CS 369: Introduction to Robotics

Prof. Thao Nguyen

Spring 2026

HAVERFORD COLLEGE

Outline for today

- Iterative IK
- Controllers

Outline for today

- Iterative IK
- Controllers

Position -> velocity

$$p = F(q)$$

end effector position — FK function — joint configuration

$$\frac{dp}{dt} = \frac{dF(q)}{dt} = \frac{dF(q)}{dq}\frac{dq}{dt}$$

velocity — J(q)

$$\frac{dq}{dt} = J^{-1}(q)\frac{dp}{dt}$$

Jacobian matrix

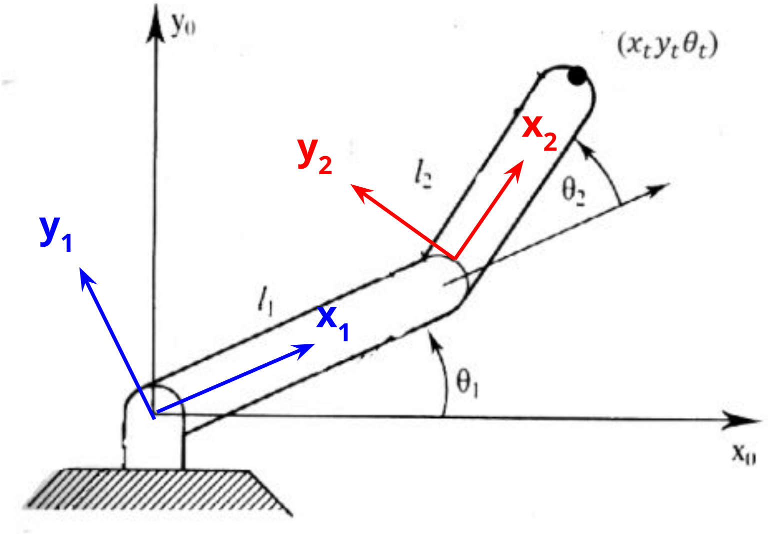
- Refers to the derivative of a vector-valued function of several variables.

- Example: $F(q) = \begin{bmatrix} x \\ y \end{bmatrix} \quad q = [\theta_1 \quad \theta_2 \quad \theta_3]^T$

$$J(q) = \frac{dF(q)}{dq} = \begin{bmatrix} \frac{dx}{dq} \\ \frac{dy}{dq} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \end{bmatrix}$$

Practice

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \left(\begin{bmatrix} l_1 \\ 0 \end{bmatrix} + \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) \end{bmatrix} \begin{bmatrix} l_2 \\ 0 \end{bmatrix} \right)$$



What is the Jacobian matrix for this robot?

Iterative IK

$$\frac{dq}{dt} = J^{-1}(q)\frac{dp}{dt}$$

tolerance

$$\text{while } \|p_{tar} - F(q)\| \geq \varepsilon:$$
$$dp = p_{tar} - F(q)$$
$$dq = J^{-1}(q) \times dp$$
$$q \mathrel{+}= dq * \alpha$$

step size

Iterative IK

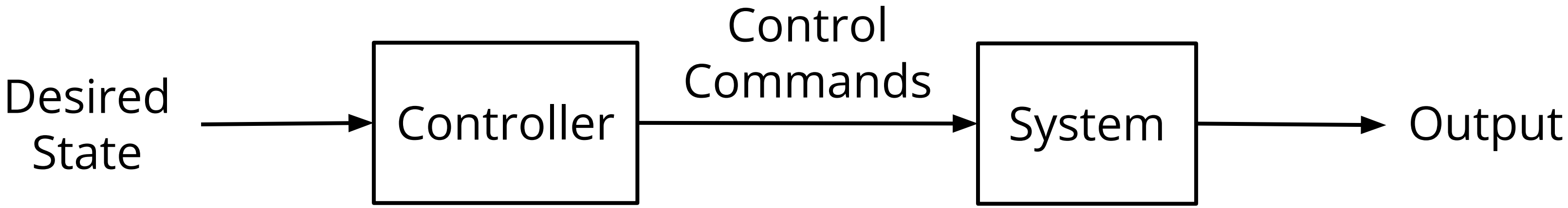
- Flexible, can handle complex / high DOF systems
- Requires multiple iterations
- Dependent on hyperparameters
- Approximate solution
- Finds one local solution

Outline for today

- Iterative IK
- Controllers

Controllers

Control the robot's movement by sending signals to the robot motors.



Open-loop controller