Robotics Final Abstract

The objective of our final project was to create a robotic arm manipulator. We viewed a successful robotic arm manipulator as meeting a few specific metrics: a full implementation of mechanical and electrical design, a functional software-control interface featuring forward kinematics, and the ability for path planning that implements inverse kinematics and object avoidance.

We deemed customizability an important feature for the robot, so we planned its mechanical design around 3D-printability. Motor dimensions were measured and modeled in SolidWorks CAD. Links were designed to interface with the motors and support directional torques and loads.The links were also designed to be standardized, so that many of the parts could be reprinted in order to insert any number of additional links without further CAD modeling. After designing the robot arm, DH parameters were retrieved from the CAD assembly. Much thought was also put into the electrical design of the robot arm (detailed specs are found in the presentation). In order to provide sufficient power and control to each servo motor, a LiPo battery and voltage converter were used.

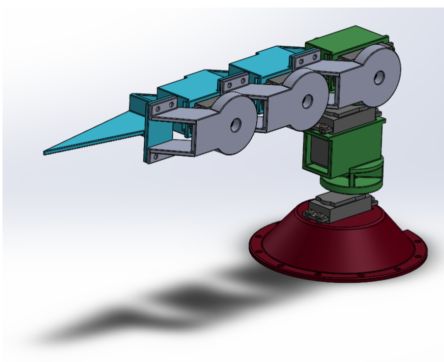

Fig. 1: CAD assembly of robot arm

In order to create an interface for controlling the robot, we modified the ArmPlayer and SerialArm classes to support serial communication with an ESP32 microcontroller that controlled the PWM signals for the servos of the arm. A screenshot of the resulting GUI is shown below.

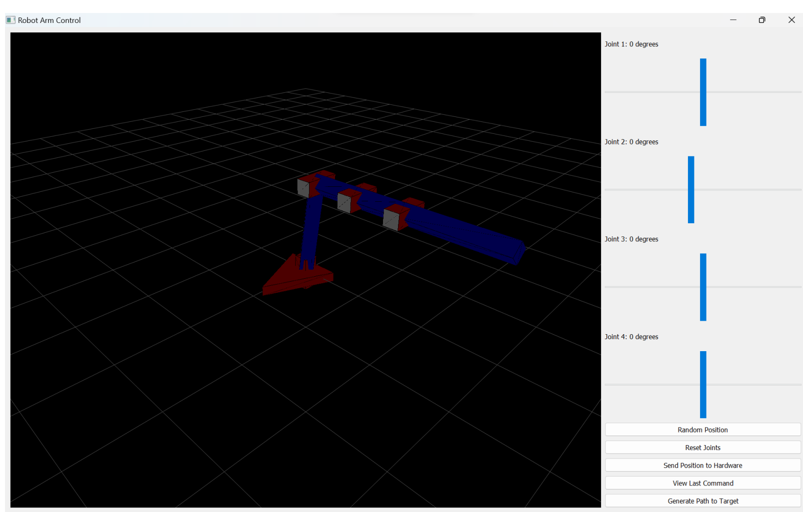

Fig 2. Screenshot of GUI used to control robot arm

After implementing kinematics and control of the arm, we implemented the midterm problem of path planning and object avoidance. This involved considering the floor limits, self collisions, and joint limits. We used the Repulsive and Attractive Fields method for path planning, and the results showed success in reaching the target. Since the simulation showed jerky motions we passed the commands through a 6th-order, 501-window-size Savitzky-Golay Filter to smooth the sequential joint transitions. After smoothing the command angle data, the sequential joint commands were ready to be passed to our robot arm for mechanical implementation.

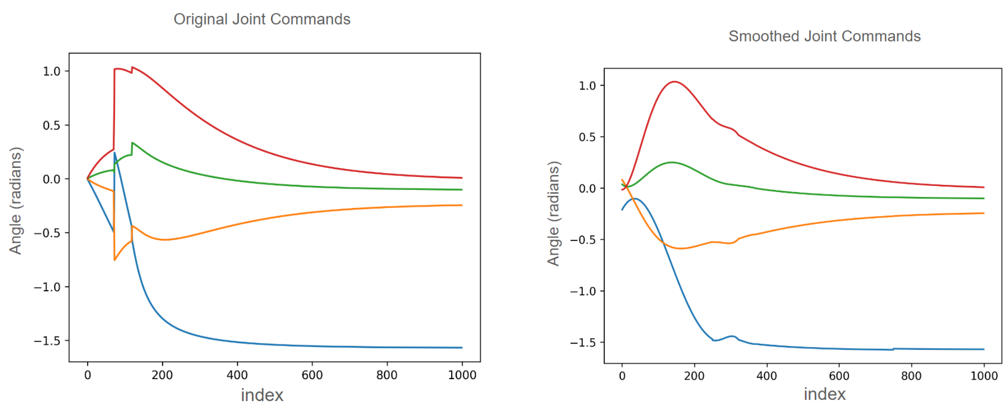

Fig 3. (Left) Joint angles over time, provided by path planning. (Right) The same joint angles after being passed through a Savitzky-Golay Filter..

This solution was implemented in both simulation and also the hardware and though the implementation of the algorithm had certain limitations in terms of reachable points and areas where it gets stuck in hardware, for the most part it performed well enough to reach the majority of the reachable space. An example of the performance along with rough measurements of error in position reached are shown in the youtube link below.

YouTube Demo: https://youtu.be/_hIVG16CuVA