Matthew DiLoreto and Taha Vasowalla
12/15/17
EECE2160
Final Project

Solving Inverse Kinematics on the Robot Arm Using Geometry, Simulation, and Approximation Methods

VIDEO LINK: https://youtu.be/ti46h2ufic0

Inverse kinematics is the process of deciding joint angles for a rigid body to result in a desired end location (Buss).

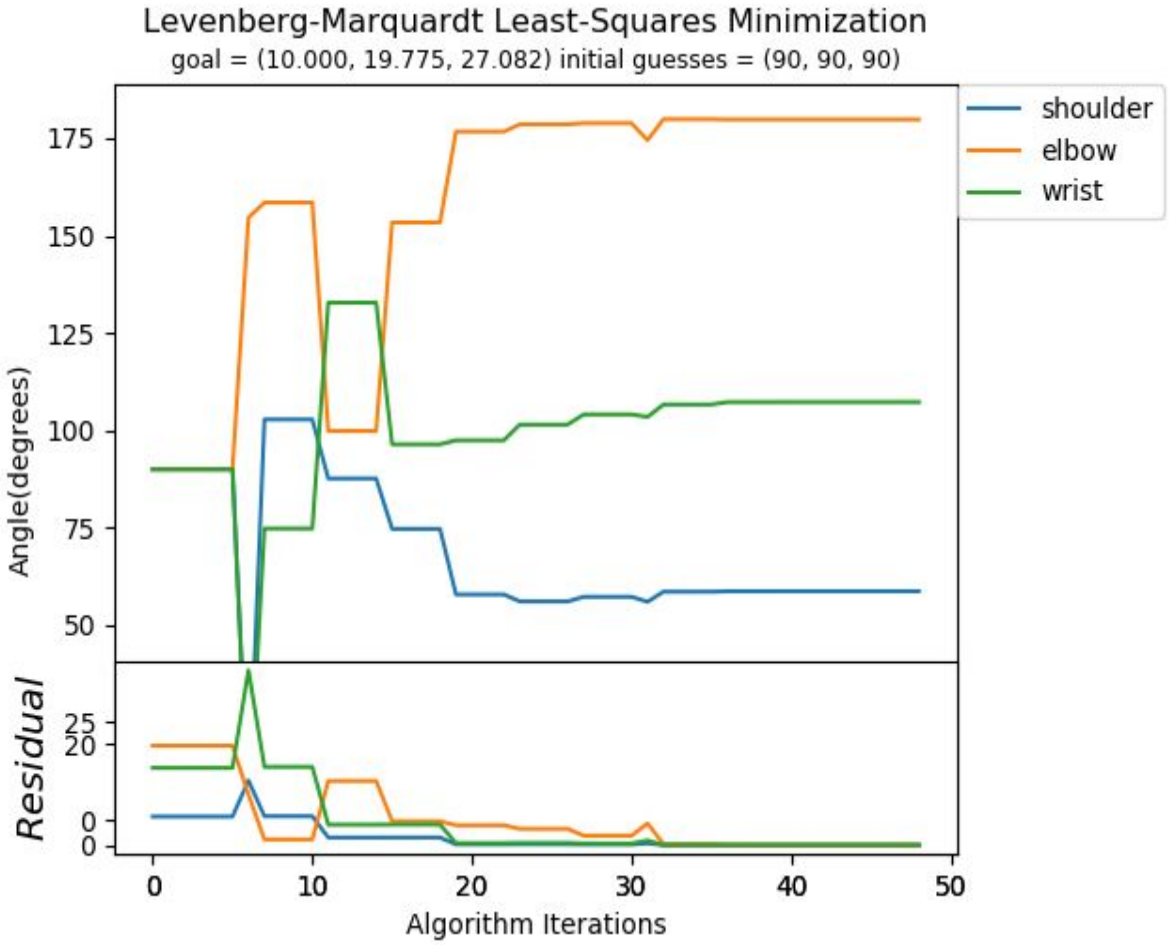
Our first step in calculating the inverse kinematics of the robot arm was making a simulation which we could test and verify easily. We programmed a 3-dimensional version of the arm with configurable inputs for servo angles using the three.js Javascript library and a web browser. We then implemented a few different geometric approaches for calculating angles. All math was performed in spherical coordinates.

The first approach was to fix the angle between the wrist segment and the radius segment to the target position at 90 degrees. This worked fairly well because the math reduced to solving a quadrilateral with 4 known sides and 1 known angle. Unfortunately, this solution is only optimal when the wrist segment is small. Once we measured the actual lengths of the arm segments, we found that the wrist segment was actually as long as the base arm segment. From these values we came up with a new approach - we set the elbow angle equal to the wrist angle and create an isosceles triangle. This produces the full range of motion while staying within the servo angle restrictions.

We used the simulink design of the pwm generator to generate the pwm signals for our servos, since it was trivial to move multiple servos simultaneously using the FPGA on the ZedBoard. We then translated the relevant Javascript code into a C++ class, and replaced the animation functions with the functions provided by the C++ code given in lab 11 to interact with the servos via the mapped memory regions for the pwm generator on the FPGA. Our main method simply uses this class and moves the arm to specific locations in xyz space. For drawing purposes, we implemented a linear interpolation of arm positions from the current point to the desired point in space, such that the arm appears to move in a straight line. The simulation we provide in the video moves the arm in such a way that, if it were able to draw in 3D space, it would draw a rectangular prism. The simulation of this movement is also seen in the video.

Another approach to Inverse Kinematics is the Levenberg Marquardt selectively damped least squares approximation method (Buss). This approach is a gradient descent algorithm that finds the local minimum of the forward kinematic function of the rigid body given appropriate initial guesses to the joint angles. Given the guesses, the algorithm calculates the forward kinetics, or the actual location of the end position of the robot arm, and compares it to the desired location. The angles are gradually changed in such a way that the distance from the actual position to the desired position decreases. When the difference between the two approaches 0,

the algorithm returns the approximated angle values, which can then be passed to either the simulation of the arm, or the the robot arm hardware. In the video, we show the process the algorithm takes to calculate the angles.



The figure above shows the iterations of the algorithm. The base angle is solved for as a constant, so only the remaining 3 servos must be approximated. As we can see, the angles are initially each set to 90°, and the algorithm changes them over time until the residuals approach 0. These calculated angles can then be passed to either the arm hardware or the simulation of the arm, resulting in the desired end location.

References

Buss, Samuel R. “Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares Methods.” 17 Apr. 2004, www.math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf.