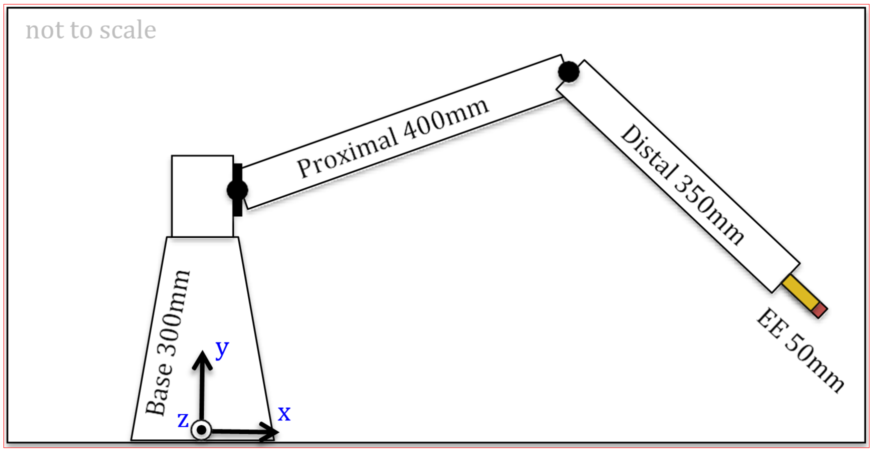
ADDITIVE AUTOMATIONS

C++ Engineer / Robotics Engineer

Take-Home Challenge

Use of external libraries is allowed but must be limited to maths packages. Robotics or kinematics libraries are not allowed.

We have a robot manipulator consisting of a base, a universal joint, a proximal link, a revolute joint, a distal link and a lipstick end effector. The base is 300mm tall and supports the vertical axis of the universal joint. The proximal link is 400mm long and the distal link is 350mm long. The lipstick end effector adds 50mm to the length of the distal link.

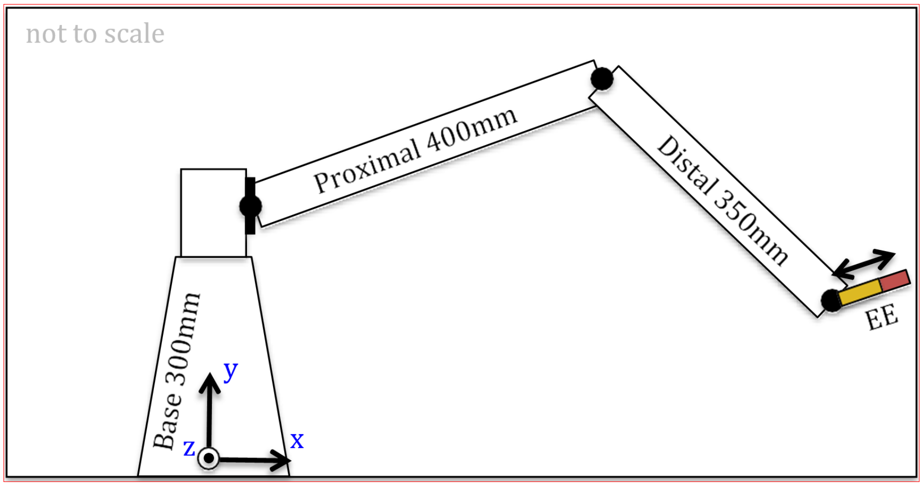


Part 1 – Forward kinematics

Write a C++ script that calculates the position of the tip of the end effector based on the known angular positions of the three degrees of freedom. The output should be Cartesian coordinates in the base origin frame of reference. The script should warn of any collisions with the floor.

Part 2 – Analytical inverse kinematics

The universal joint has a range of -90° to +90° for both axes. The revolute joint has a range of -135° to +135°. The neutral position has both links parallel to x. Write a C++ script that calculates the angles of each of the three degrees of freedom required to reach some desired end effector Cartesian coordinates. Use an analytical method. The script should handle out-of-reach situations and collisions with the floor.



Part 3 – Numerical inverse kinematics

A revolute joint is added between the distal link and the end effector (range of -135° to +135°). An actuator is added to extend and retract the lipstick (from 50mm to 100mm) as a prismatic joint. Write a C++ script that calculates the angles of each of the five degrees of freedom required to reach some desired end effector Cartesian coordinates. Use an iterative optimisation method. The script should handle out-of-reach situations, singularities & collisions with the floor.