Robotic Arm with Six Degrees of Freedom

Darian Brokaw

Ethan Matthews

Gabriella Rodriguez

Florida Polytechnic University

Advanced Kinematics and Control of Robotic Systems (EEL5668C)

Dr. Hoan Ngo

December 2, 2025

https://github.com/EthanMatth/KinematicsArm

Introduction

For this project, the team built and programmed two different robotic arms. The first was a conventional rigid 6 degrees of freedom robotic arm, similar in construction and design to a PUMA 560 robotic arm. The team developed and implemented an inverse kinematic system for this robot by breaking the problem into two smaller problems, one for location and one for orientation. The other robotic arm was a soft tentacle-like robotic arm based on the work from the University of Science and Technology of China[1] on this topic. This simple 2D tentacle was controlled by 2 sets of strings, pulled by high torque motors, which had the effect of curling the arm so that it could grasp objects.

Materials & Methods

Bill of Materials

- 6pcs MG996R Servo Motor
- 5pcs SG90 Micro Servo Motor
- 5V 2A power Supply
- Female DC Power Jack Plug Adapter Barrel Connector

Methods

To solve the inverse kinematics for the conventional robotic arm, the team broke the problem in two. The first half was for position (as given by Cartesian x, y, z coordinates) and the second half was for rotation (as given by roll, pitch and yaw inputs). The x and y values were converted to polar coordinates giving radius and distance according to the formulas below:

$$r = \sqrt{x^2 + y^2}, \theta = tan^{-1}(y/x)$$

Then by ignoring the claw orientation, it was a simple problem to point the claw at the given angle θ_1 and distance r. The z value was then simply used for the height h. This allowed

the robot to place the claw at any point in a cylindrical volume around it. Using geometric calculations, a triangle was created such that the length is r and the height is h. The hypotenuse P was calculated as shown below. Then the robot was determined to be in elbow-up position if one link was at least 90 degrees and the other was less than 90 degrees, or elbow-down if both links were greater or less than 90 degrees. From there, the values of γ, α and β were calculated as shown below. Finally, these values were plugged into the below equations to solve for θ_2 and θ_3. Note that in the equation for θ_3, the values are summed if the robot is elbow-up and subtracted if the robot is elbow-down. Similarly, the result of θ_2 is negative if the robot is elbow-up and positive if the robot is elbow-down.

$$P = \sqrt{r^2 + h^2}, \qquad \gamma = cos^{-1}\left(\frac{L_1^2 + L_2^2 - P^2}{2L_1L_2}\right), \qquad \alpha = cos^{-1}\left(\frac{L_1^2 + P^2 - L_2^2}{2L_1P}\right),$$

$$\beta = tan^{-1}(h/r), \qquad \theta_3 = \beta \pm \alpha, \qquad \theta_2 = \pm(\pi - \gamma)$$

After this, the orientation was calculated. First the orientation of frame 3 in euler form was calculated and subtracted from the input euler rotation. This result was then fed to the last 2 joint frames to calculate inverse kinematics for the claw's rotation relative to frame 3, which, given the previous calculations, results in an inverse relative to the base frame. This is similar to how the inverse kinematics for the PUMA 560 was calculated[2].

The needed r values were calculated based on the angles from the previous step:

$$r_{31} = -s_{23} \quad r_{12} = -c_1s_{23} \quad r_{11} = c_1c_{23} \quad r_{21} = s_1c_{23} \quad r_{32} = -s_1s_{23} \quad r_{33} = 0$$

Then the euler angles of of frame 3 can be calculated:

$$\left\{if\ r_{31} = -1,\ \beta_3 = \frac{\pi}{2},\ \alpha_3 = 0,\ \gamma_3 = arctan\left(\frac{r_{12}}{r_{22}}\right)\right\}$$

$$\left\{if\ r_{31} = 1,\ \beta_3 = -\frac{\pi}{2},\ \alpha_3 = 0,\ \gamma_3 = -\ arctan(\frac{r_{12}}{r_{22}})\right\}$$

$$\left\{else,\ \beta_3 = arctan(\frac{-r_{31}}{\sqrt{r_{11}^2 + r_{21}^2}}),\ \alpha_3 = arctan(\frac{r_{21}}{r_{11}}),\ \gamma_3 = \frac{\pi}{2}\right\}$$

Next the relative euler angles can be found with the target or input angles:

$$\beta_6 = \beta_t - \beta_2 \quad \alpha_6 = \alpha_t - \alpha_2 \quad \gamma_6 = \gamma_t - \gamma_2$$

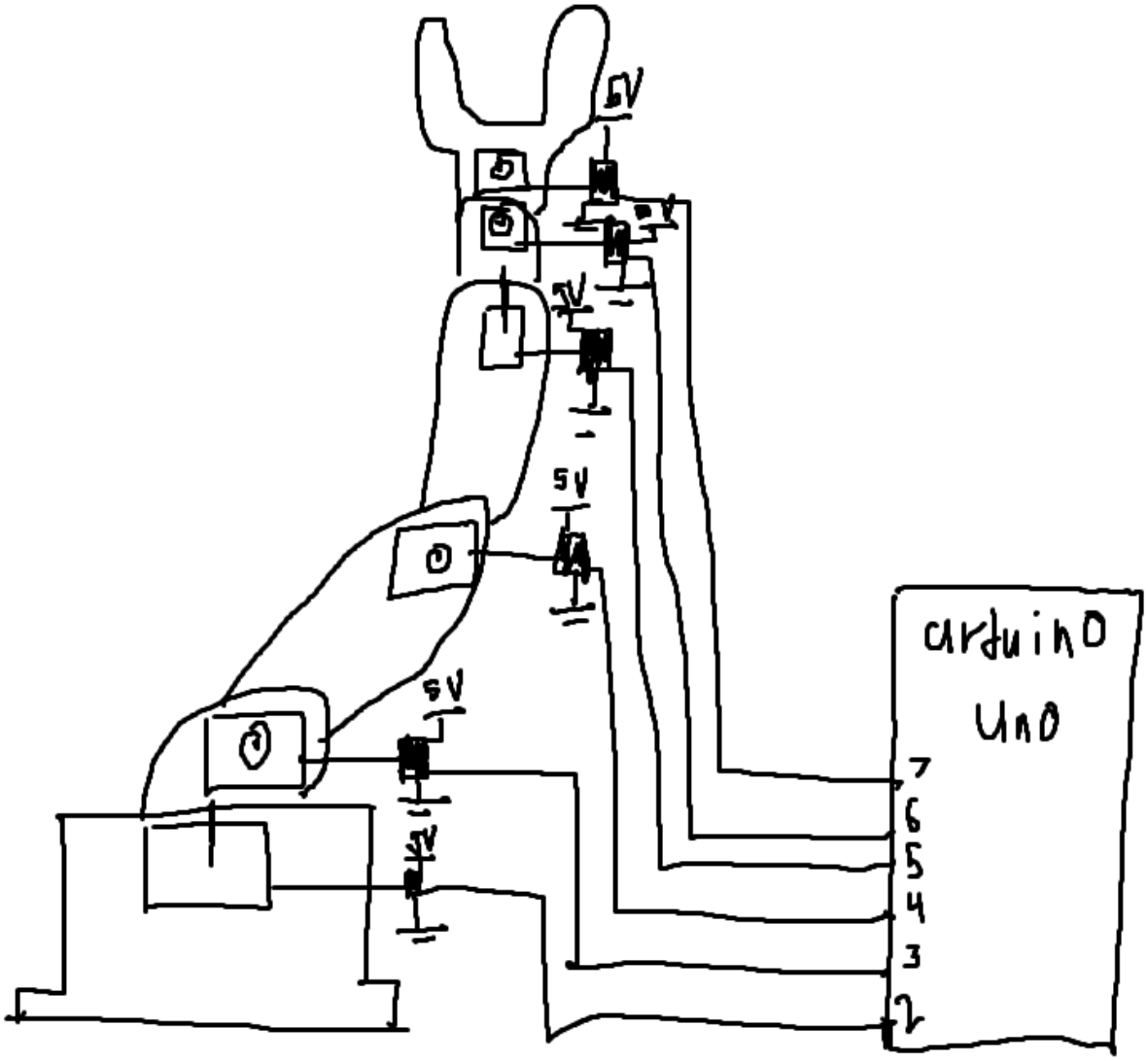
Where subscript t denotes input. With these angles we can calculate the values for θ_4 and θ_5 as shown bellow:

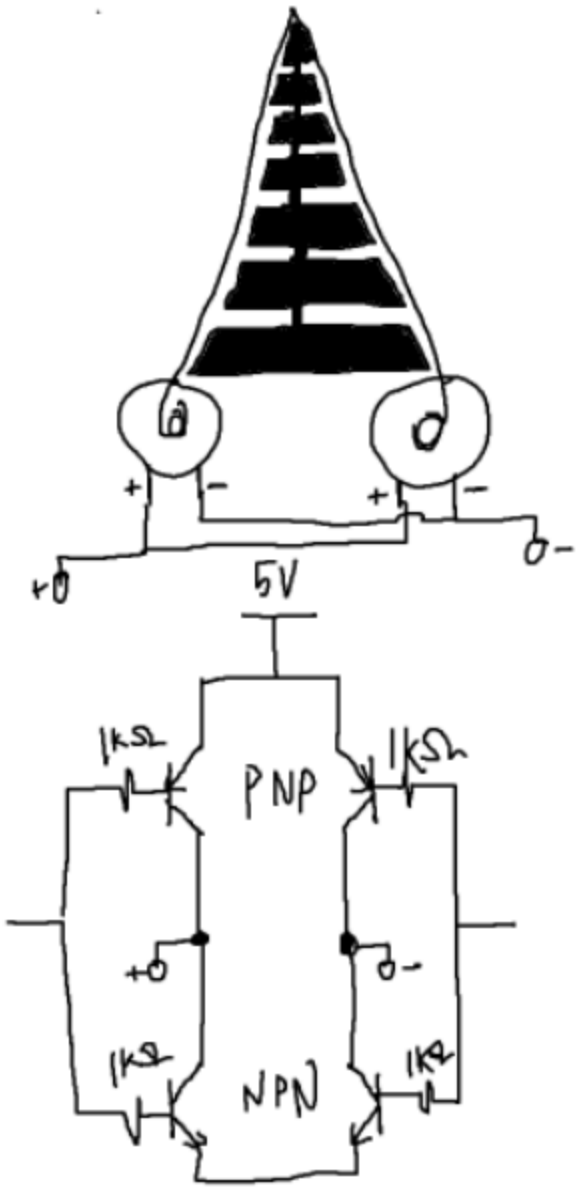
$$\theta_4 = arctan(-\ (\frac{c_{\alpha_6} s_{\beta_6} s_{\gamma_6} - s_{\alpha_6} c_{\gamma_6}}{s_{\beta_6} s_{\gamma_6}})) \text{ and } \theta_5 = arctan(\frac{s_{\alpha_6} c_{\beta_6}}{s_{\alpha_6} s_{\beta_6} c_{\gamma_6} - s_{\alpha_6} s_{\gamma_6}})$$

For the soft robot, the team 3D-printed hexagon-shaped links; they were all connected in the middle for stability. They also each had two holes in them, one on each side, which we threaded fishing line through before tying it all together at the top of the robot. We 3D-printed a small base for the robot, where we attached two motors and empty spools of thread. These were intended to hold onto the fishing line as it wrapped around the motor to bend the robot arm. To control the motors, we set up an H-Bridge circuit to set the direction of the motors. Depending on which side the input signal was on, the motors would rotate clockwise or counter clockwise.

Diagrams of the wiring for the robot arm and soft robot are shown below. Our program code for the robots can be found at the following GitHub repo link: https://github.com/EthanMatth/KinematicsArm .

Block Diagram/Wiring Diagram





Results & Discussion

The final version of the robot arm was quite accurate, albeit a bit shaky. Since only 2 amps of power were supplied to the robot (as opposed to the 10 amps other teams provided), it was only able to move a maximum of two servos at a time before wearing itself thin. As a result,

there were some cases where attempting to pick up a rubber duck with the use of 4 servos at once would cause the robot to swing downwards while holding it. This was a slight problem during the competition, since the ducks were picked up but not lifted high enough to be dropped into the box. The soft robot could twist itself in both directions for a while, before the fishing line on one side got tangled up in its motor.

Conclusions

By the end of the semester, our robot arm was successfully able to pick up a rubber duck based on a given input and place it in a box nearby. We were also able to develop a working inverse kinematics program that uses sliders to determine x, y, z coordinates. This way, we could watch our robot arm follow a line on the x-, y-, or z-axis with ease. As for our soft robotics project, we were able to show that it can curl up to grab things like an octopus tentacle would, so long as it's oriented in the right direction for that.

Acknowledgement

We'd like to express gratitude to our professor, Dr. Ngo, for instructing us in how to calculate forward and inverse kinematics on the PUMA robot to then apply these methods on our own robots, as well as providing us with the files we needed to 3D-print our robotic arm. We'd also like to thank everyone in the Florida Polytechnic University Makerspace for permitting us to use their workshop as a meeting place and giving us access to their many resources including spare nuts/screws.

References

[1] Z. Wang , N. M. Freris, and X. Wei, “SpiRobs: Logarithmic spiral-shaped robots for versatile grasping across scales: Device,” Cell Press Journal, https://www.cell.com/device/fulltext/S2666-9986(24)00603-3?rss=yes&amp=&amp= (accessed Dec. 2, 2025).

[2] K. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge: Cambridge University Press, 2024.