Robotic Arm Final Project Report

EEL 4664- Kinematics and Control of Robotics

Group Members:

Kilnauel Santana, Gianna Reyes, Lillian Wright

12/1/2025

Introduction:

The goal of this project was to build and control a 3D-printed robotic arm capable of autonomously performing a pick and place task. Following the guide that was provided we assembled a six degree of freedom servo based robotic arm, programmed it using an Arduino Mega, use Bluetooth connection along with the MIT App Inverter to control the arm, and implemented an automated behavior to repeatedly pick up a rubber duck and place it into a box positioned on the opposite side of the robot.

This project allowed us to apply concepts from kinematics, servo control, and embedded programming. The final robot demonstrates continuous movement being controlled via the android app. The extra credit portion of this project demonstrates continuous, autonomous pick and place functionality triggered by a single user command, meeting the course requirements.

Bill of Materials:

Component	Quantity	Price	Comments
Arduino Mega	1	$35.00	Main controller
MG996R high torque servo	3	3x($10)= $30.00	Waist, shoulder, elbow
SG90 micro servo	3	3x($3)= $9.00	Wrist roll, wrist pitch, gripper
3D printed parts	Full set	$0.00	Provided for by instructor
Breadboard	1	$5.00	Power distribution
Jumper wires	1 pack	$4.00	Signal connections
HC-05 Bluetooth module	1	$7.00	For manual phone control
External power supply	1	$10.00	Required to power servos
Power barrel adapter/ screw terminal	1	$3.00	Power connection for servo
Misc. Hardware *screws, zip ties, nuts)	-	$3.00	Assembly materials

Estimated total cost for this project start to finish ≈ $109

Methods:

For the mechanical assembly, we followed the step by step build instructions from the guide that was provided to assemble the base, shoulder, elbow joint, wrist rotation, and gripper.

Servos were aligned to neutral before mounting to prevent binding. Figure 1 shows the complete assembled robot.

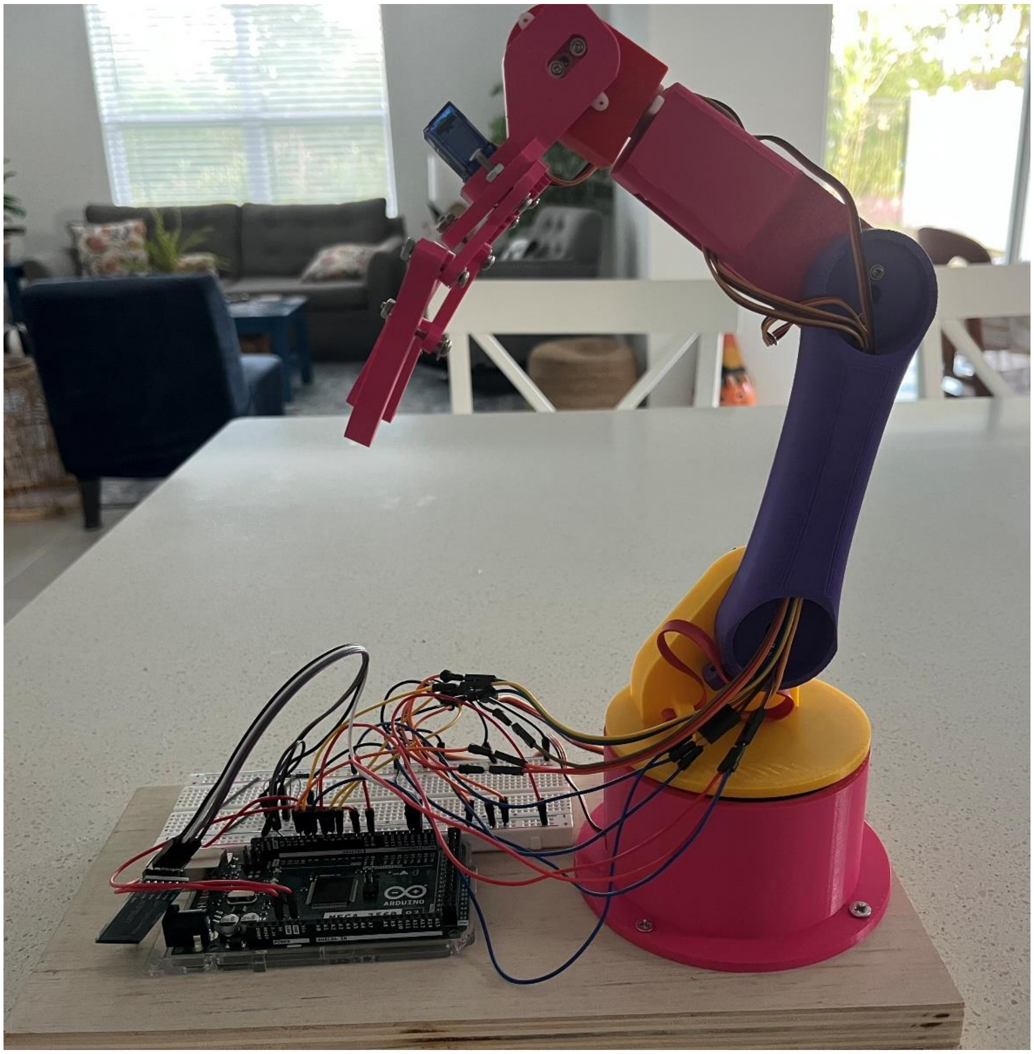

Figure 1: Mechanical assembly of the robotic arm

In addition to the mechanical assembly, we designed the electrical system to ensure stable and reliable control of the servos. The Arduino Mega was used and provided enough PWM pins to independently drive al six servos. Each servo was wired so that's its signal line connected directly to a PWM enable pin on the Mega, and the servo power was connected to the Megas 5V power supply. This prevented voltage drops and burnouts that can occur. All grounds were tied together to ensure a common reference.

To enable manual control the HC-05 Bluetooth module was wired to the Megas serial pins. Using the MIT App Invertor, we used the custom Android app with sliders that allowed us to control each joint angle in real time. This made testing and calibration significantly easier, as the app provided smooth, continuous control of the robot before implementing autonomous functionality. Figure 2 shows the circuit setup of the servo motors, HC-05 module, along with the external power source to the Mega.

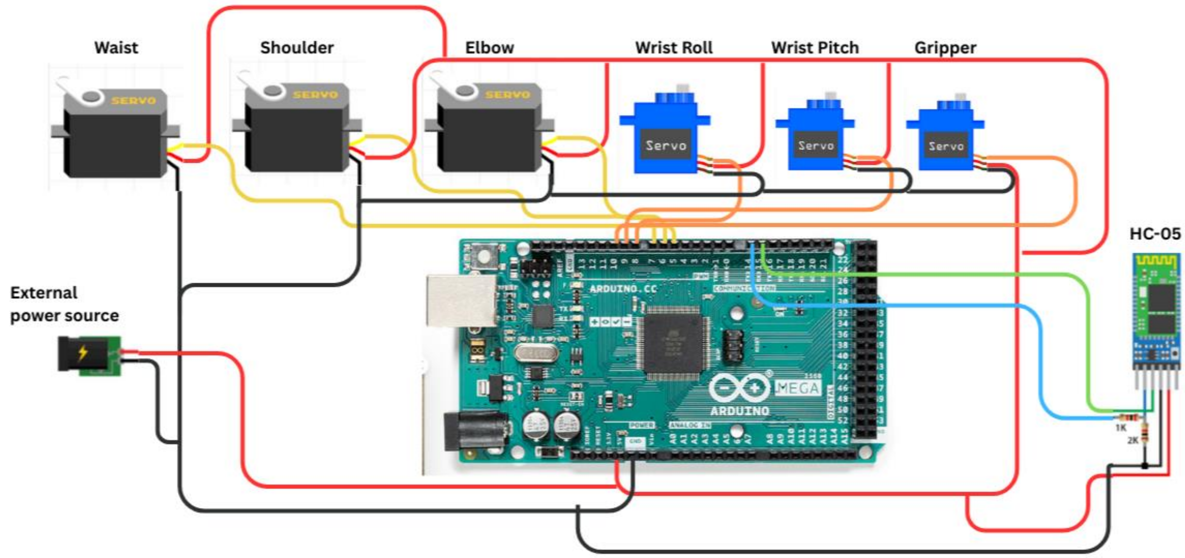

Figure 2: Circuit design of the motors and Bluetooth connector

Calculations:

We want to compute the joint angles θ_1, θ_2, and θ_3 required to move a 3-DOF robotic arm's end effector to a target position (x, y, z). The robot has:

- Base rotation joint θ_1

- Shoulder joint θ_2

- Elbow joint θ_3

Link lengths are L1 and L2, and the tip is extended by the claw length Lc pointing downward.

<u>Step 1:</u>

Since the claw points straight downward, the wrist is located Lc above the tip along the +z axis:

x_w = x

y_w = y

$z\_w = z + Lc$

Step2:

The base rotation aligns the arm with the target in the XY-plane:

$\theta_1 = \text{atan2}(y\_w, x\_w)$.

Step3:

The wrist is now expressed in cylindrical coordinates:

$r = \text{sqrt}(x\_w^2 + y\_w^2)$

$z = z\_w$

Step 5:

Distance to wrist:

$d^2 = r^2 + z^2$

Step 6:

Using the law of cosines:

$D = (d^2 - L1^2 - L2^2) / (2 \cdot L1 \cdot L2)$

Step 7:

Elbow-down configuration:

$\theta_3 = \text{atan2}(-\text{sqrt}(1 - D^2), D)$

Step 8:

Compute values:

$\phi = \text{atan2}(z, r)$

$\beta = \text{atan2}(L2 \cdot \sin(\theta_3), L1 + L2 \cdot \cos(\theta_3))$

Step 9:

$\theta_2 = \phi - \beta$

Solution

```
IK solution (radians): theta1 = 0.0000, theta2 = 1.4846, theta3 = -2.2988
IK solution (degrees): theta1 = 0.00, theta2 = 85.06, theta3 = -131.71

Desired TIP:   (100.00, 0.00, -100.00) mm
```

Figure 3: Matlab solutions to inverse kinematics

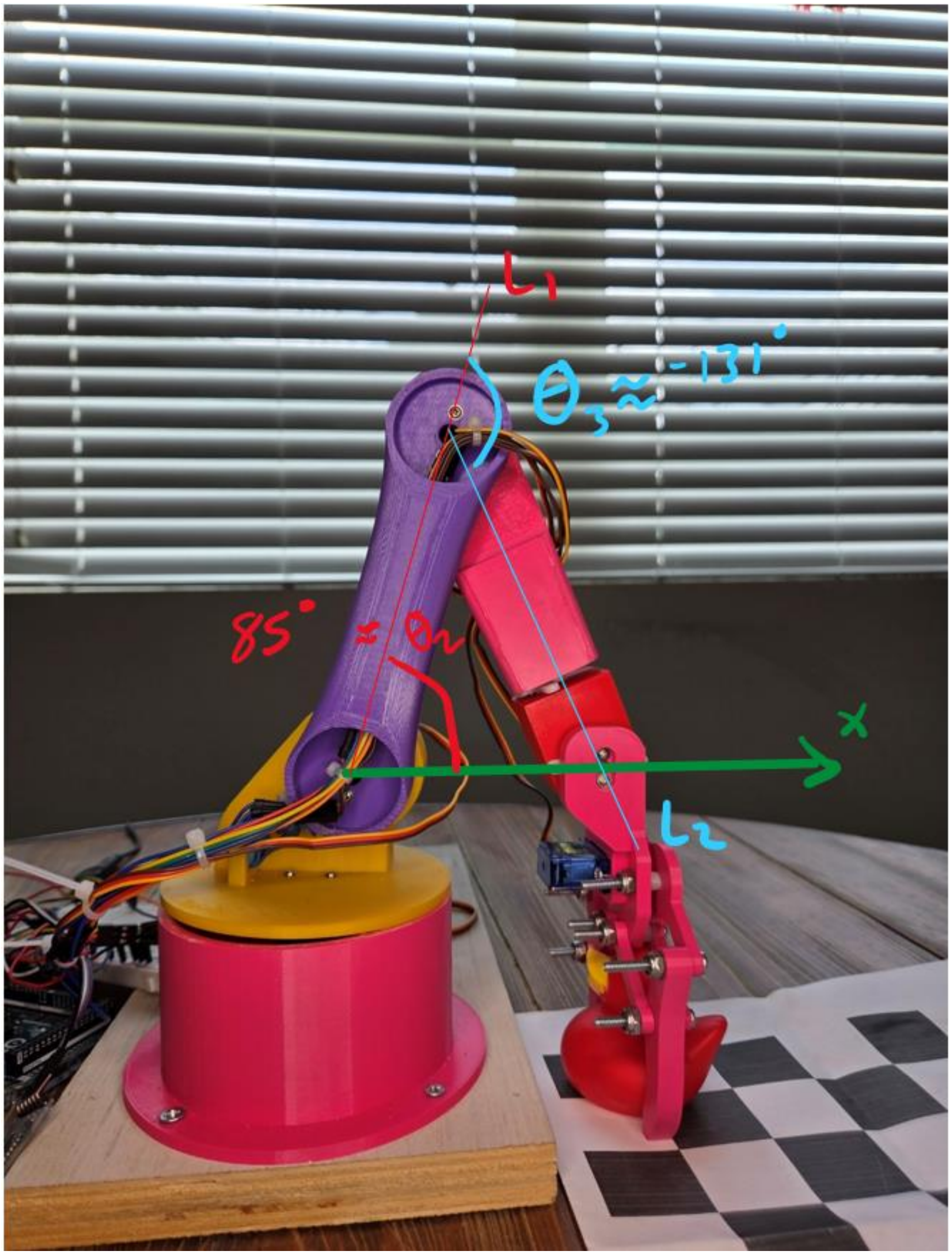

Figure 4: Robot arm showing simulation of joint angles

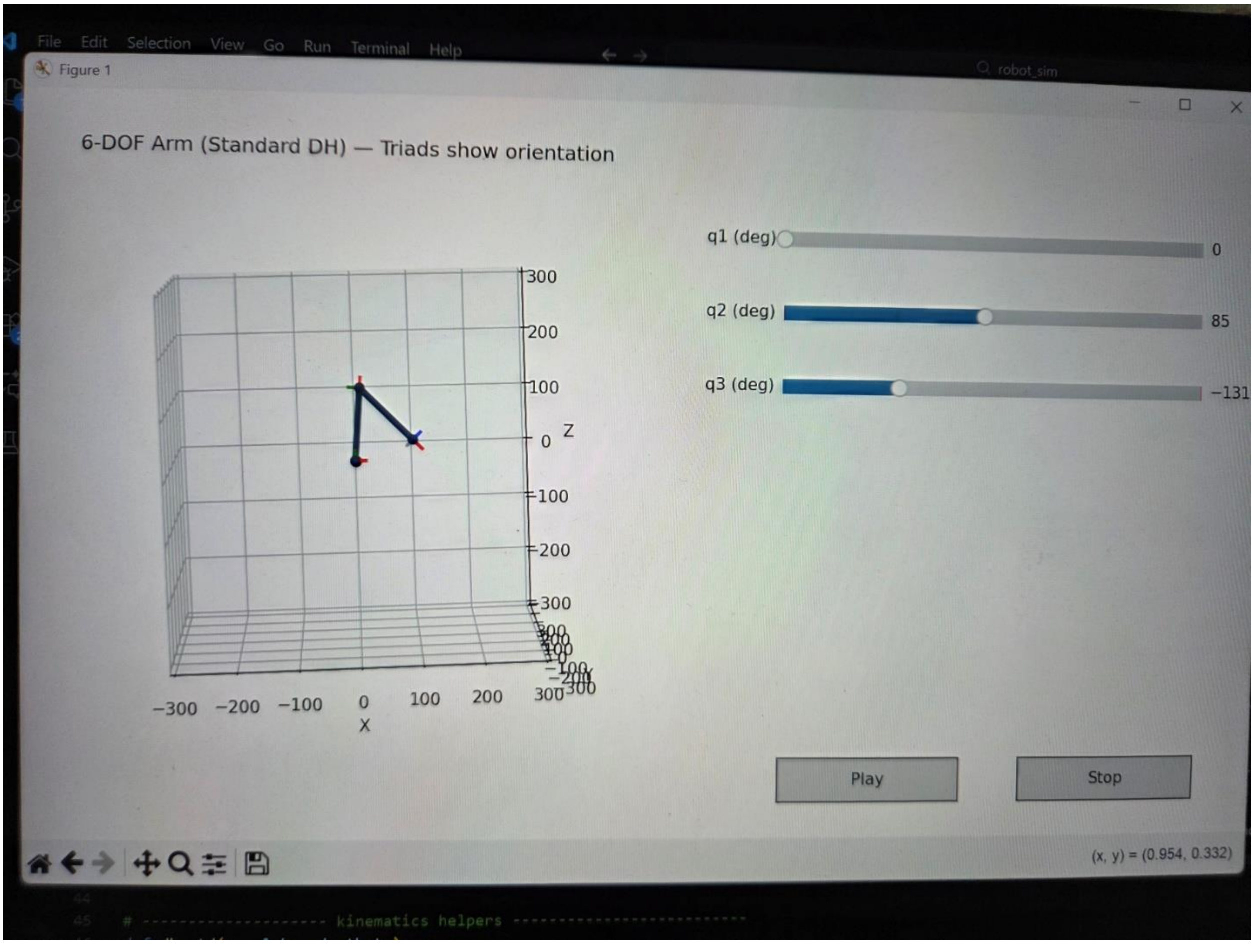

Figure 5: VS code simulation of position 100, 0, 0

Figure 4 and 5 shows the comparison between the actual robotic arm angle and the simulation which was created using VS code.

Results and Discussion:

The final robotic arm successfully demonstrated both manual and autonomous functionality. Using the MIT App Invertor interface, we were able to smoothly control each joint allowing us to test the servo limits, verify motion ranges, and calibrate the gripper to reliably hold the duck. The Bluetooth connection proved stable and responsive, making it easy to adjust joint angles during early development stages.

Once autonomous mode was implemented, the robot was able to complete repeated pick and place cycles. After pressing the start button, the robot consistently moved to pickup point, lowered the arm, closed the gripper around the duck, transported it across the workspace, and finally released it into the box. The sequence ran continuously as designed, meeting the project requirement for autonomous operation.

During testing, we observed that the shoulder and elbow servos were responsible for most of the load and needed slower speed increments to avoid sudden jerks. Additionally, the gripper required several adjustments to ensure a secure hold. The external power supply was essential, attempting to power the servos from the Arduino alone did not work. The servos could not handle that much weight just on the Arduinos power source alone. After these improvements, the final system was able to reliably perform repeated cycles without overheating or drifting out of alignment.

Overall, the arms performance validated our kinematic planning and programming logic, and it was able to consistently execute the pick and place task required for the final extra credit project demonstration.

Conclusion:

In conclusion, this project provided a hands on experience with robotic kinematics, embedded control, and system integration. By assembling a 3D-printed robotic arm, wiring the electronics, and developing both Bluetooth based manual control and autonomous behavior, our team gained practical understanding of how manipulators are designed and programmed. The final robot successfully completed continuous pick and place cycles using a single start command, meeting all project requirements. The project also reinforced important engineering skills, including debugging servo motion, managing power distribution, and developing reliable control sequences. Overall, the project was a valuable opportunity to bridge theoretical robotics concepts with a functioning real world system.

Acknowledgement:

We would like to thank Dr. Hoan Ngo for providing the 3D printed parts, electronic components, and guidance throughout this course. We also acknowledge the Makerspace staff for giving us access to tools and assistance during assembly. Their support made it possible for us to successfully complete this project.

References

D. Kovacevic, “DIY Arduino Robot Arm with Smartphone Control”, *HowToMechatronics.com*, Aug. 2019. [Online]. Available: https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/. [Accessed: Nov. 24, 2025].

Handson Technology. "MG996R Metal Gear Servo Motor User Guide." Handson Technology, www.handsontec.com. Accessed October 2023.

J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 4th ed., Pearson, 2017.

MIT App Inventor, “Get Started with MIT App Inventor 2”. [Online]. Available: https://appinventor.mit.edu/explore/ai2/tutorial. [Accessed: Nov. 24, 2025].