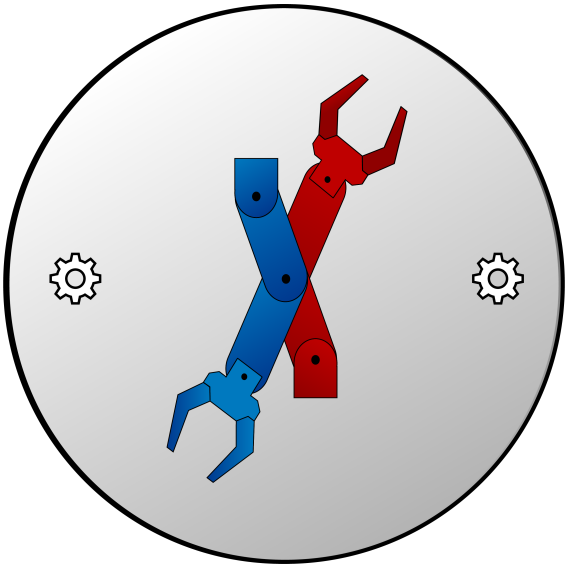
Test Document

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

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.

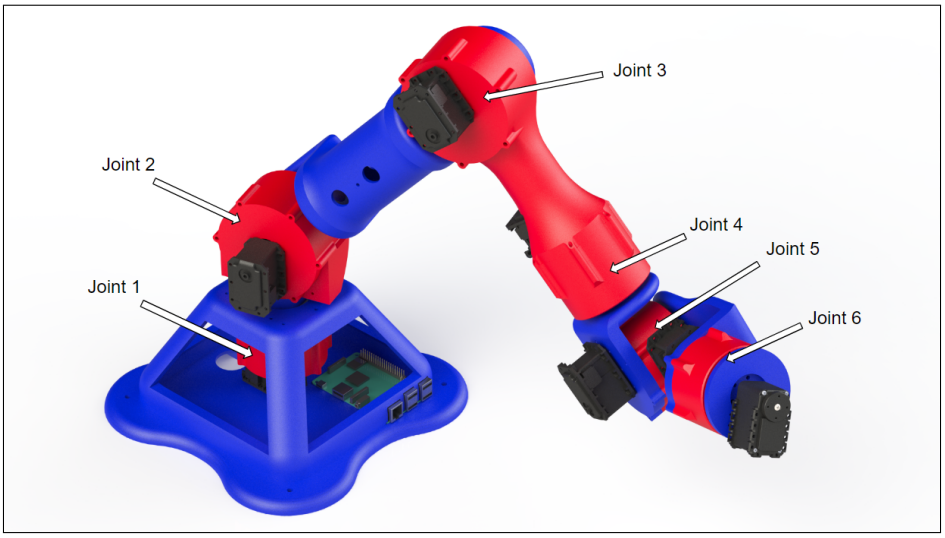


Figure 1: Labeled Image of Manipulator Render

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 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.

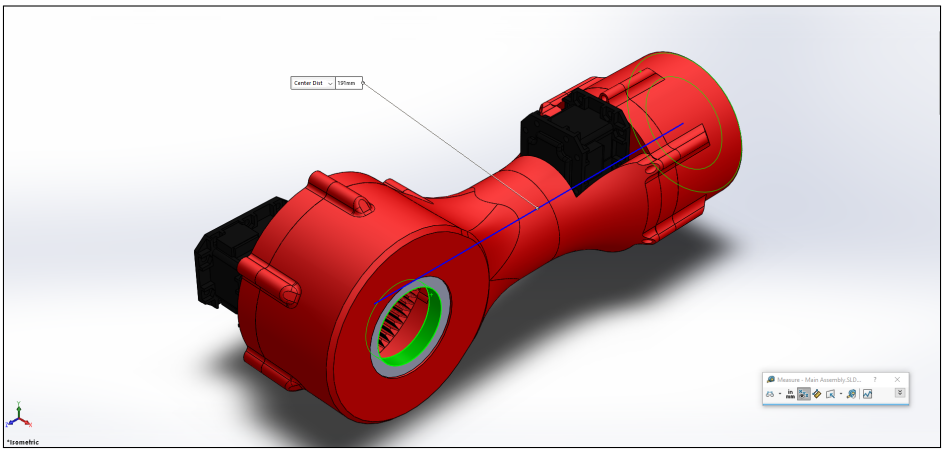


Figure 2: Measurement Test for Link Offset

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.

Specification **2.1.a** states: **The software shall be hosted publicly on an online repository and maintain an MIT license for distribution.**

Specification **2.2.a** states: **The system shall have a user interface capable of**

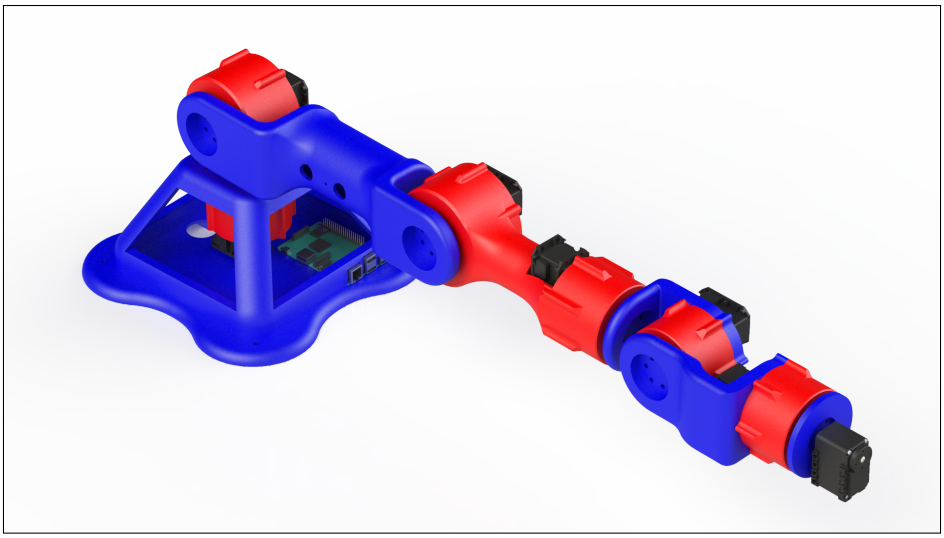


Figure 3: Manipulator in its Zeroed Configuration

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 4.

As shown in Figure 4, 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 a couple points outside of the workspace for the purpose of error checking. The

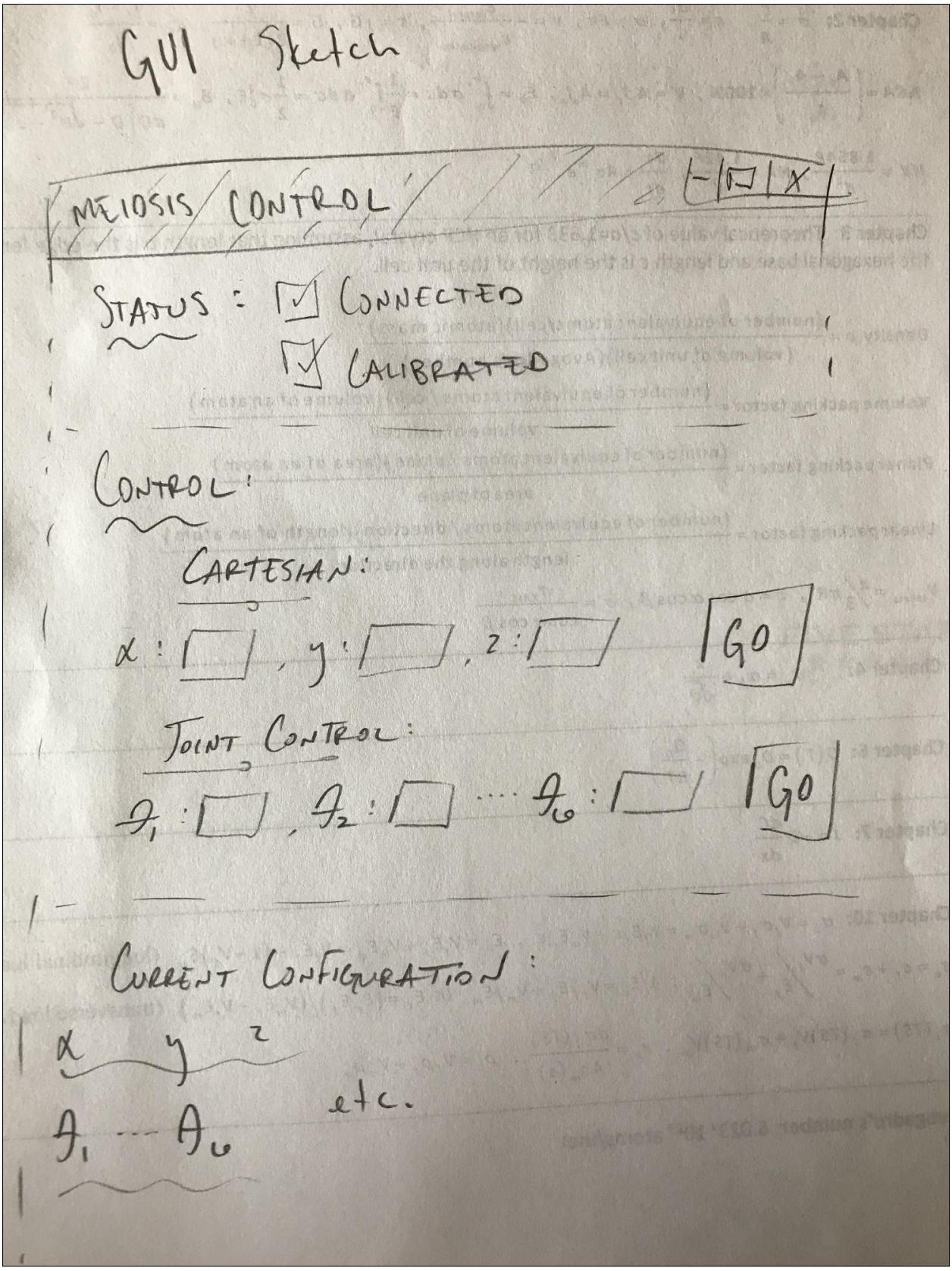


Figure 4: GUI Template

points (in cartesian XYZ and Euler angles) are given as. The

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 [4], which is capable of performing high-speed floating point arithmetic [5]. Based on this fact, **the manipulator meets specification 2.2b.**

3 Conclusion

Table 2: 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

References