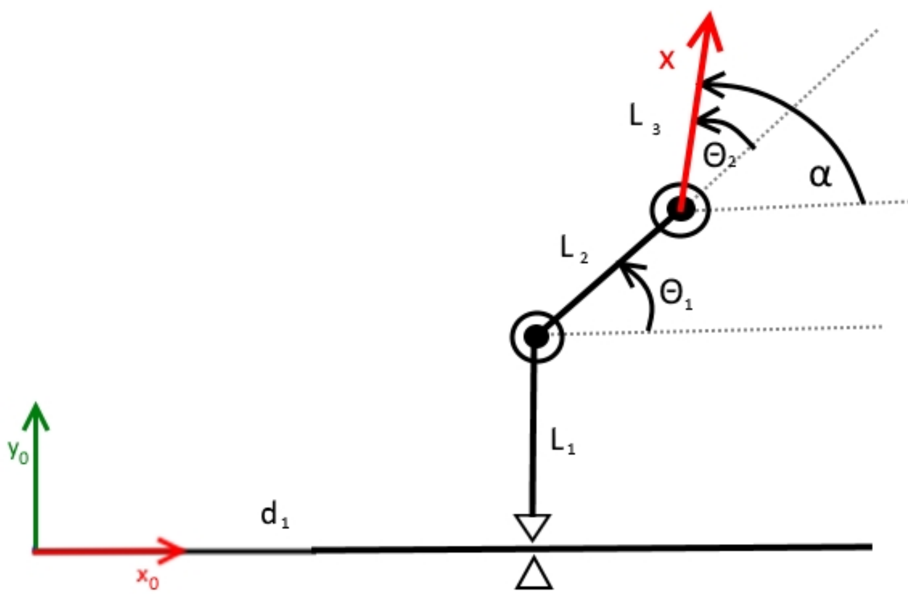
Project 2 Report

Cat Ockman's Team

Inverse Kinematic Model



$$X = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} = \begin{bmatrix} d1 + L2cos\Theta 1 + L3cos(a) \\ L1 + L2sin\Theta 1 + L3sin(a) \\ \Theta 1 + \Theta 2 \end{bmatrix} = f(q)$$

$$x = d_1 + L_2\cos(\Theta_1) + L_3\cos(\alpha)$$

$$y = L_1 + L_2\sin(\Theta_1) + L_3\sin(\alpha)$$

$$\alpha = \Theta_1 + \Theta_2$$

$$\Theta_1 = \sin^{-1}\left(\frac{y - L1 - L3sin(\alpha)}{L2}\right)$$

$$\Theta_2 = \alpha - \Theta_1$$

$$d_1 = x - L_2\cos(\Theta_1) - L_3\cos(\alpha)$$

$$
{}^{0}_{3}T = \begin{bmatrix} cos\alpha & -sin\alpha & 0 & x \\ sin\alpha & cos\alpha & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

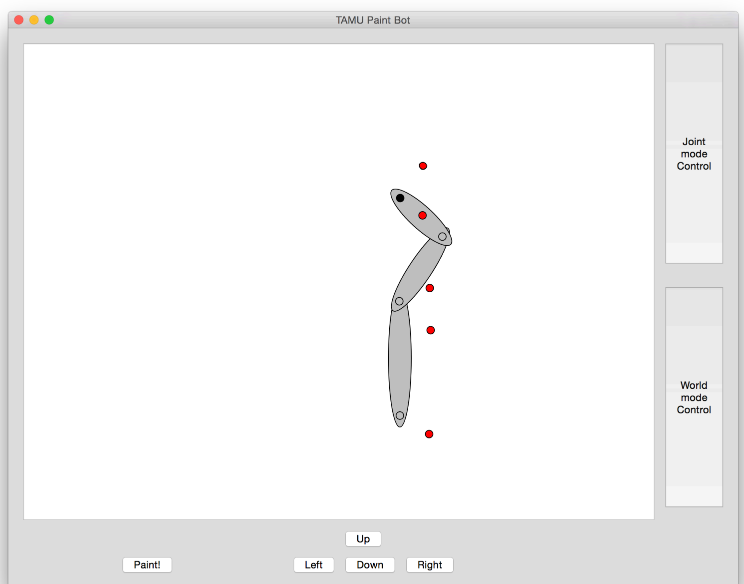
For this project, we used the above kinematic models to create a simulation of a PRR robot using reverse kinematics. The user has the option to switch between the forward kinematics mode (as presented for Project 1) and the inverse kinematics mode(as presented for Project 2). One of the challenges with this project was dealing with the multiple solutions and more complex constraints from dealing with inverse kinematics.

A major constraint:

$$(L1 - L2)^2 <= x^2 + y^2 <= (L1+L2)^2$$

This inequality ensures that the robot does not get stuck trying to move into an impossible state. When this state is reached, the robot uses strict lateral motion (on the prismatic joint) to move into a state where motion is again possible. This ensures that all area within the reach of the robot can be painted.

The completed interface:



The buttons on the right side of the screen allow the user to switch between forward kinematics control and inverse kinematics control. The screen shown above is the inverse kinematics interface. The forward kinematics interface is unchanged from Project 1.