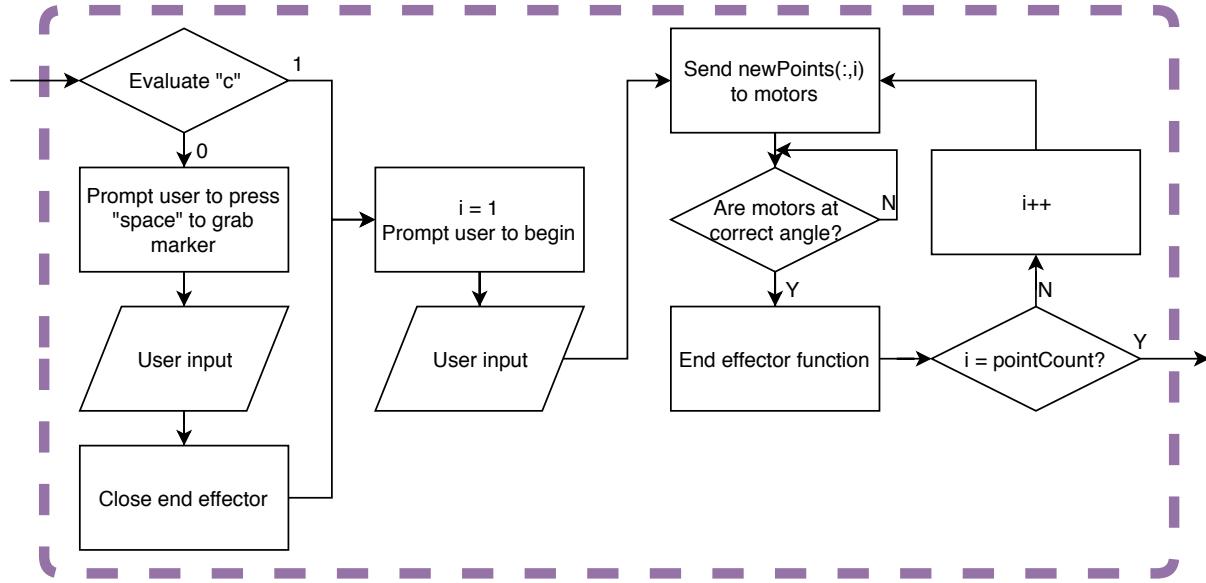


Figure 31: Software Flowchart Subsection 6

and grab the marker if drawing was the specified task, otherwise the software jumps straight into prompting the user to begin the task, and when the user begins the task the counts for each position are sent to the servos one at a time. The counts for the next position are not sent to the servos until the servos have reached the desired positions and the end effector function has been completed if there is one.

5.12 Project Status and Future

- Print Differential
- Cut Aluminum Supports
- Print Base
- Print Link 1
- Test Link 1 and Differential
- Print Differential Support
- Test Differential
- Test Link 1 Pulleys
- Test Link 1 with Motors
- Test Motor Control of Differential
- Test Link 1 With Load
- Implement Kinematics in Python
- Test Kinemtics with Motors

The Gantt chart can be seen on the following page.

5.13 Parts List

Table 3: Bill of Materials with Costs

Notes	Retailer	Quantity	Unit Cost (USD)	Total Cost (USD)
120 tooth		3	31	93
12 tooth	Automation Direct	3	8.5	25.5
3 pack, 300 tooth		1	11.5	11.5
3 pack, 208 tooth		1	9.5	28.5
MX-12W		6	65.9	395.4
U2D2	Trossen Robotics	1	49.9	49.9
500 mm, 1/2 pulleys		2	3.95	7.9
350 mm, 3 pulley		1	2.95	2.95
Pi 3 B	Amazon	1	37.99	37.99
Bearings		1	8.99	8.99
2 Sch 10 12" Al tube	Industrial Metal Sales	1	2.99	2.99
12 V, 5 V power supply	Digi-Key Electronics	1	43.21	43.21
Automation Direct				0
Trossen Robotics				13.15
Amazon	Shipping	N/A	N/A	0
Industrial Metal Sales				26.36
Digi-Key				8.99
			Total	756.33

Acknowledgements & Attributions

We would like to acknowledge the following people for their contributions in creating this report.

— Dr. Isenberg

— Dr. Schipper

References

- [1] S. Hutchinson M. Spong and M. Vidyasager. *Robot Modeling and Control*. 2006.
- [2] ROBOTIS. Ax-12a e-manual, 2019.

A Appendix

A.I Relevant Figures and Materials

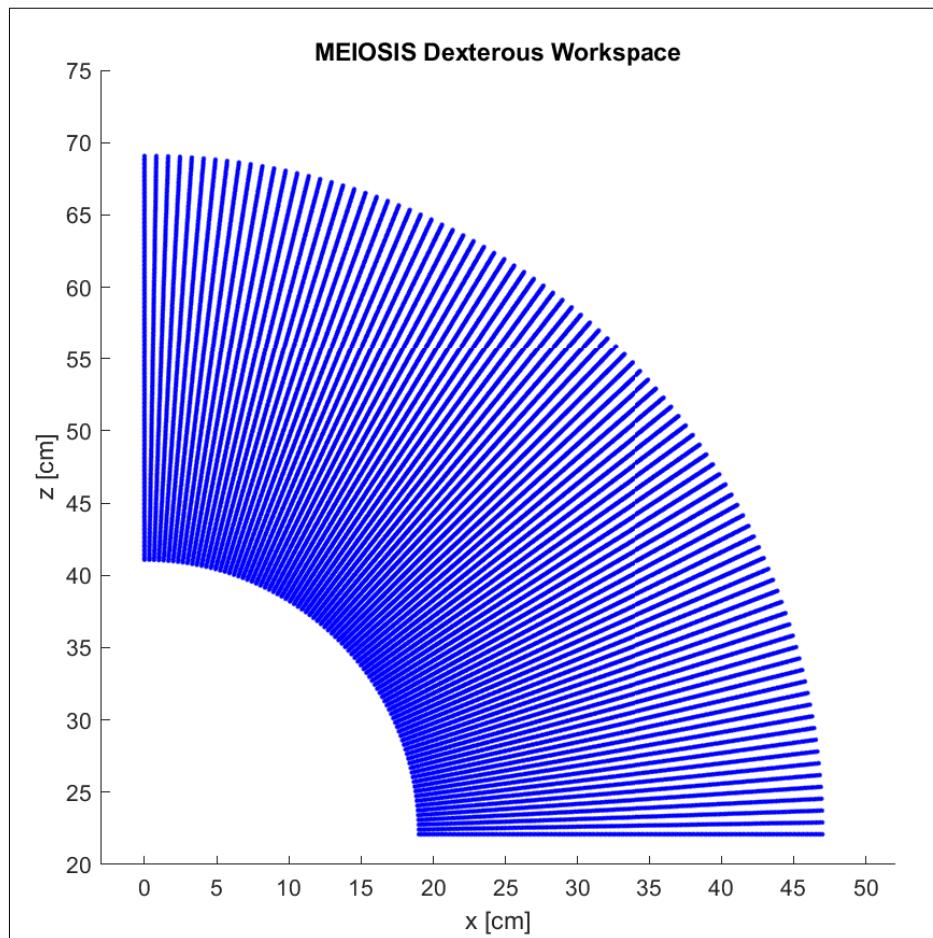


Figure 32: Cross Section of Dexterous Workspace Quadrant

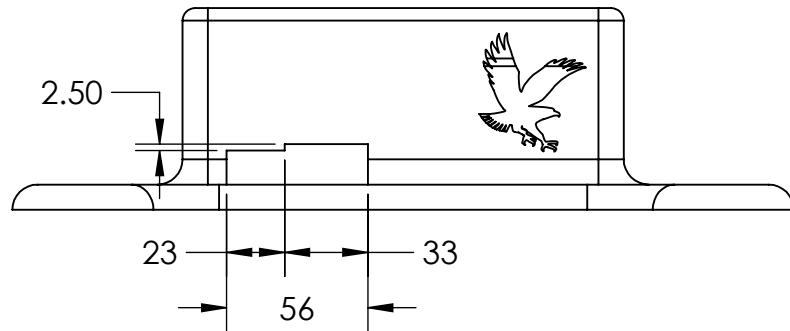
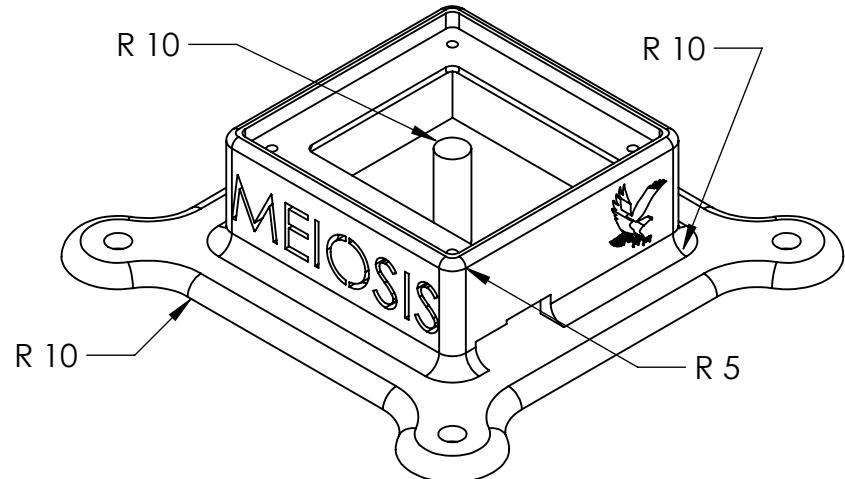
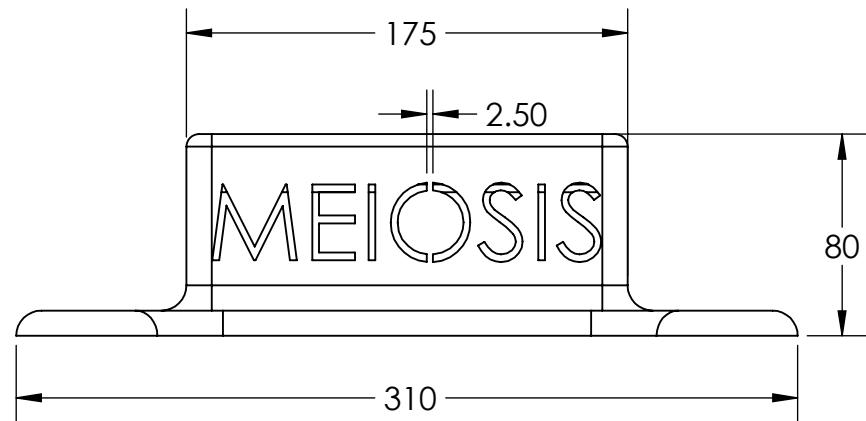
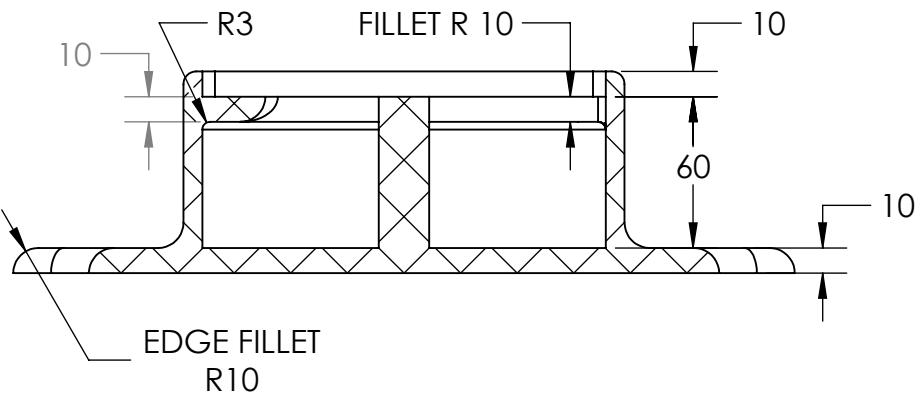
A.II CAD Drawings

The complete drawing package is attached.

2

1

B



A

A

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN
MILLIMETERS
TOLERANCES: ± 3.0

INTERPRET GEOMETRIC
TOLERANCING PER:
MATERIAL
PLA, 30% Infill
FINISH
N/A
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	Ryan W.	11/13/2019
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		
COMMENTS:		

TITLE:

Base Bottom

SIZE	DWG. NO.	REV
A		1.0
SCALE: 1:5	WEIGHT:	SHEET 1 OF 13

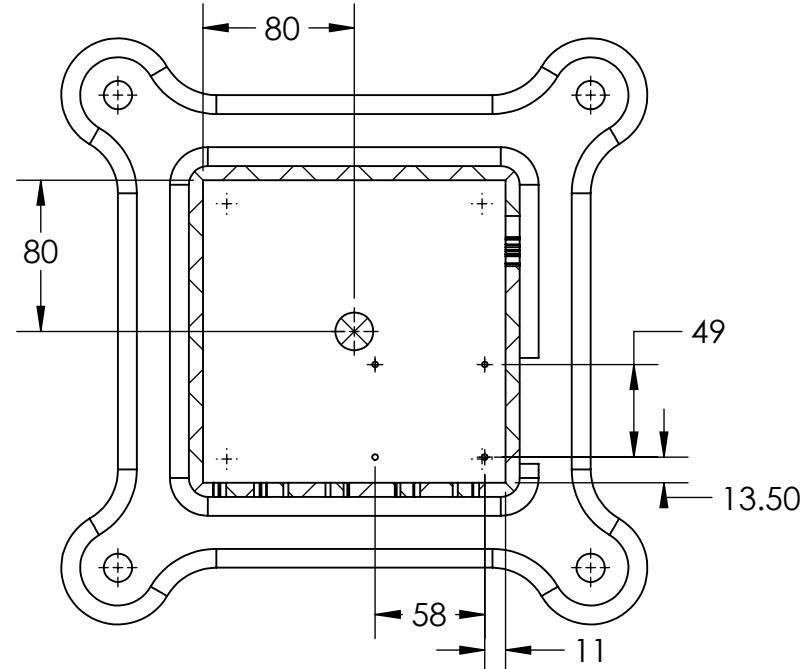
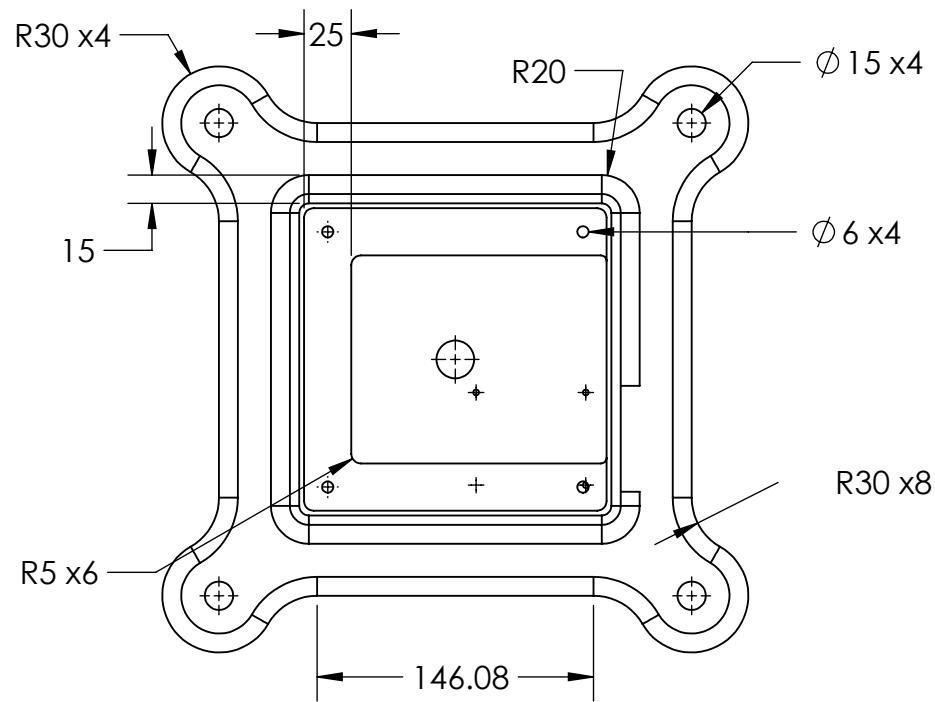
2

1

2

1

B



B

A

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
TOLERANCES: ± 3.0

INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL PLA, 30% Infill
FINISH N/A
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	Ryan W.	11/13/2019
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

Base Bottom

SIZE	DWG. NO.	REV
A		1.0
SCALE: 1:5		WEIGHT:
SHEET 2 OF 13		

2

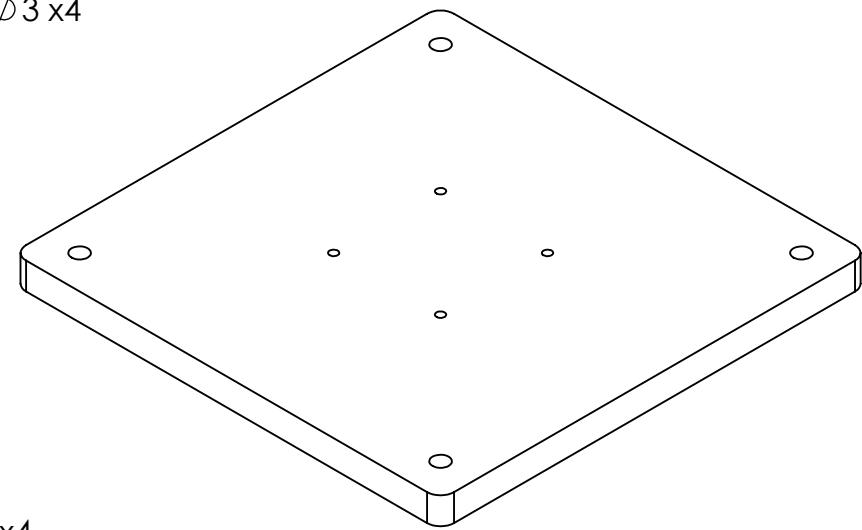
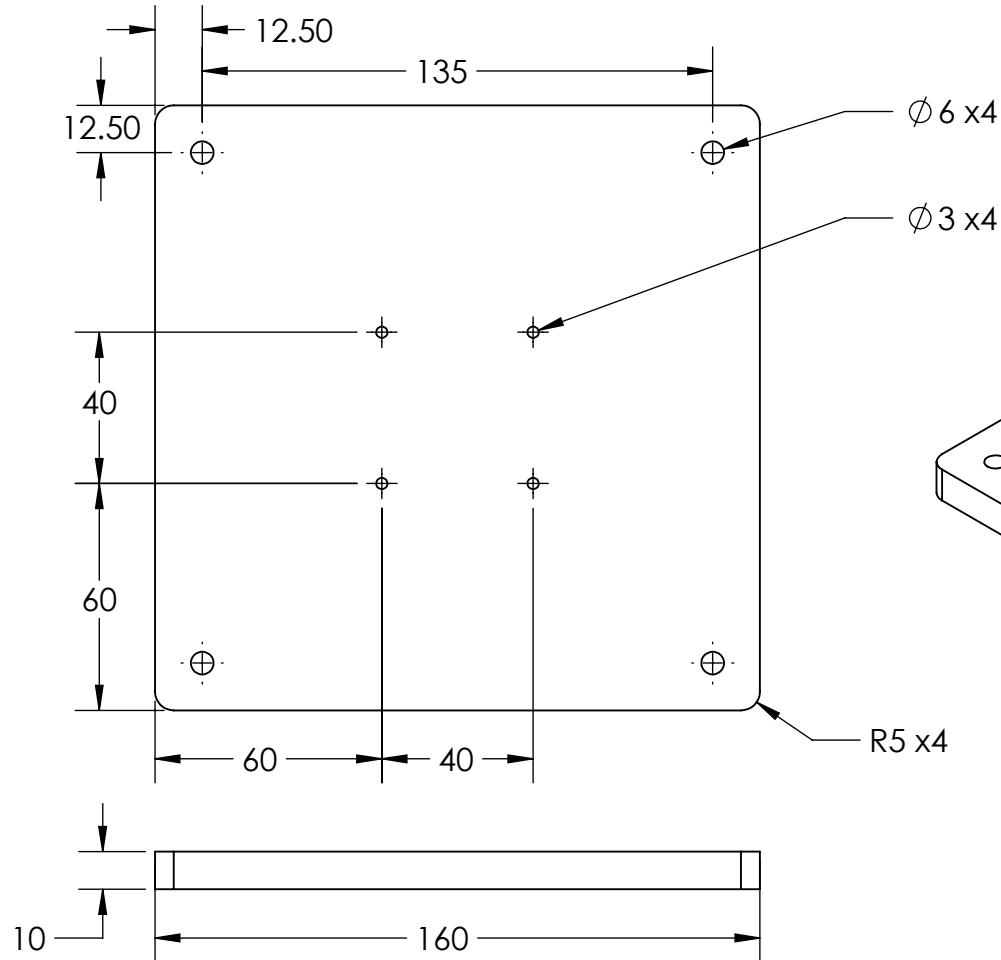
1

2

1

B

B



A

A

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN
MILLIMETERS
TOLERANCES: ± 3.0

INTERPRET GEOMETRIC
TOLERANCING PER:
MATERIAL
PLA, 30% Infill
FINISH
N/A
DO NOT SCALE DRAWING

DRAWN	NAME	DATE
CHECKED	Ryan W.	11/8/2019
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

Base Top

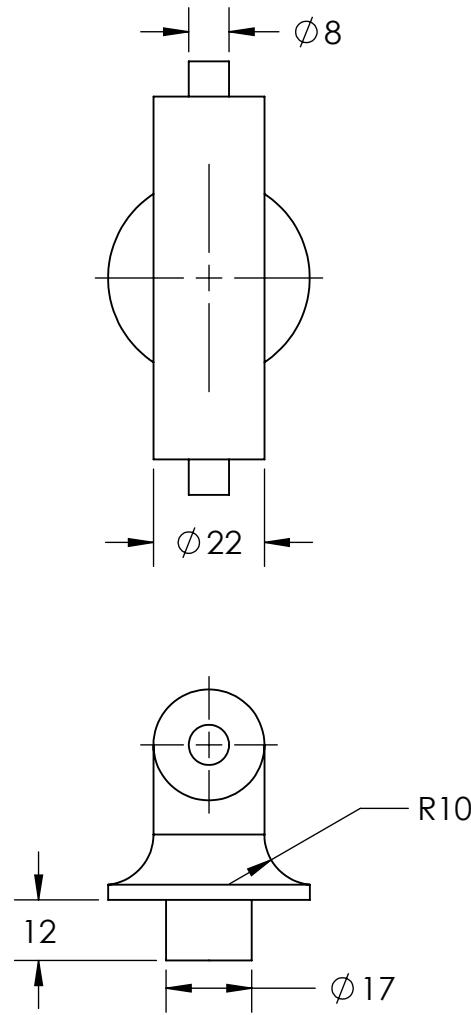
SIZE	DWG. NO.	REV
A		1.0
SCALE: 1:5	WEIGHT:	SHEET 3 OF 13

2

1

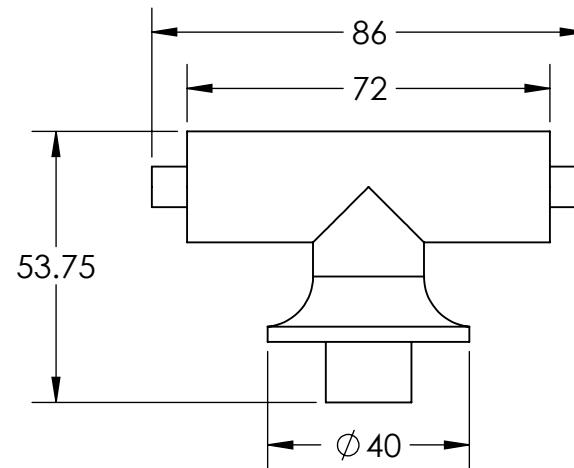
2

2



1

1



A

B

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN
MILLIMETERSTOLERANCES: ± 3.0 INTERPRET GEOMETRIC
TOLERANCING PER:MATERIAL
PLA, 30% InfillFINISH
N/A

DO NOT SCALE DRAWING

NAME DATE

DRAWN Ryan W. 11/8/2019

CHECKED

ENG APPR.

MFG APPR.

Q.A.

COMMENTS:

TITLE:

Shoulder Differential
T-Bar

SIZE DWG. NO.

A

REV

1.0

SCALE: 1:2 WEIGHT:

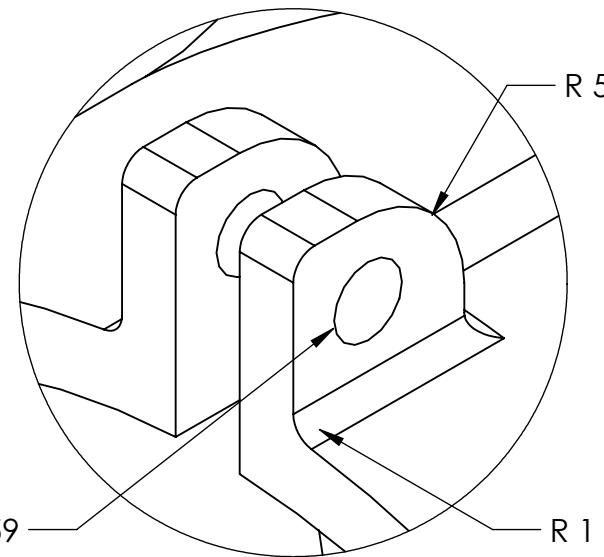
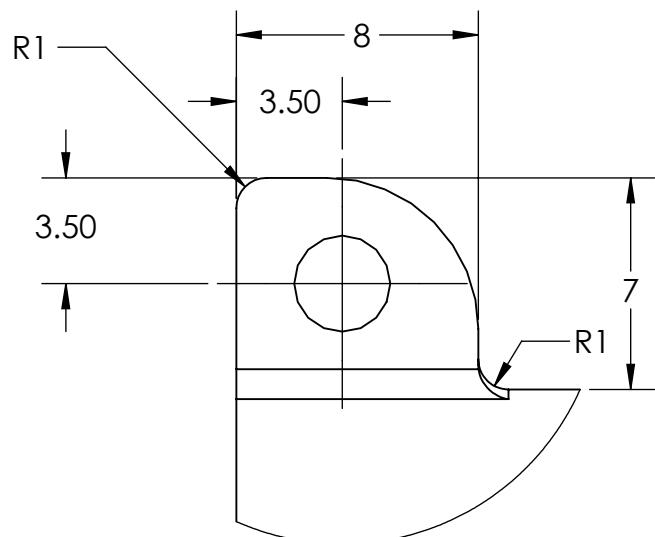
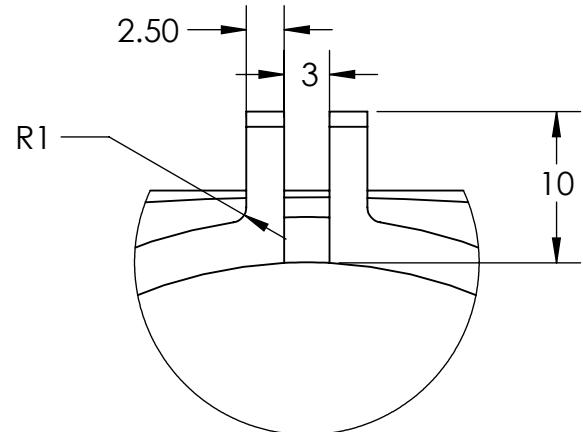
SHEET 4 OF 13

2

1

2

1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN
MILLIMETERS
TOLERANCES: ± 3.0

INTERPRET GEOMETRIC
TOLERANCING PER:
MATERIAL PLA
FINISH N/A

DO NOT SCALE DRAWING

NAME	DATE
DRAWN	Ryan W. 11/20/2019
CHECKED	
ENG APPR.	
MFG APPR.	
Q.A.	

COMMENTS:
THIS DRAWING SHOWS
THE CLAMPING
MECHANISM TO BE USED
FOR LINK 2_A, LINK 2_B,
LINK 3_A, AND LINK 3_B.

TITLE:

CLAMP DESIGN

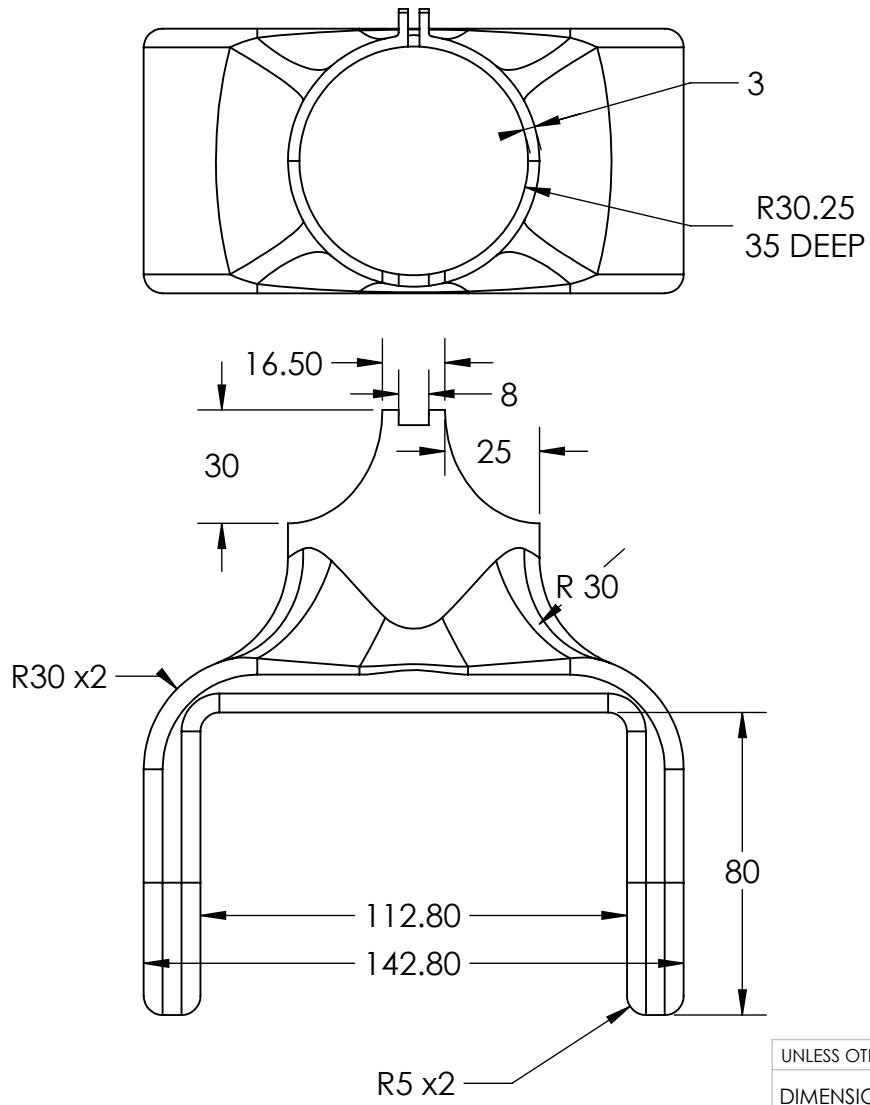
SIZE	DWG. NO.	REV
A		1.0
SCALE: 4:1	WEIGHT:	SHEET 5 OF 13

2

1

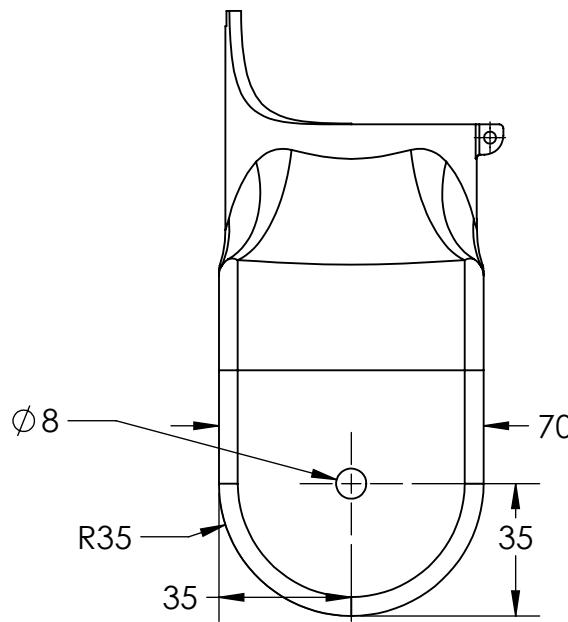
B

B



R 11

SCALE 1:4



A

A

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN
MILLIMETERSTOLERANCES: ± 3.0 INTERPRET GEOMETRIC
TOLERANCING PER:MATERIAL
PLA, 30% InfillFINISH
N/A

DO NOT SCALE DRAWING

NAME DATE

DRAWN Ryan W. 11/11/2019

CHECKED

ENG APPR.

MFG APPR.

Q.A.

COMMENTS:

TITLE:

Link 2_a

SIZE DWG. NO. REV
A 1.0

SCALE: 1:2 WEIGHT: SHEET 6 OF 13

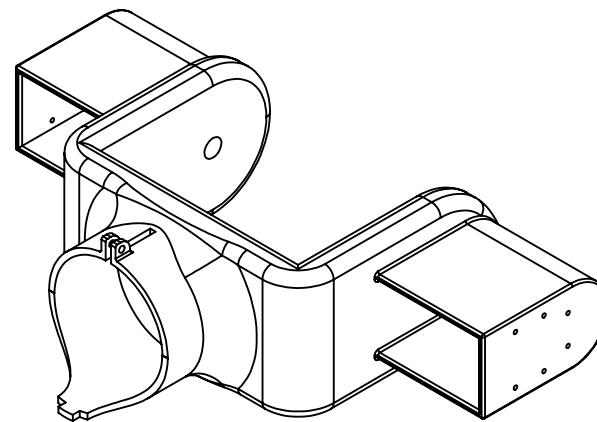
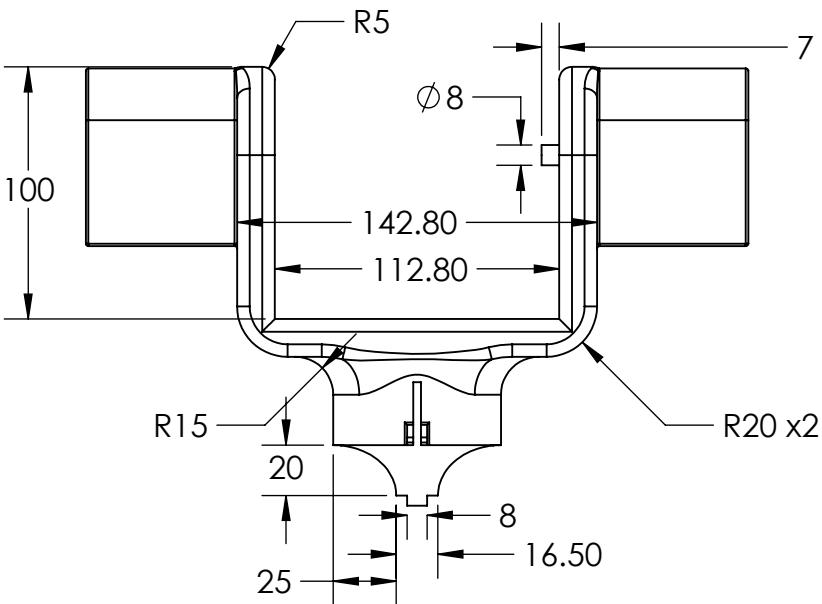
2

1

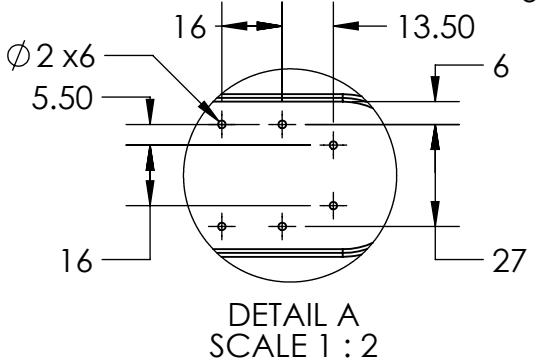
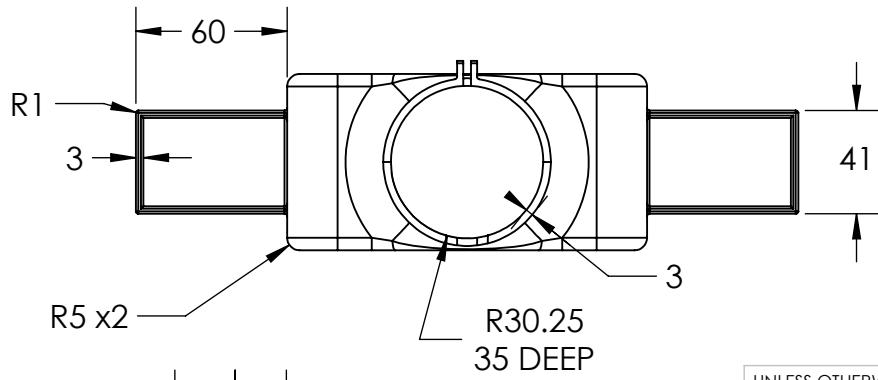
2

1

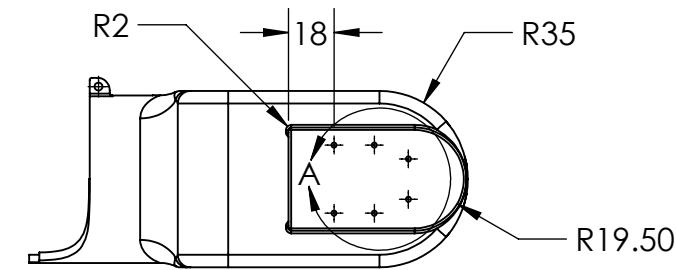
B



B



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
TOLERANCES: ± 3.0
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL PLA, 30% Infill
FINISH N/A
DO NOT SCALE DRAWING



TITLE:

Link 2_b

SIZE	DWG. NO.	REV
A		1.0
SCALE: 1:3	WEIGHT:	SHEET 7 OF 13

2

1

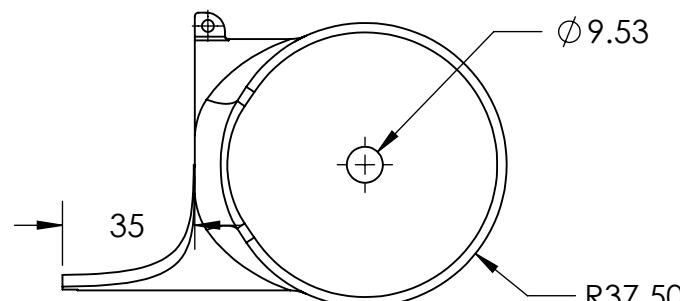
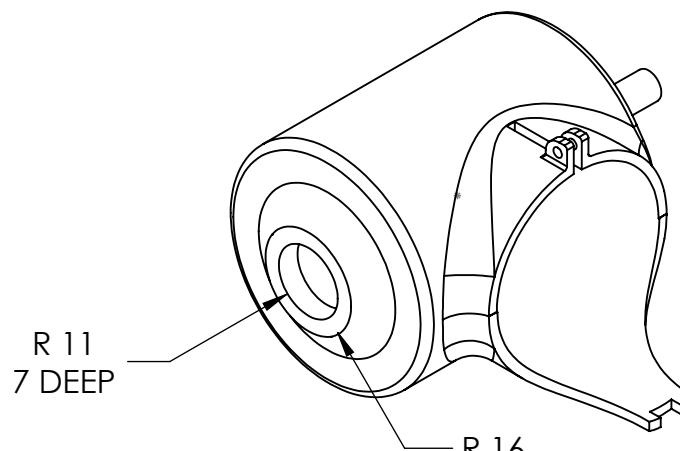
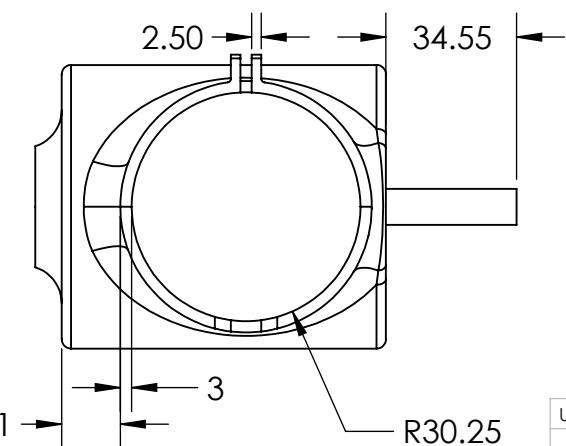
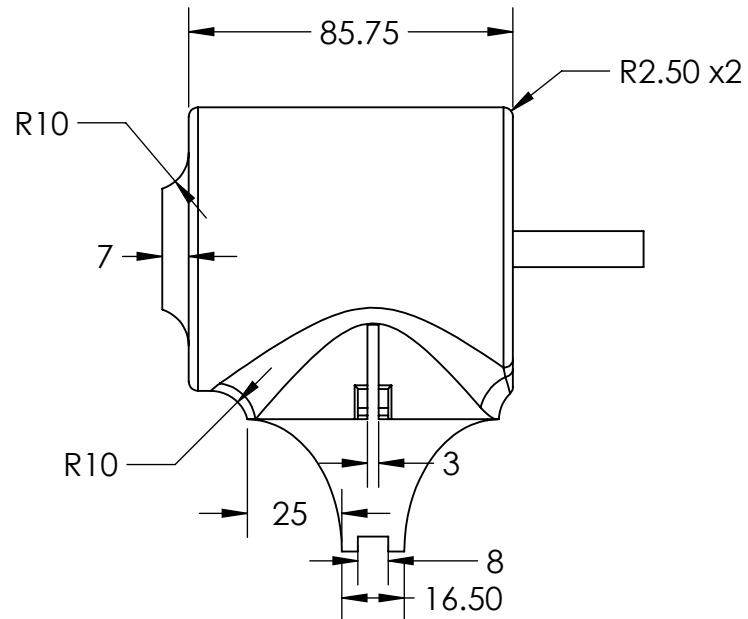
A

2

1

2

1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN
MILLIMETERS
TOLERANCES: ± 3.0
INTERPRET GEOMETRIC
TOLERANCING PER:
MATERIAL
PLA, 30% Infill
FINISH
N/A
DO NOT SCALE DRAWING

DRAWN	NAME	DATE
CHECKED	Ryan W.	11/11/2019
ENG APPR.		
MFG APPR.		
Q.A.		
COMMENTS:		

TITLE:

Link 3_a

SIZE	DWG. NO.	REV
A		1.0
SCALE: 1:2	WEIGHT:	SHEET 8 OF 13

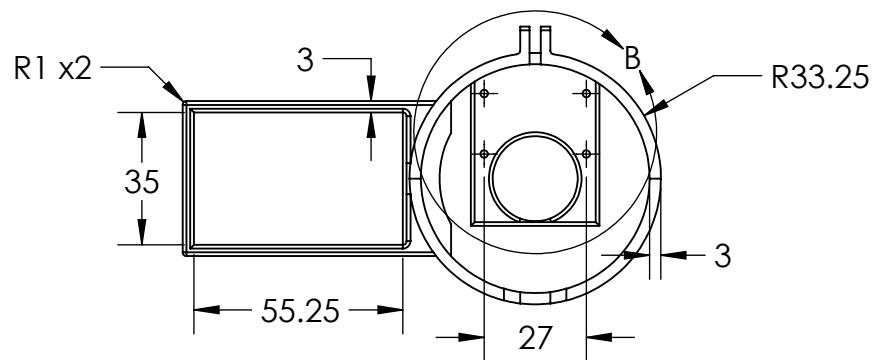
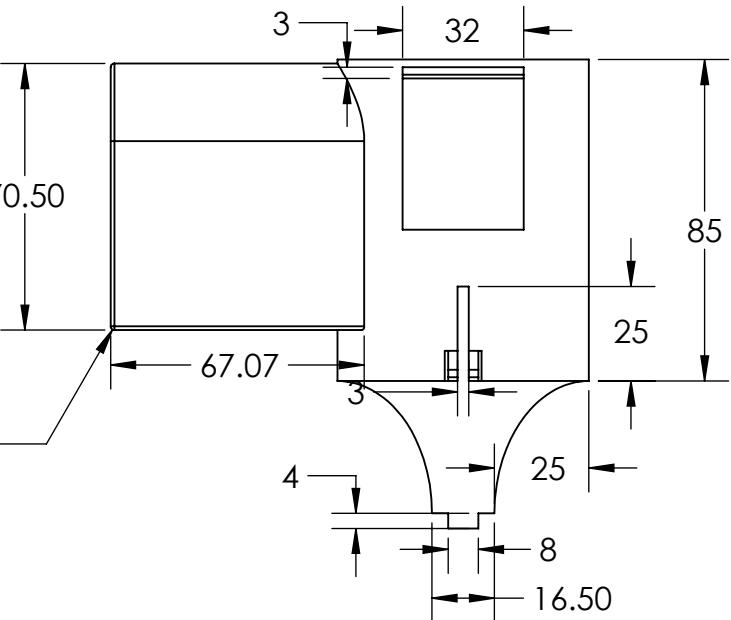
B

A

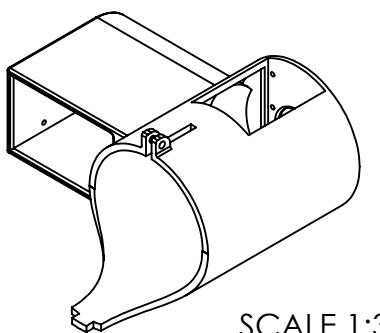
2

1

B



A

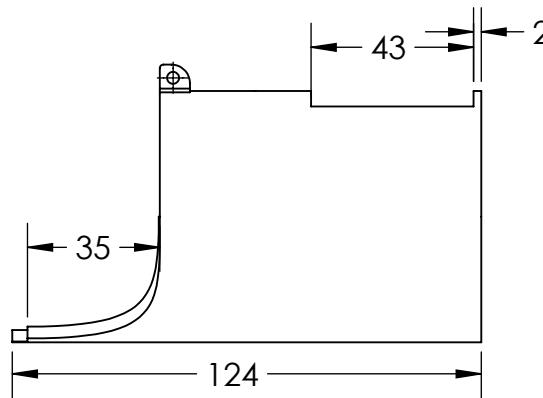
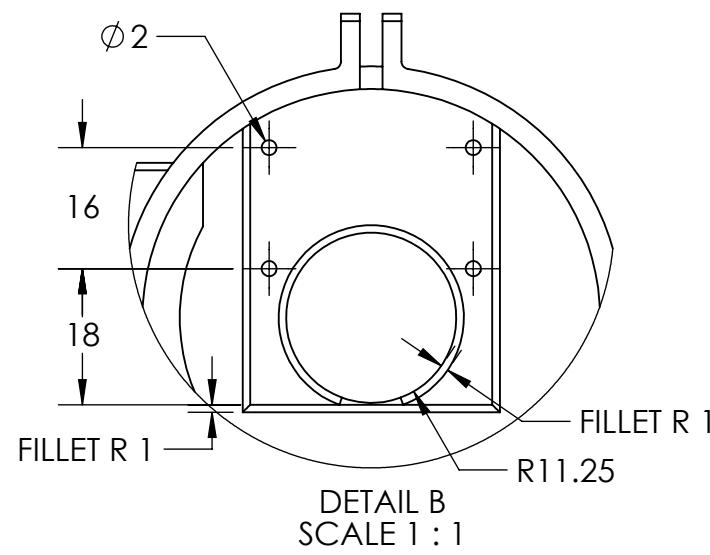


UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN
MILLIMETERS
TOLERANCES: ± 3.0

INTERPRET GEOMETRIC
TOLERANCING PER:
MATERIAL PLA, 30% Infill
FINISH N/A
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	Ryan W.	11/15/2019
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		
COMMENTS:		

SIZE	DWG. NO.	REV
A		1.0
SCALE: 1:2	WEIGHT:	SHEET 9 OF 13



B

A

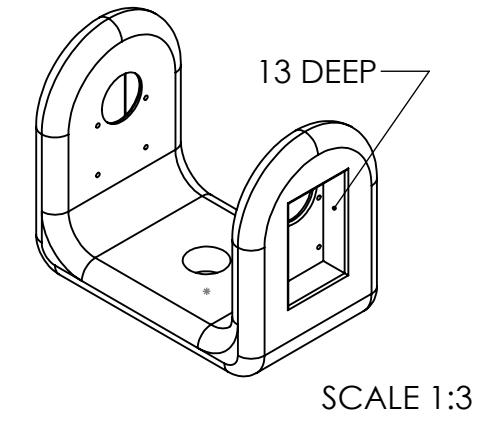
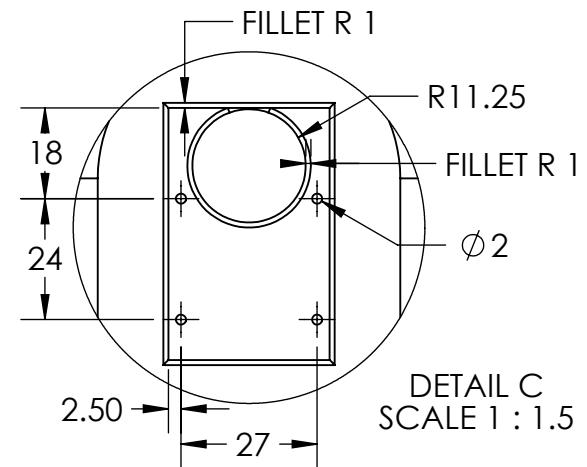
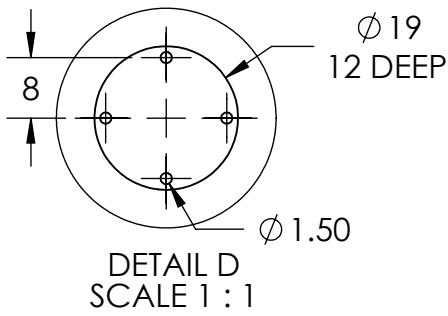
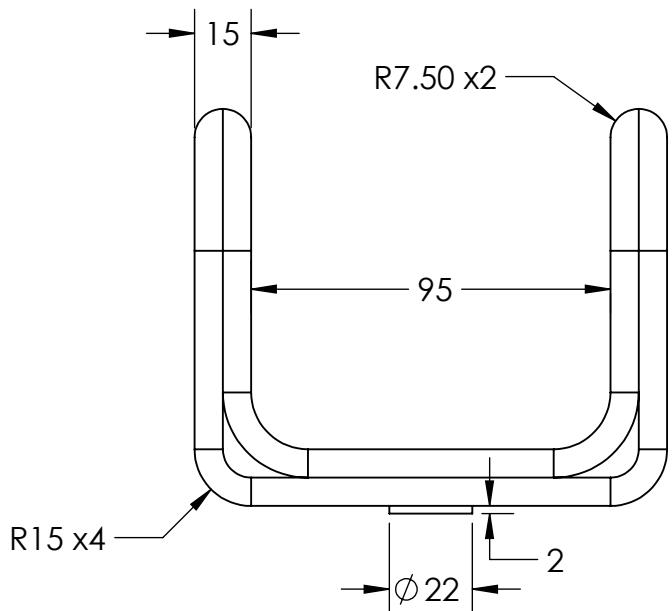
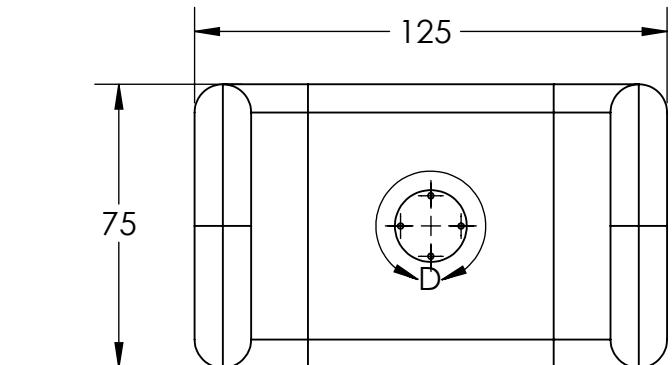
2

1

2

1

B



UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN MILLIMETERS		DRAWN	Ryan W. 11/11/2019
TOLERANCES: ± 3.0		CHECKED	
		ENG APPR.	
		MFG APPR.	
INTERPRET GEOMETRIC TOLERANCING PER:		Q.A.	
MATERIAL	PLA, 30% Infill	COMMENTS:	
FINISH	N/A		
DO NOT SCALE DRAWING			

TITLE:

Link 4

SIZE	DWG. NO.	REV
A		1.0
SCALE: 1:2	WEIGHT:	SHEET 10 OF 13

2

1

B

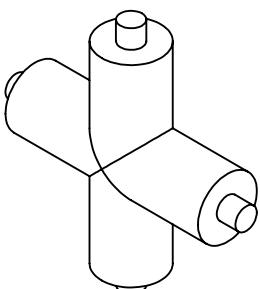
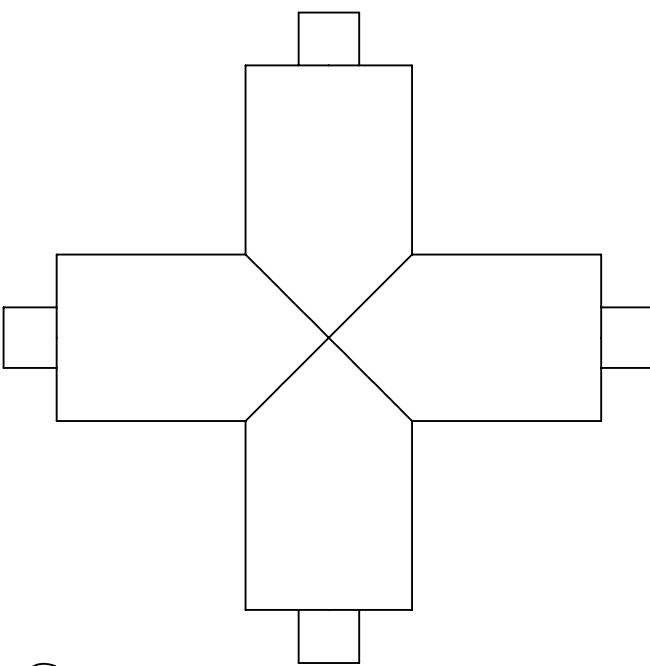
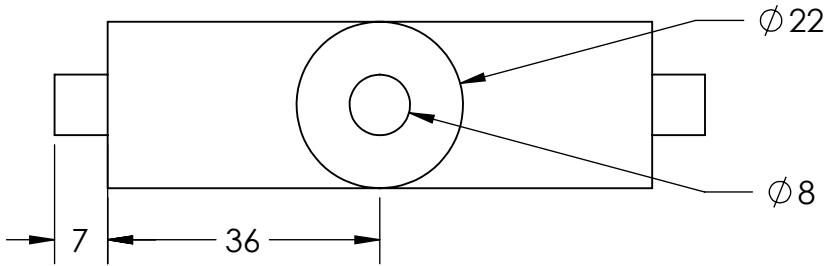
A

2

1

B

B

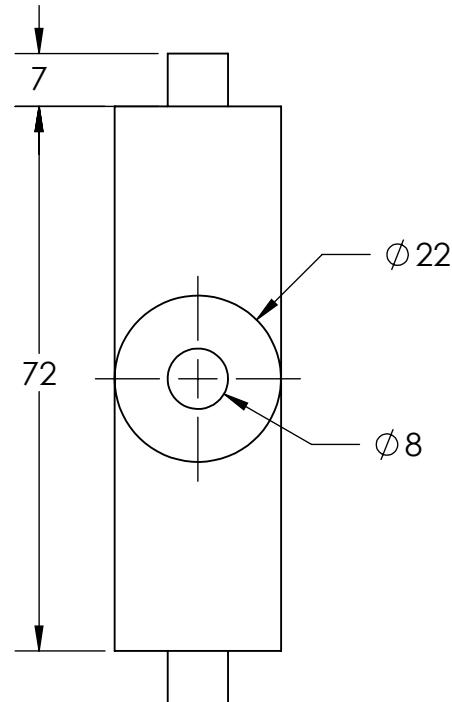


SCALE 1:2

2

1

UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN MILLIMETERS	Ryan W.	11/8/2019	
TOLERANCES: ±3.0			
INTERPRET GEOMETRIC TOLERANCING PER:			
MATERIAL			
PLA, 30% Infill			
FINISH			
N/A			
DO NOT SCALE DRAWING			



TITLE:

Wrist Differential Crossbar

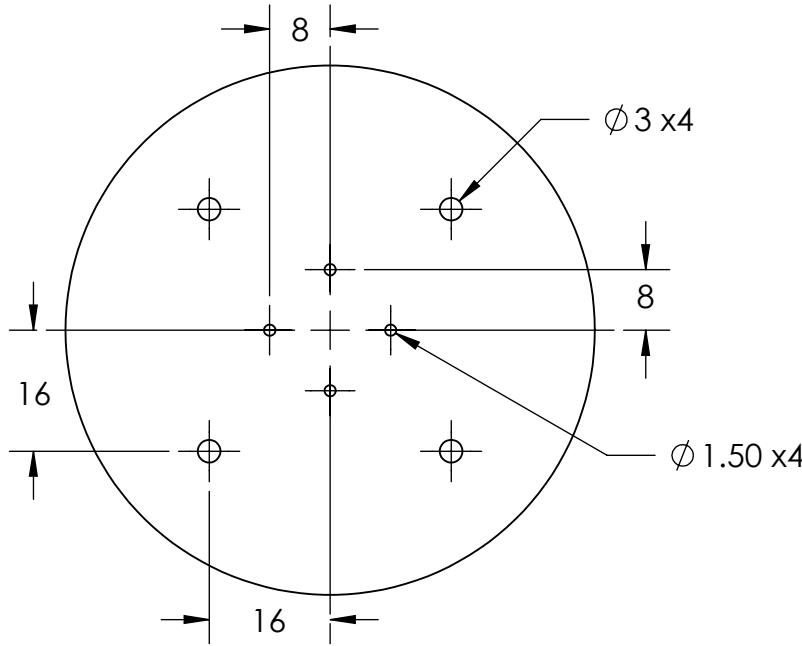
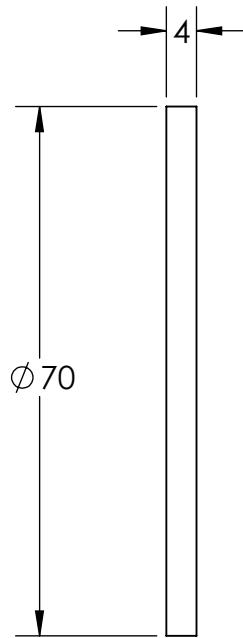
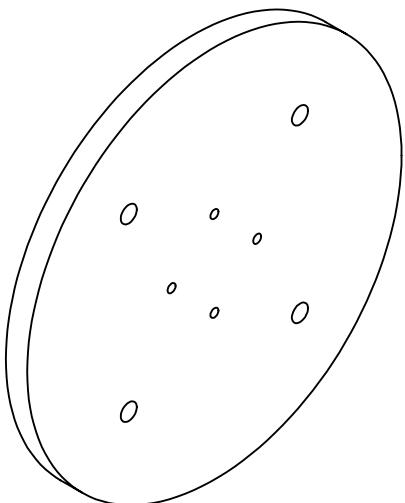
SIZE	DWG. NO.	REV
A		1.0
SCALE: 1:1	WEIGHT:	SHEET 11 OF 13

2

1

2

1



UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN MILLIMETERS		Ryan W.	11/8/2019
TOLERANCES: ± 3.0			
INTERPRET GEOMETRIC TOLERANCING PER:			
MATERIAL	PLA, 20% Infill		
FINISH	N/A		
DO NOT SCALE DRAWING		COMMENTS:	
		Connection plate between smart servo and bevel gear in the wrist.	

Wrist Connection Plate

SIZE	DWG. NO.	REV
A		1.0
SCALE: 1:2	WEIGHT:	SHEET 12 OF 13

B

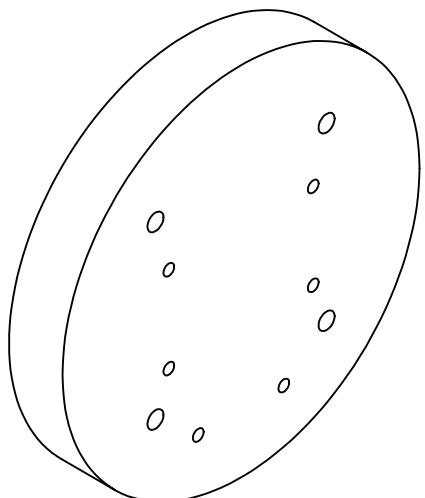
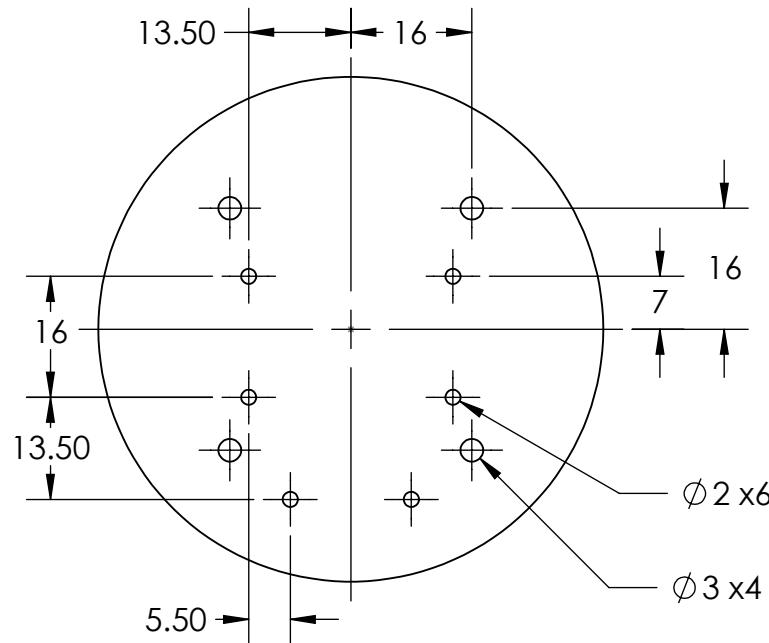
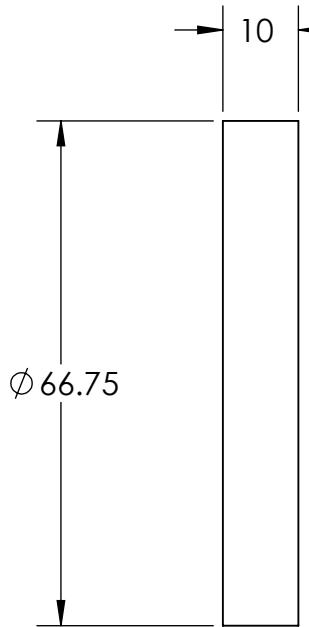
A

2

1

2

1



B

A

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
TOLERANCES: ± 3.0
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
PLA, 20% Infill
FINISH
N/A
DO NOT SCALE DRAWING

O:		NAME	D:
DRAWN	Ryan W.	11/1	
CHECKED			
ENG APPR.			
MFG APPR.			
Q.A.			

COMMENTS:

TITLE:

End Effector

SIZE	DWG. NO.
------	----------

REV
1.0

SCALE: 1:2 WEIGHT:

SHEET 13 OF 13

A.III Salient Code

Listing 1: Actuator Dynamics MATLAB Code

```
1 % dynamixel motor model experiment
2 close all;clear;clc
3
4 % test loads mass moments of inertia
5 m = [.146 .088 0.108];           % mass (kg)
6 b = [.61277 .37227 0.28257];     % length (m)
7 h = [.01915 0.01915 0.0299];      % height (m)
8 J = (1/12)*m.*(h.^2 + b.^2);
9
10 % path to csv files relative to script
11 datapath = 'data/AX12A/';
12 files = dir(strcat(datapath,'*.csv'));
13 numFiles = length(files);
14 % initialize variables
15 [damp, wn, Tp] = deal(zeros(numFiles,1));
16
17 for ii = 1:numFiles
18     % load experimental data, skip 5 header lines
19     M = csvread(strcat(datapath, files(ii).name),5,0);
20     % clean data by removing outliers
21     nani = (find(diff(M(:,1)) > 100));
22     M(nani,:) = [];
23     % show response
24     figure();
25     plot(M(:,1),M(:,2))
26     title('Experimental Data')
27     % find % OS
28     peak = max(M(:,2));                  % peak value
29     peaki = find(M(:,2)==peak, 1, 'first'); % peak value index
30     ss = M(end,2);                      % steady state
31     os = ((peak - ss) / ss) * 100;       % % OS
32     % damping ratio
33     damp(ii) = -log(os/100) / sqrt(pi^2 + log(os/100)^2);
34     % find where the motor begins responding
35     start = M(find(diff(M(:,2)) > 1, 1, 'first'), 1);
36     % time to peak
37     Tp(ii) = (M(peaki,1) - start) / 1000;
38     % natural frequency
39     wn(ii) = pi / (sqrt(1 - damp(ii)^2)*Tp(ii));
40 end
41
42 sol = zeros(4,1);
```

```

43 % no load case, 2*zeta*omega_n
44 sol(1) = 2*mean(damp(end-2:end))*mean(wn(end-2:end));
45 % no load case, omega_n^2
46 sol(2) = mean(wn(end-2:end))^2;
47
48 % obtain average damping ratio and natural frequencies for load cases
49 zeta = [mean(damp(1:3));mean(damp(4:6));mean(damp(7:9))];
50 omegan = [mean(wn(1:3));mean(wn(4:6));mean(wn(7:9))];
51
52 alpha = 2.*zeta.*omegan;
53 beta = omegan.^2;
54 A = zeros(3,2);
55 b = zeros(3,1);
56
57 for jj = 1:3
58     A(jj,:) = [1, -(alpha(jj)*J(jj) + beta(jj)*J(jj))];
59     b(jj) = alpha(jj) + beta(jj);
60 end
61
62 sol(3:4) = A \ b;

```

Listing 2: Forward Kinematics MATLAB Function

```

1 function [r6,T6]= MeiosisFK(theta)
2
3 %      Mapping between joint space and motor space
4 N = 10;          %Gear Ratio
5 A = [ 1/(2*N), 1/(2*N),    0,  0,    0,    0;
6           1/(2*N), -1/(2*N),   0,  0,    0,    0;
7           0,        0,-1/N,  0,    0,    0;
8           0,        0,    0,  1,    0,    0;
9           0,        0,    0,  0,-1/2,  1/2;
10          0,        0,    0,  0,  1/2,  1/2];
11 gamma = A*theta;
12
13 % Define Constants
14 % LB = 12.275;
15 % L1 = 0;
16 % L2 = 25;
17 % L3 = 20;
18 % L4 = 7.2;
19 % L5 = 0;

```

```

20 %      L6 = 5.3;
21
22 %Relative Positions
23 rBfromI = [ 0.00000000; 0.00000000; 0.00000000];
24 r1fromB = [ 0.00000000; 0.00000000; 0.12275000];
25 r2from1 = [ 0.00000000; 0.00000000; 0.00000000];
26 r3from2 = [ 0.00000000; 0.25000000; 0.00000000];
27 r4from3 = [ 0.00000000; 0.20000000; 0.00000000];
28 r5from4 = [ 0.00000000; 0.07000000; 0.00000000];
29 r6from5 = [ 0.00000000; 0.04750000; 0.00000000];
30 %r7from6 = [0;      0;      0]; % dist. from 3rd wrist coor. frame to
31 %                                the end effector is 5.25 cm
32
33 %Orientations wrt I:
34 T1 = rotz(gamma(1));
35 T2 = T1*rotx(gamma(2));
36 T3 = T2*rotx(gamma(3));
37 T4 = T3*rotz(gamma(4));
38 T5 = T4*rotx(gamma(5));
39 T6 = T5*rotz(gamma(6));
40
41 %Positions wrt I:
42 %rB = rBfromI;
43 r1 = r1fromB;
44 r2 = r1 + T1*r2from1;
45 r3 = r2 + T2*r3from2;
46 r4 = r3 + T3*r4from3;
47 r5 = r4 + T4*r5from4;
48 r6 = r5 + T5*r6from5;
49 end

```

Listing 3: Inverse Kinematics MATLAB Function

```

1 function [theta, error] = MeiosisIK(pos,R)
2
3 eOff = [0;47.5;0];
4 npos = pos - R*eOff;
5 xc = npos(1);
6 yc = npos(2);
7 zc = npos(3);
8 L1 = 122.75;

```

```

9      d = 0;
10     L2 = 250;
11     L3 = 270;
12
13     % Inverse Position
14     if (xc^2 + yc^2 -d^2) < 0
15         theta = 1000*[1;1;1;1;1;1];
16         error = 1;
17     else
18         t1 = atan2(yc,xc) - atan2(d,sqrt(xc^2 + yc^2 -d^2)) - pi/2;
19         D = (xc^2 + yc^2 - d^2 + (zc - L1)^2 - L2^2 - L3^2)/(2*L2*L3);
20         t3 = atan2(-sqrt(1-D^2),D);
21         t2 = atan2(zc - L1,sqrt(xc^2 + yc^2 - d^2)) - atan2(L3*sin(t3),L2 +
22             L3*cos(t3));
23
24     % Inverse Orientation
25     T3 = rotz(t1)*rotx(t2)*rotx(t3);
26     T = T3.*R;
27     t6 = atan2(T(2,1),-T(2,3));
28     t4 = atan2(T(1,2),T(3,2));
29     %t4 = atan2(sin(t4),cos(t4));
30
31     if sin(t4) > -10e-6 && sin(t4) < 10e-6
32         t5 = atan2(T(3,2)/cos(t4),T(2,2));
33     else
34         t5 = atan2(T(1,2)/sin(t4),T(2,2));
35     end
36
37     gamma = [t1,t2,t3,t4,t5,t6].';
38
39     % Mapping between joint space and motor space
40     N = 10;           %Gear Ratio
41     % B = [ N, N, 0, 0, 0, 0;
42     %        N,-N, 0, 0, 0, 0;
43     %        0, 0,-N, 0, 0, 0;
44     %        0, 0, 0, 1, 0, 0;
45     %        0, 0, 0, 0,-1, 1;
46     %        0, 0, 0, 0, 1, 1];
47     A = [ 1/(2*N), 1/(2*N), 0, 0, 0, 0;
48         1/(2*N),-1/(2*N), 0, 0, 0, 0;
49         0, 0,-1/N, 0, 0, 0;
50         0, 0, 0, 1, 0, 0;
51         0, 0, 0, 0,-1/2, 1/2;
52         0, 0, 0, 0, 1/2, 1/2];

```

```
53     theta = A\gamma;
54     error = 0;
55 end
56 end
```
