MITMECHE

2.12/2.120
Introduction to Robotics

Mar. 2-3, 2023

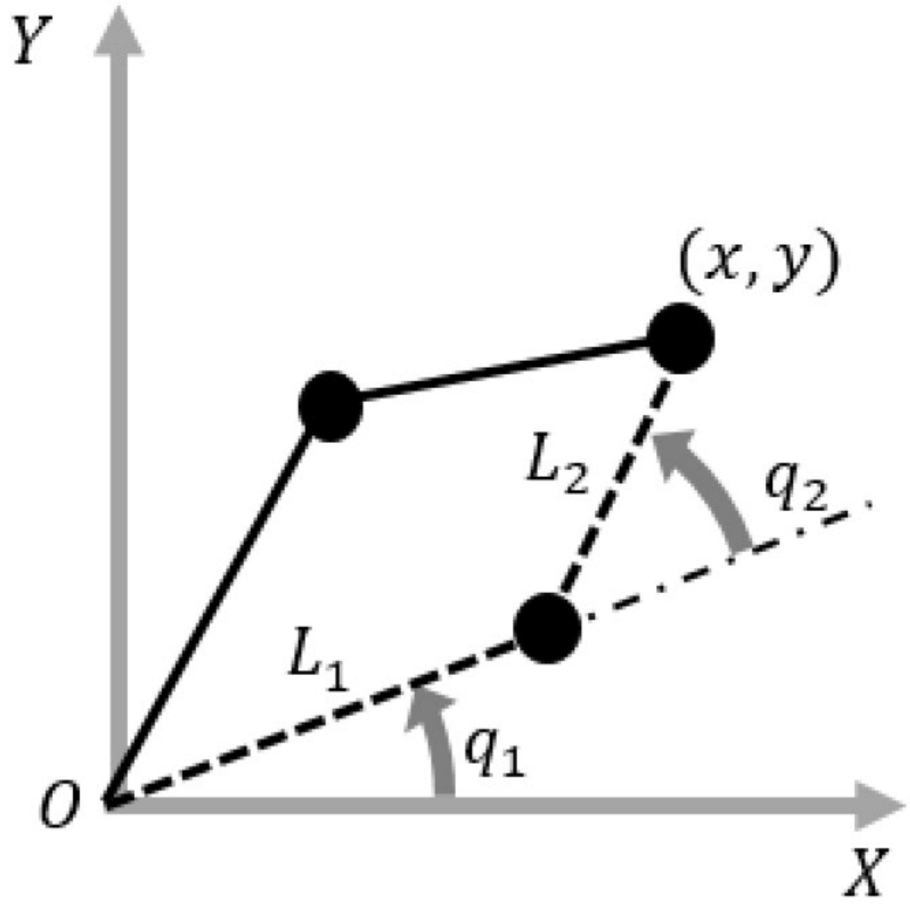
Lab 4: 2.12 Robot Arm Trajectory Control

Objectives

- Implement forward, inverse kinematics, and trajectory planning on an actual 2 DOF robot arm.
- Fine-tune your system to achieve the best possible circular trajectory.
- Optional: create your own trajectory.

- Term Project: Brainstorm ideas.

FK and IK Equations



Forward kinematics equations:

$$x = L_1 \cos q_1 + L_2 \cos(q_1 + q_2)$$

$$y = L_1 \sin q_1 + L_2 \sin(q_1 + q_2)$$

Inverse kinematics equations:

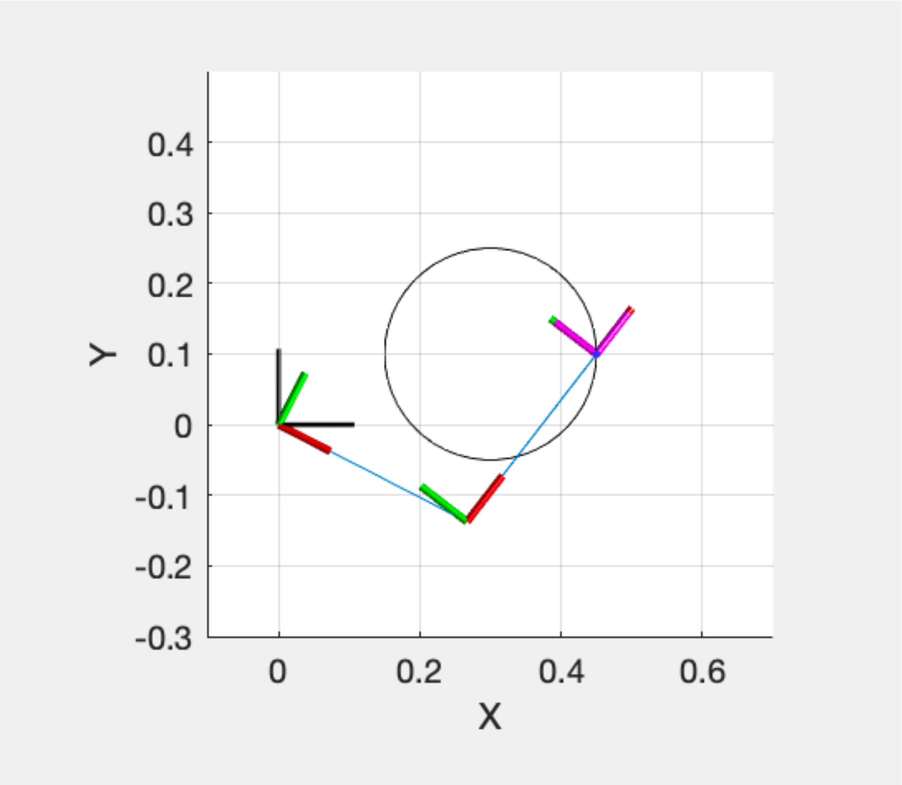
$$q_2 = \pm 2 \cdot \mathrm{atan2}\left(\sqrt{(L_1 + L_2)^2 - (x^2 + y^2)}, \sqrt{(x^2 + y^2) - (L_1 - L_2)^2}\right),$$

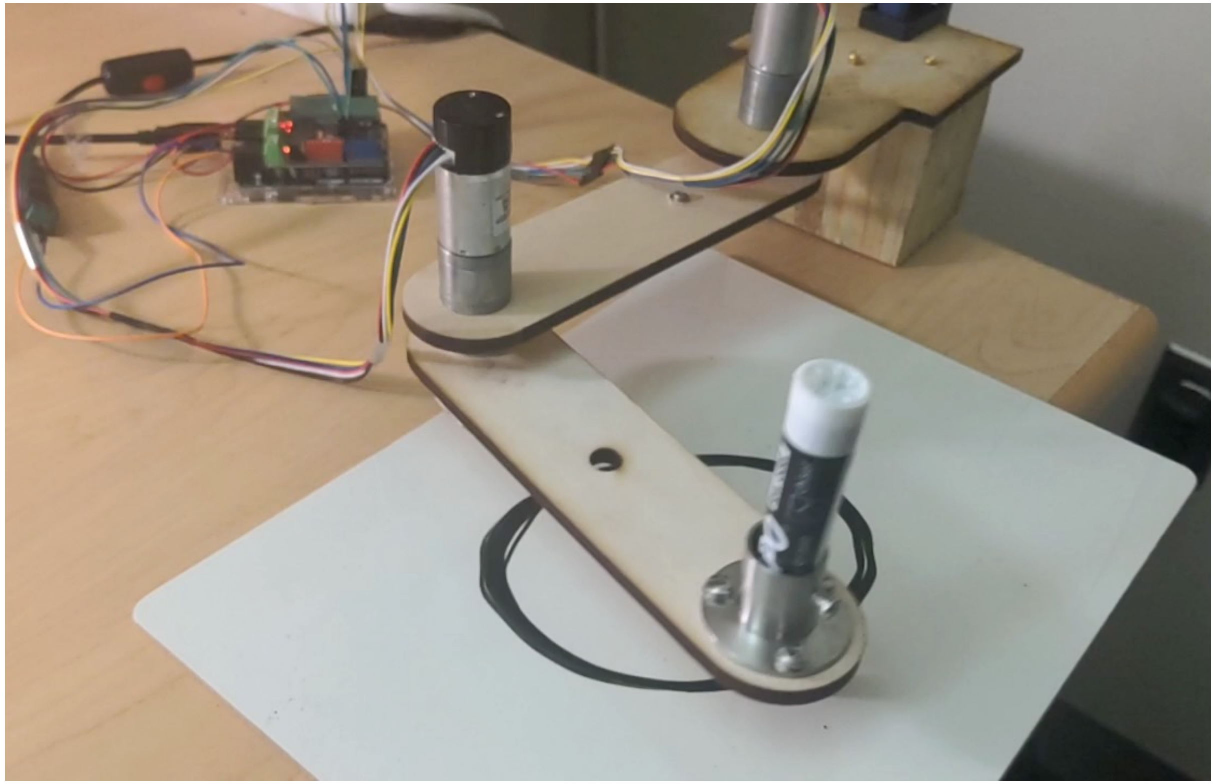
$$q_1 = \mathrm{atan2}(y, x) - \mathrm{atan2}(L_2 \sin q_2, L_1 + L_2 \cos q_2),$$

Trajectory Planning

- Refer to Matlab example: https://www.mathworks.com/help/robotics/ug/2d-inverse-kinematics-example.html
- Program a circular trajectory.

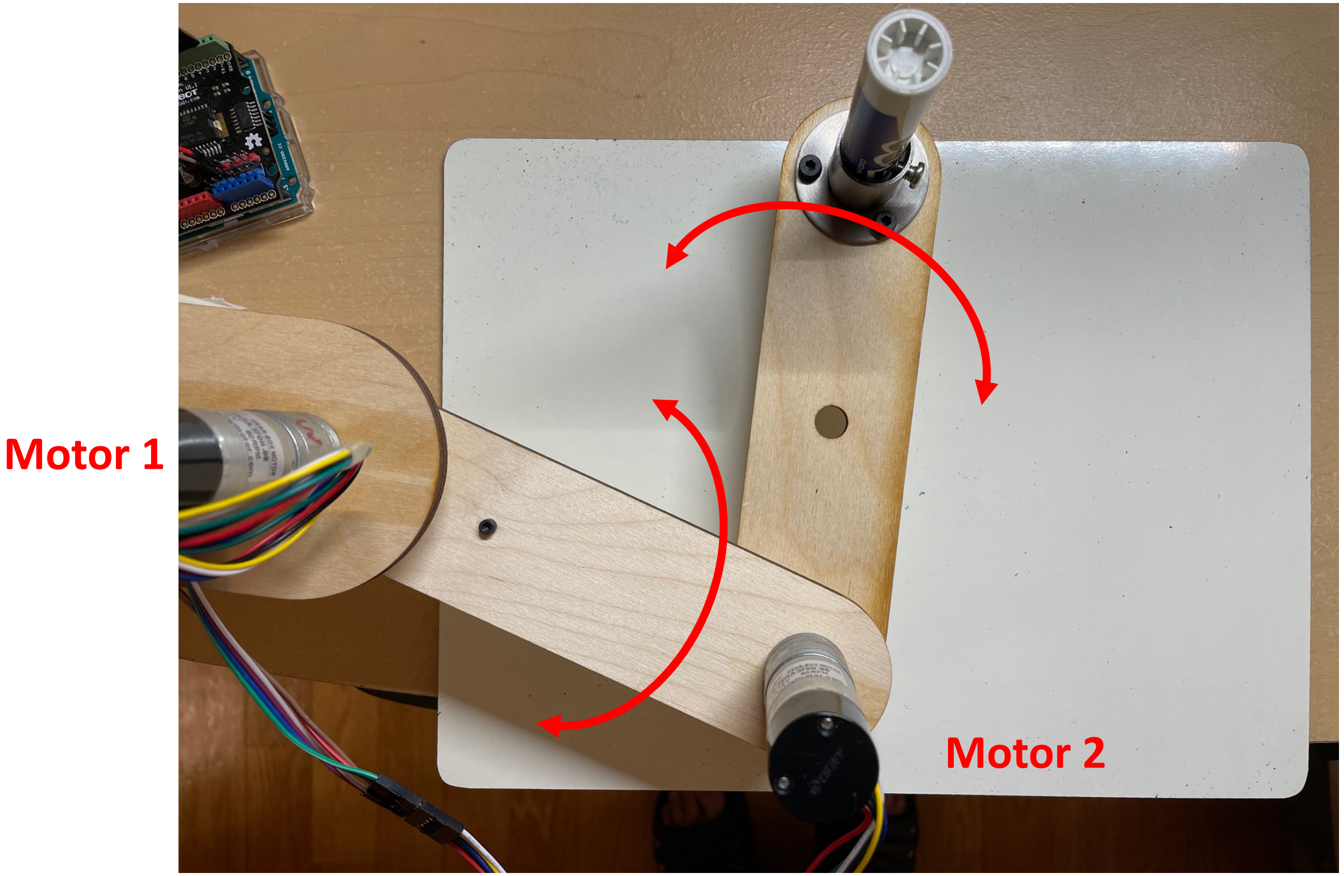
https://www.youtube.com/watch?v=4_bsCFdCCDM





Controller Gains

- The two motors require different controller gains.



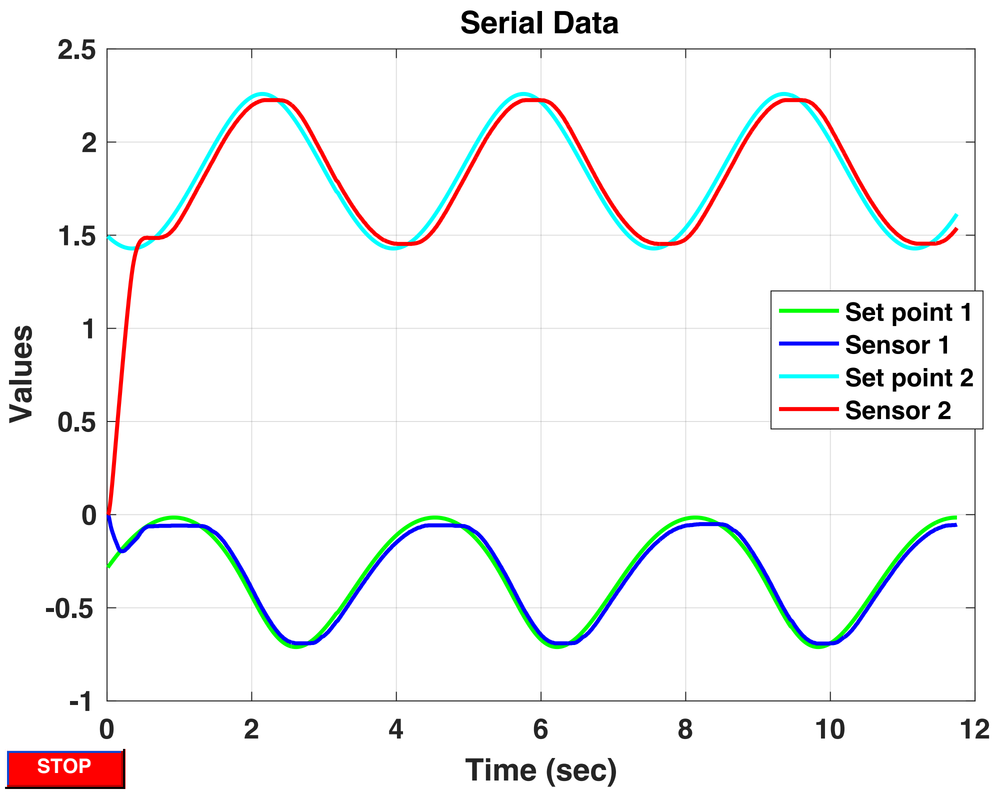
Moment of inertia

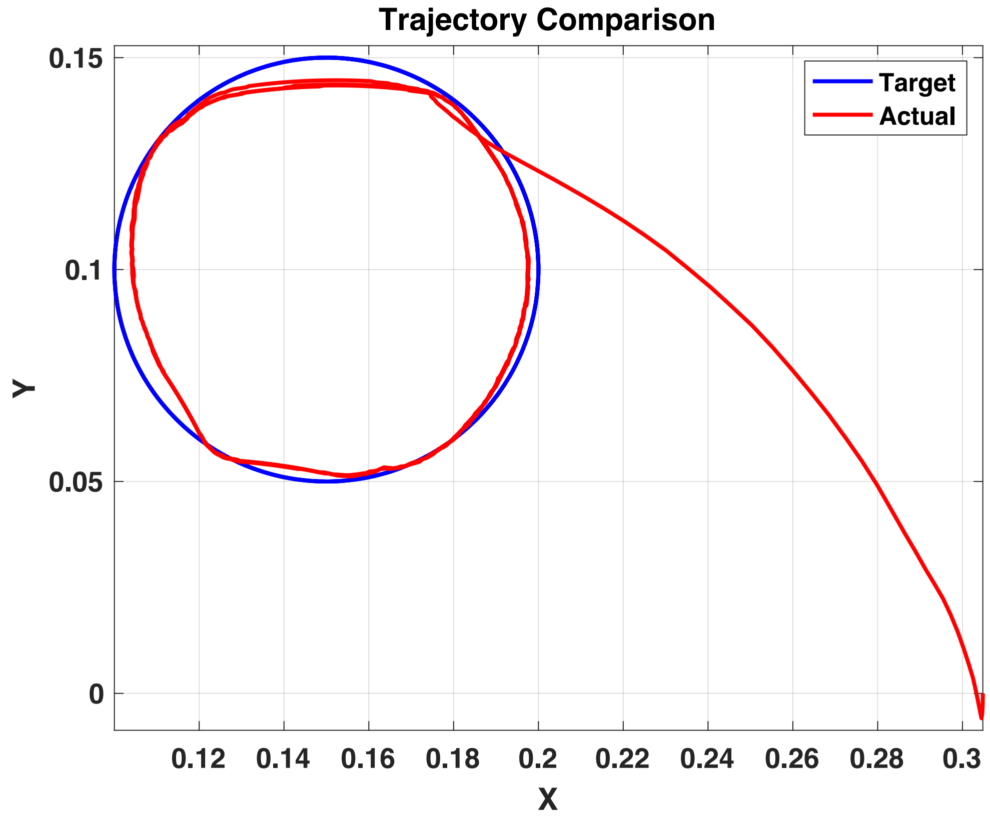
$$I \propto mL^2$$

Fine-tune Robot Arm

- Mechanical assembly.
- Motors 1 and 2 controller gains.
- Sampling period.
- Waypoint spacing.

Use SerialRead2.m to capture the following plots.





Code Files and Deliverables

- Arduino code in GitHub: https://github.com/mit212/lab4_2023.
- Demonstrate to the lab staff your best circular trajectory.