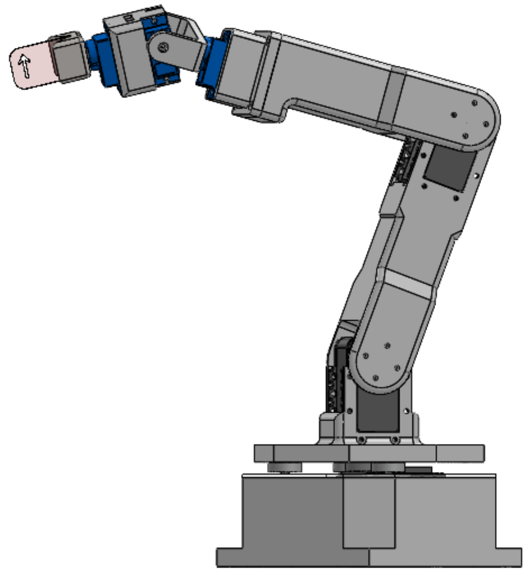
RBE 501
Final Project



Creating a 6 DOF Arm for the RBE 501 Course

By Kevin Boenisch, Meha Mohapatra, Nicole Kuberka and Sriranjani Kalimani

Abstract

The objective of this project is to develop a 6 degree of freedom (DOF) robot arm that could be used in a laboratory component of the RBE 501 course. The proposed manipulator is inspired by industrial robots like the FANUC LR Mate 200iD and developed with a primary focus on simplicity and low cost, allowing for the use of many of these robots in a classroom environment. A full design was created in SolidWorks, components were selected to create a BOM and determine the cost of the robot, and Matlab was used for simulation of the arm. Finally, a 3D-printed mockup of the arm was constructed to further test and analyze the design.

Introduction

The project develops a lab framework for the RBE 501 course similar to the undergraduate robotics courses. Unified Robotics 3 (RBE 3001), where students apply their knowledge about forward, inverse, and differential kinematics by working on cumulative labs, inspired the idea to create a similar experience for RBE 501 students. Based on the team's experiences in the course and speaking to other students, two distinctive topics would be most advantageous to work into a lab component of the class. First, using screw axes in forward and inverse kinematics, in particular when the robot involves a spherical wrist and 6-DOF. Second, implementing dynamics to operate the robot, in particular torque control and polynomial trajectories. These two concepts formed the foundation for designing the RBE 501 lab robot.

RBE 3001 deals with a relatively simple 3-DOF robot. In order to meet the identified needs of RBE 501, industrial robots like the FANUC LR Mate 200iD (Figure 1) were used for inspiration. The proposed robot has 6 degrees of freedom, which includes a spherical wrist and 3 rotary joints. The first three joints use torque controllable servos and the wrist uses compact micro servos. Not only does the robot have a higher level of complexity than the 3 DOF RBE 30001 arm, but it is also highly applicable to the real world.

Figure 1: FANUC LR Mate 200iD [2]

Materials & Methods

Mathematical Model:

The first step in the development of any robot is to analyze it mathematically. The 6x1 matrix of screw axes is composed of the following and is used to help model the robot links and joints:

$$S = [w_1\ w_2\ w_3\ v_1\ v_2\ w_3\ ]^T$$

where w is the axis of rotation and v is the linear velocity. To calculate each component, refer to Appendix D.

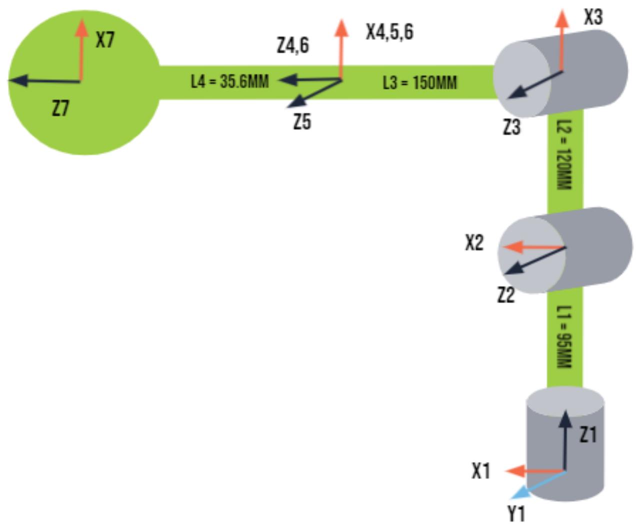


FIgure 2: Home configuration for 6DOF Robot

The home configuration is calculated using the following components:

$$M = [R\, p_7;\ 0\ 0\ 0\ 1]$$

where R is the rotation matrix given by comparing the end effector coordinates to the base, refer to Appendix C.

Simulation:

To set up the simulated manipulator, a SerialLink object was created with six revolute joints using DH parameters calculated based on Figure 3 and with the help of Peter Corke's Robotics Toolbox.

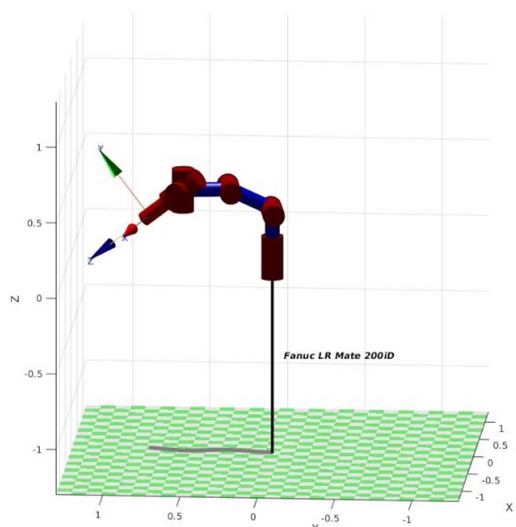
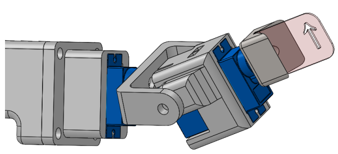
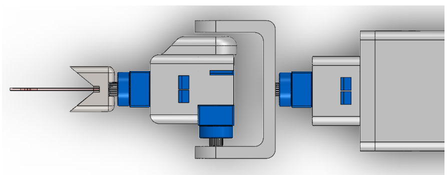
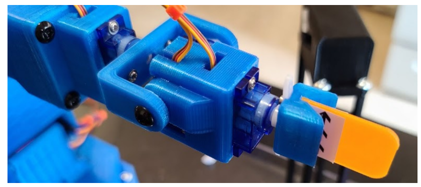


Figure 3: Simulation Model

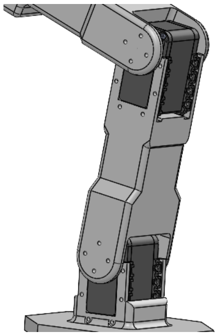
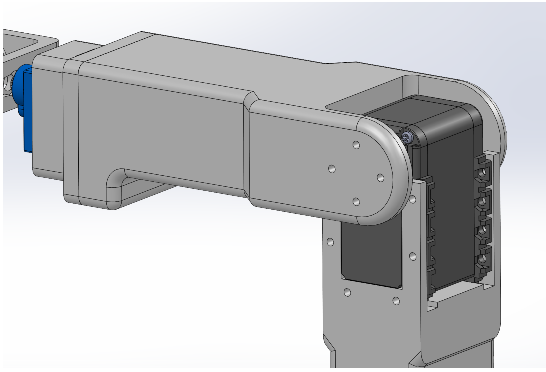
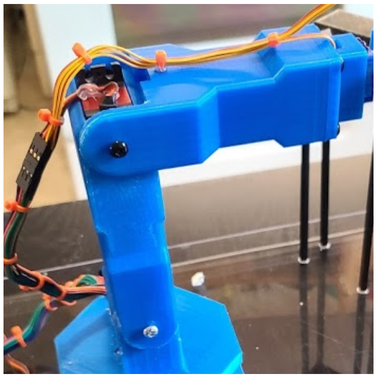
Robot Design:
To meet the identified needs of the robot for a RBE 501 lab, component selection needed to be considered closely. Implementation of dynamic control in the second half of the course requires servo motors with torque control capabilities, however it was determined that options for this were limited, expensive, and did not exist in a “micro” form factor. This presented a challenge to the design of the spherical wrist, as installing 3 large servos in the wrist would be costly and add significant weight to the end of the arm, potentially requiring larger servo motors for other joints. Additionally, having 6 torque controllable joints may introduce unnecessary complexity and confusion for students because having dynamic control over the wrist is unnecessary when moving very light objects. On the other hand, reducing the robot to only have 3-DOF would interfere with the learning objectives in the first half of the course. Therefore, the team determined that the first three joints would use the Dynamixel AX-12A and the spherical wrist would use the TG9e micro servos. Similar to RBE 3001, the spherical wrist would be disregarded for the first half of the course.

The robot was designed using SolidWorks. The spherical wrist was the first to be developed as it needed to be small and lightweight. Effectively mating 3D-printed parts with the TG9e micro servos proved challenging, so commercially produced servo horns were integrated into the wrist design. The wrist was made to be easily removable from the rest of the arm, keeping open the possibility of adding a torque controllable spherical wrist in the future or even completely removing the wrist to have a pure 3-DOF robot. See Figure 4 to 6.

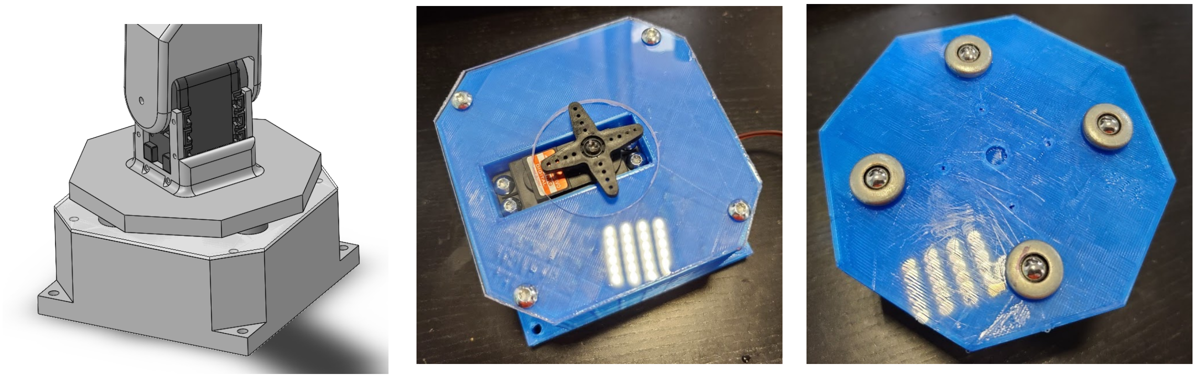
Figures 4-6: Spherical Wrist Design

The upper and lower links of the robot were designed to effectively integrate with the Dynamixel AX-12A servos with a sleek design that is easy to 3D print. The length of each link was determined to create a compact and lightweight robot and allow for clean math. The link lengths can easily be increased should a larger task space be desired in the future.

Figures 7-9: 2nd and 3rd Links

The turntable base was designed for the first joint. The turntable involved 3 ball bearings that roll on a polycarbonate plate; this design choice was made to improve the strength and durability of this relatively high-stress connection.



Figures 10-12: Link1 (turntable) and base

The end task for the RBE 501 lab would remove and insert plates into slots based on color and size within the task space. To hold these plates, a magnet was integrated into the end effector. To release, the end effector simply needs to move away in a twisting motion. The plates were designed to be physically orientation-specific in two axes, with an arrow indicating orientation in the third axis. Both the end effector and slots were designed to allow for a certain tolerance in the end effector's pose.

Figure 13-15: Object, targets, and end effector

Bill of Materials:

The total cost for one robot is estimated at $195.29 which is around $100 less than the RBE 3001 kit. The robot components may change based on budget and design changes. See below.

Item	Unit Cost	Qty
DYNAMIXEL AX-12A Servo	$44.90	3
Turnigy™ TG9e Eco Micro Servo	$2.77	3
3D Printed Components for Arm	$0.02/g	455
Arduino Mega 2560	$40.30	1

Hardware & Cables as needed	Minimal	1
1/16 Polycarbonate for Turntable	Minimal	1
Neodymium Magnets, Pack of 10	$2.79	1
Estimated Total For 1 Robot	$195.29	1

Table 1: Bill of materials for one robot

Experiments

The robot was tested with the Matlab simulation using Peter Corke's Robotics Toolbox. The following tests in Table 4 were conducted. This robot hardware would need to be tested each semester before use.

Index	Test	Procedure	Expected Findings
1	*General mobility*	Move all joints through a range of angles	Joint space
2	*Workspace Constraints*	Move the arm until it reaches singularity and out of bounds	Positional edge cases in task space
3	*Forward Kinematics Test*	Move the robot to a set of N random positions and verify correctness	Correctness of FW Kinematics
4	*Differential Kinematics Test*	Move the robot to a set of N random positions and verify correctness	Correctness of Jacobian
5	*Inverse Kinematics Test*	Move the robot along a path of discrete points	Correctness of Inverse Kinematics, Damping & accounting for previously found constraints
6	*Dynamics Test*	Setup dynamics of the arm and verify correctness of gravity compensation and joint torques	Dynamics verification
7	*Durability (With hardware)*	Test physical ability of arm to carry load and move it to various positions	Verify torque constraints

Table 3: Experiments and procedure

Results

The final 6 DOF robot arm was designed in SolidWorks, integrating 3 Dynamixel AX-12A servos and 3 TG9e micro servos, shown in Figure 16.

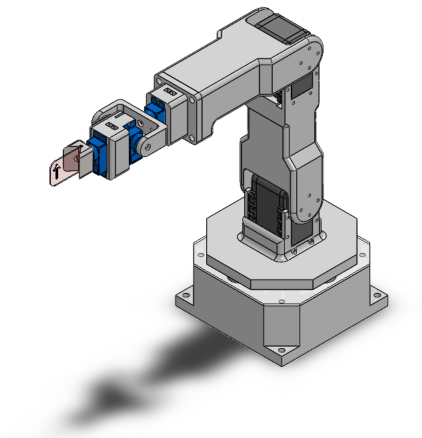


Figure 16: Isometric view of final arm

Experiments in simulation validate the correctness of the robot setup and its kinematics and dynamics. The following images show the model at each test and its assertions. Tests were set up to compare the calculated kinematics with predetermined kinematics based on the robot's serial link setup (using DH parameters).

After completing the design and simulation of the robot, the team moved towards achieving its "reach goal" of constructing and operating a real-life version of the arm. Due to limitations in available supplies, costs, and a tight timeline the team was unfortunately unable to use the chosen Dynamixel AX-12A servos. Instead, the design was modified slightly to use hobby servos from the WPI robotics lab. These servos do not have torque control or as high precision as the AX-12As, but were a suitable replacement for this mockup. All of the parts of the arm were 3D-printed, assembled, and then mounted to a clear polycarbonate base. Two example targets were designed and added for demonstration purposes. Finally, an Arduino Mega 2560 was used as a microcontroller, although almost any microcontroller sold today would be suitable. The completed mockup is shown below.

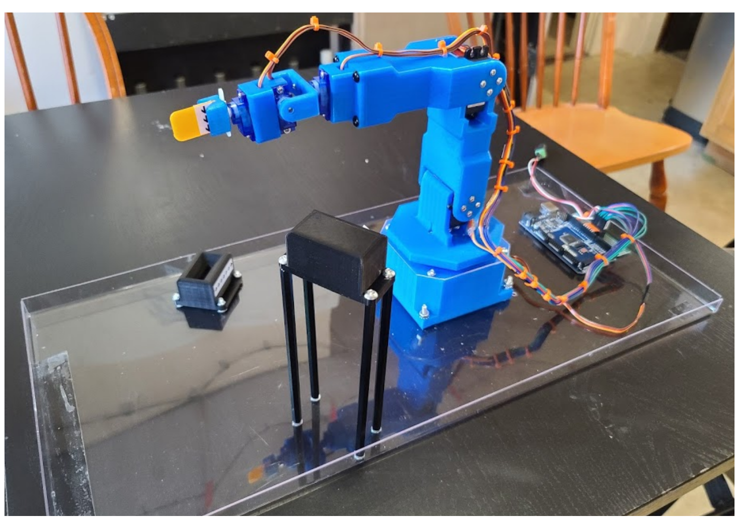


Figure 17: Completed Robot Model

To control the robot arm, quintic trajectory generation was used to find joint positions in the time domain, which were then sent by the microcontroller to the servo motors which have built-in position control. The full set of positions for each joint were calculated using Matlab and the arm was controlled with code written in C++. In the future, code would be implemented for direct communication between the microcontroller and the results that are calculated by Matlab. Ultimately, the robot arm was able to complete an example task as intended. This can be seen in this video: https://youtube.com/playlist?list=PLHcbIhsfuV05K_iyjRM_qiM24hNeqtPPX

Discussion

The current design of the SolidWorks model of the arm underwent a redesign to achieve the end product, now using torque controllable servos. Despite not being able to use the chosen Dynamixel AX-12A servos due to factors out of the team's control, hobby servos were substituted to build a full robot model with 3D printed parts. Creating this physical model was crucial to validate the design and verify the strength of the 3D printed parts and their interfaces.

At first, operating the model was difficult due to the jerky motion of the servos. However, adding quintic polynomial control almost eliminated this issue and resulted in smooth and controlled motion. This also showed that with the addition of high-precision, torque controllable servos, the arm will be significantly more precise and easy to use.

The simulation was verified by comparing DH parameter methods to the equivalent screw axes based methods and Peter Corke's Robotics Toolbox. This confirmed that the simulated robot was created correctly and that each lab was possible.

Future Work

While the team was successfully able to design a 6-DOF robot for the RBE 501 course, further work can be completed to fully prepare this system for implementation into labs. The full robot, including Dynamixel AX-12A servos instead of the current placeholders, should be built and extensively tested. With this exact model built, it will be possible to confirm assumptions such as the accuracy of reaching targets and whether the task space is a suitable size for a potential challenge that students will complete.

Further framework code should be developed to allow students to directly control the robot from the results of their Matlab scripts and thoroughly tested to avoid frustrations in the lab. The lab curriculum should also be tested with the help of instructors and TAs to ensure it aligns with the goals of the class and presents the right level of difficulty for students.

Conclusion

The project was able to put together a 6-DOF educational robot that could allow the creation of lab time for 501. While there is still further work to be done before being implemented. We have laid out the groundwork that can be built upon.

References

[1] K. Harrington, “3001 HephaestusArm2,” *HephaestusArm2*, Apr-2005. [Online]. Available: https://github.com/Hephaestus-Arm/HephaestusArm2. [Accessed: 2021].

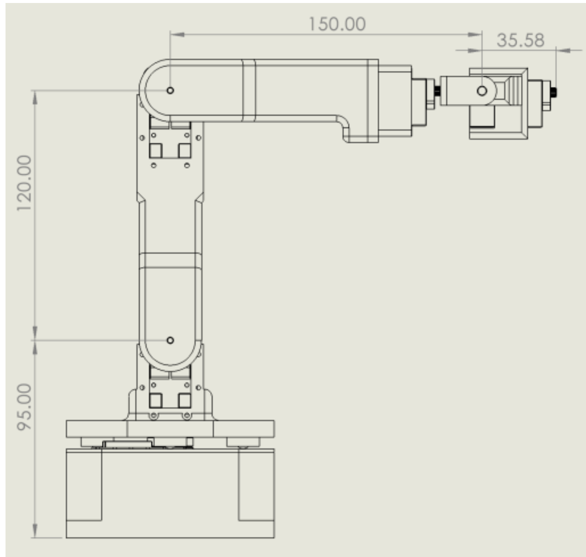
[2] “FANUC LR Mate 200iD Robot: Small Payload Robot: FANUC America,” *FANUC*. [Online]. Available: https://www.fanucamerica.com/products/robots/series/lr-mate/lr-mate-200id. [Accessed: 29-Mar-2021].

Authorship

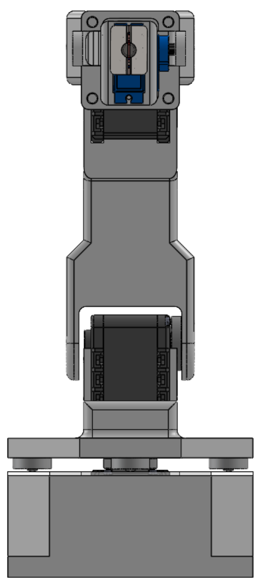
Section	1st Writer	2nd Writer	1st Editor	2nd Editor
Abstract	Nicole	Kevin	Meha	Sri
Intro	Meha	Nicole	Kevin	Sri
Materials & Methods	Kevin	Meha	Kevin	Nicole
Experiments	Sri	Kevin	Nicole	Kevin
Results	Sri	Kevin	Nicole	Meha
Discussions	Kevin	Nicole	Meha	Sri
Conclusion	Nicole	Kevin	Meha	Sri

Table 4: Authorship as of May 12

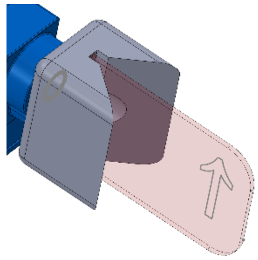
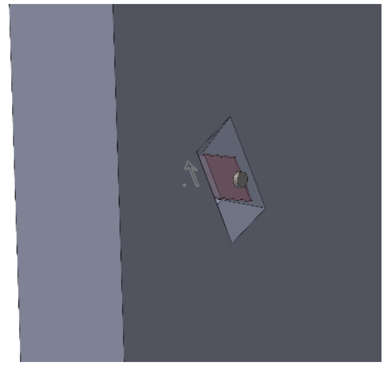
Appendix A: SolidWorks Robot Model



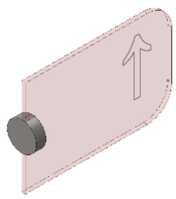
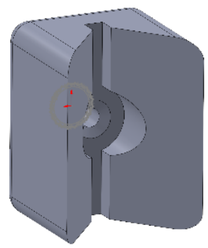
Side View with Dimensions in mm



Front view

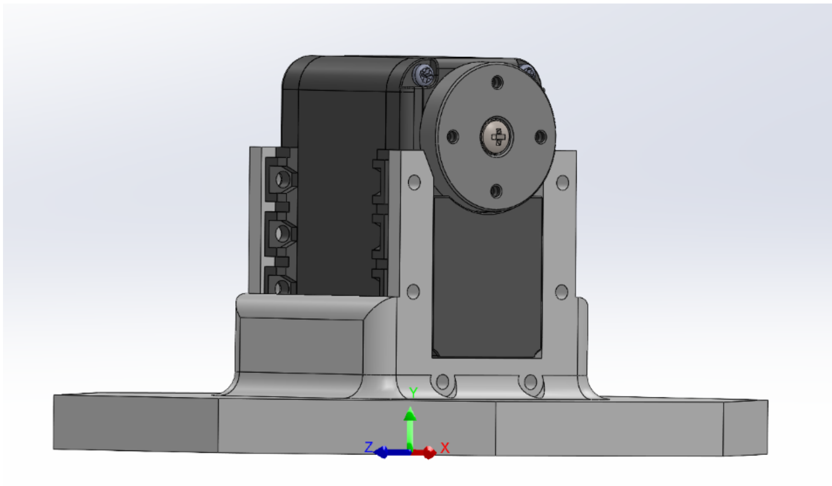
End Effector with magnetically held object, Object placed into example target

Object with magnet, End effector without object

Appendix B: Robot Model Inertias

Link 1



Mass = 79.59 grams

Volume = 117818.78 cubic millimeters

Surface area = 42023.18 square millimeters

Center of mass: (millimeters)
X = 0.83
Y = 24.98
Z = 0.00

Principal axes of inertia and principal moments of inertia: (grams * square milli
Taken at the center of mass.

Ix = (0.20, 0.98, 0.00)	Px = 39088.38
Iy = (-0.98, 0.20, 0.00)	Py = 43302.83
Iz = (0.00, 0.00, 1.00)	Pz = 45614.12

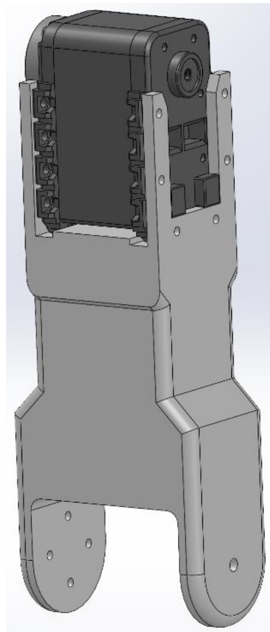
Moments of inertia: (grams * square millimeters)
Taken at the center of mass and aligned with the output coordinate system.

Lxx = 43133.93	Lxy = 826.63	Lxz = -0.01
Lyx = 826.63	Lyy = 39257.28	Lyz = 0.01
Lzx = -0.01	Lzy = 0.01	Lzz = 45614.12

Moments of inertia: (grams * square millimeters)
Taken at the output coordinate system.

Ixx = 92781.74	Ixy = 2476.40	Ixz = -0.01
Iyx = 2476.40	Iyy = 39312.10	Iyz = 0.03
Izx = -0.01	Izy = 0.03	Izz = 95316.75

Link 2



Mass = 86.31 grams

Volume = 137957.47 cubic millimeters

Surface area = 36477.82 square millimeters

Center of mass: (millimeters)
X = 0.76
Y = 81.92
Z = 0.00

Principal axes of inertia and principal moments of inertia: (grams * square millime
Taken at the center of mass.

Ix = (0.02, 1.00, 0.00)	Px = 17557.62
Iy = (-1.00, 0.02, 0.00)	Py = 108356.36
Iz = (0.00, 0.00, 1.00)	Pz = 114782.07

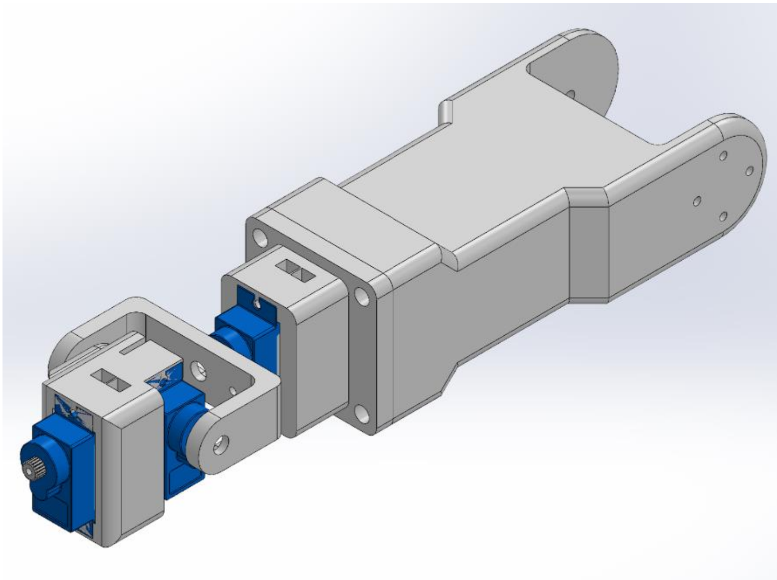
Moments of inertia: (grams * square millimeters)
Taken at the center of mass and aligned with the output coordinate system.

Lxx = 108322.81	Lxy = 1745.03	Lxz = -0.01
Lyx = 1745.03	Lyy = 17591.16	Lyz = 0.04
Lzx = -0.01	Lzy = 0.04	Lzz = 114782.07

Moments of inertia: (grams * square millimeters)
Taken at the output coordinate system.

Ixx = 687574.04	Ixy = 7133.23	Ixz = -0.01
Iyx = 7133.23	Iyy = 17641.29	Iyz = 0.04
Izx = -0.01	Izy = 0.04	Izz = 694083.42

Link 3



Mass = 77.84 grams

Volume = 159848.71 cubic millimeters

Surface area = 43627.70 square millimeters

Center of mass: (millimeters)
X = -0.14
Y = 0.08
Z = 99.61

Principal axes of inertia and principal moments of inertia: (grams * square millimet
Taken at the center of mass.

Ix = (0.00, -0.01, 1.00)	Px = 14722.67
Iy = (1.00, 0.00, 0.00)	Py = 202318.78
Iz = (0.00, 1.00, 0.01)	Pz = 205455.44

Moments of inertia: (grams * square millimeters)
Taken at the center of mass and aligned with the output coordinate system.

Lxx = 202317.64	Lxy = 3.47	Lxz = -464.12
Lyx = 3.47	Lyy = 205417.02	Lyz = -2706.90
Lzx = -464.12	Lzy = -2706.90	Lzz = 14762.24

Moments of inertia: (grams * square millimeters)
Taken at the output coordinate system.

Ixx = 974593.22	Ixy = 2.52	Ixz = -1577.79
Iyx = 2.52	Iyy = 977693.66	Iyz = -2052.49
Izx = -1577.79	Izy = -2052.49	Izz = 14764.40

Appendix C: Rotation Matrix and Home Configuration

R **Rotation Matrix**			
Axis of Base	**Axis of. End Effector**		
	x	y	z
x	0	0	1
y	0	-1	0
z	1	0	0

<table>
<tr><th colspan="4">M Home Configuration</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td rowspan="3">p7</td></tr>
<tr><td>0</td><td>-1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>

Appendix D: W vector, P vector, and Si screw axis of home configuration

w components	p components

w1 = [0; 0; 1];
w2 = [0; 1; 0];
w3 = [0; 1; 0];
w4 = [1; 0; 0];
w5 = [0; 1; 0];
w6 = [1; 0; 0];
	p1 = [0; 0; 0];
	p2 = [0; 0; L1];
	p3 = [0; 0; L2+L1];
	p4 = [L3; 0; L2+L1];
	p5 = [L3; 0; L2+L1];
	p6 = [L3; 0; L2+L1];
	p7 = [L3+L4; 0;H2+H1];

Si screw axis	
S = [0; 0; 0; 1; 0; 1; 0; 1; 1; 0; 1; 0; 1; 0; 0; 0; 0; 0; 0; -0.095; -0.215; 0; -0.214; 0; 0; 0; 0; 0; 0.215; 0; 0.215; 0; 0; 0; 0; 0.15; 0];	

Appendix E: 501 Created Labs

Lab Breakdown

Overall

- Labs will build on each other.
- Students are given an objective & lab specific stub file.
- Students will deliver a focused lab report & do a physical implementation.

Lab 1: Basic Kinematics In Space Frame

OBJECTIVE: Complete Forward, Inverse, and Velocity Kinematics in the Space Frame

Given
- RVC2
- Completed Serial Link
- Lab1.m template code
- Stub function files

Assignment
- Calculate screw axes
- Calculate forward kinematics using POE
- Calculate space jacobian
- Calculate inverse kinematics

Physical Implementation
- Forward kinematics
- Inverse kinematics
- Avoids Singularities

Lab Report
- Explain how each function works
- Explain how assert works
- Explain singularities and you choose to avoid them

Lab 2 : Basic Kinematics in Body Frame

OBJECTIVE: Complete Forward, Inverse, and Velocity Kinematics in the Body Frame

Given
- RVC2
- Lab2.m template code
- Stub function files
- Files from previous labs

Assignment
- Calculate screw axes
- Calculate forward kinematics
- Calculate body jacobian
- Calculate analytical jacobian

- Calculate inverse kinematics

Physical Implementation
- Forward kinematics
- Inverse kinematics
- Avoids Singularities

Lab Report
- Explain how each function works
- Explain the difference between space and body frame and why each perspective is important
- Explain the value of the analytical jacobian

Lab 3 : Dynamics

OBJECTIVE: Understand the importance & how to calculate forward & inverse dynamics

Given
- RVC2
- Lab3.m template code
- Files from previous labs

Assignment
- Calculate transforms b/w links
- Calculate spatial inertia matrices
- Check forward dynamics
- Check inverse dynamics
- Calculate wrench to control motion

Physical Implementation
- Smooth motions to each setpoint

Lab Report
- Explain what dynamics is and why it's important to robotics
- Explain each check/test's value

Lab 4 : Pick and Place

OBJECTIVE: Combine previous knowledge with computer vision to sort objects

Given
- RVC2
- Files from previous labs
- Built-In MATLAB Camera Toolbox

Assignment
- Calibrate Camera
- Create masks
- Calculate transformations wrt camera position
- Develop fluid code to sort objects by size and/or color

Physical Implementation
- Sort plates by size and/or color

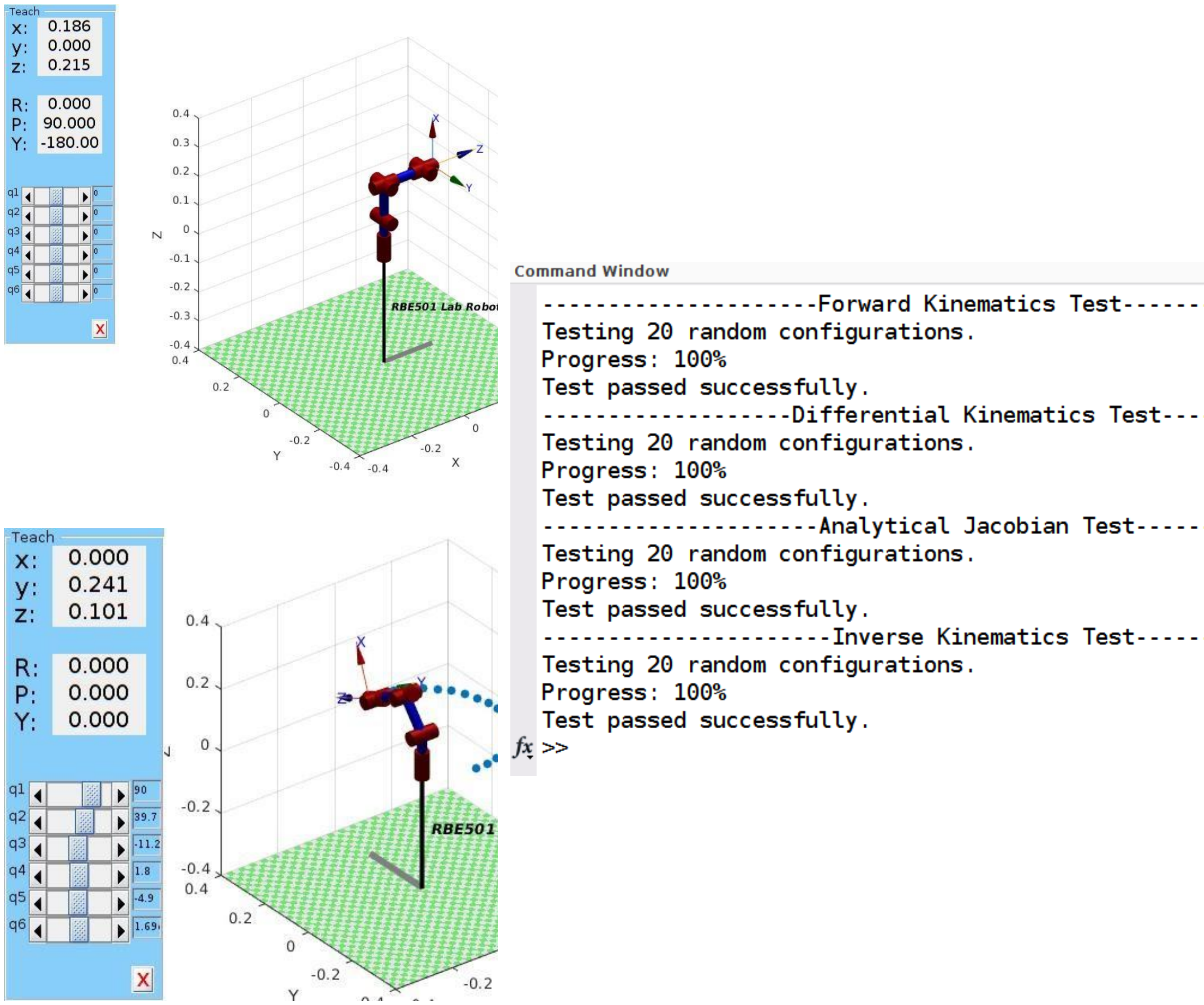
Lab Report

- Explain and show how calibration works
- Explain and show process of sorting (combines computer vision and previous labs)

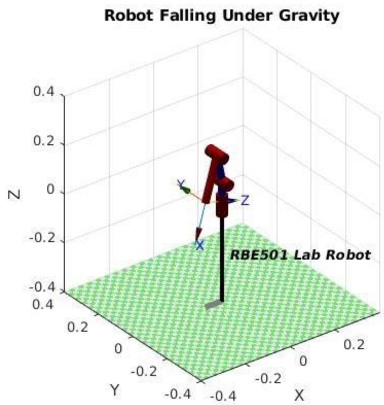
The code for this project can be retrieved here. For access to student lab material as part of the explained curriculum, refer to the Student Lab Material folder of the repository at Student Lab Material. The Matlab folder of the repository contains fully developed code that can be used as reference for the developed curriculum.

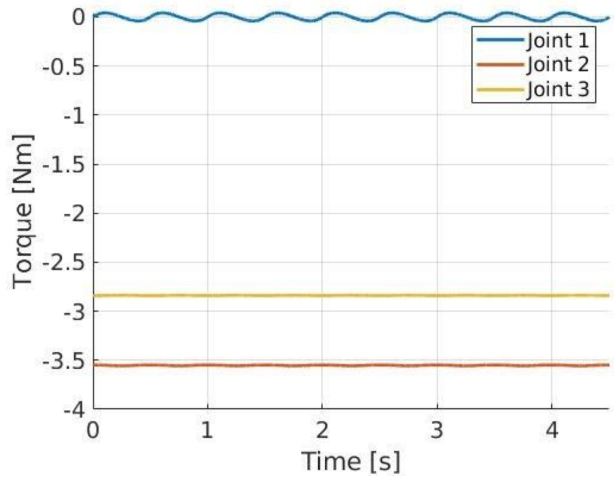
Appendix F: Simulation Results

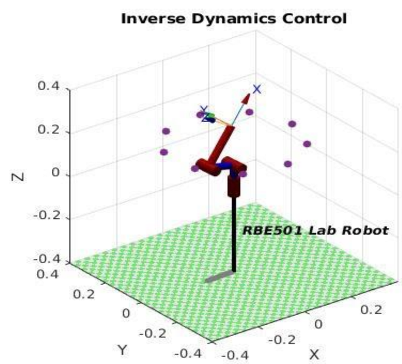
The following images show simulation outputs from Tests 1-5 and 6 respectively of the designed experiments (see section Experiments).



Results of Tests 1-5 from Table 2







Command Window
```
--------------Simulation of Robot Falling Under Gravity---------------
147                    drawnow
Simulation complete. Press Enter to continue.
--------------------------Gravity Compensation------------------------
Joint Torques: [0.000000 -0.075559 -0.075559] Nm
Simulation complete. Press Enter to continue.
---------------------Dynamic Control of a 3-DoF Arm-------------------
Generating task space path... Done.
Calculating the Inverse Kinematics... Done.
Generate the Joint Torque Profiles... Done.
Simulate the robot...Done.
Program completed successfully.
---------------------Dynamic Control of a 3-DoF Arm v2----------------
Generating task space path... Done.
Calculating the Inverse Kinematics... Done.
Generate the Joint Torque Profiles... Done.
Simulate the robot...Done.
Program completed successfully.
fx >>
```

Dynamic Modeling Results