ECE599 Homework-1

Fei Liu

April 11, 2025

1 Inverse Kinematics (IK) for a Planar Manipulator

Objective

Implement an inverse kinematics (IK) controller for a planar 3-link robot manipulator. Your goal is to compute the joint configuration $\mathbf{x} \in \mathbb{R}^3$ such that the end-effector reaches a desired pose $\boldsymbol{\mu} \in \mathbb{R}^2 \times \mathbb{S}^1$, where the target includes both 2D position and orientation.

Mathematical Background

The forward kinematics of a planar manipulator maps the joint configuration $\mathbf{x} \in \mathbb{R}^3$ (i.e., joint angles) to the end-effector pose:

$$\mathbf{f}^{\mathrm{ee}}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^2 \times \mathbb{S}^1,$$

where:

- $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$: end-effector position (x, y),
- $f_3(\mathbf{x})$: end-effector orientation (angle) on the unit circle $\mathbb{S}^1$.

Residual Definition: Given a target pose $\boldsymbol{\mu} \in \mathbb{R}^2 \times \mathbb{S}^1$, define the residual function as:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}^{\mathrm{ee}}(\mathbf{x}) - \boldsymbol{\mu}.$$

To account for the periodic nature of orientation, the angular component of the residual is computed using the logarithmic map on $\mathbb{S}^1$:

$$f_3(\mathbf{x}) - \mu_3 = \mathrm{Im}\left(\log\left(e^{-i\mu_3} \cdot e^{i f_3(\mathbf{x})}\right)\right).$$

Gauss–Newton Update: We minimize the residual using a Gauss–Newton method, updating the joint configuration iteratively:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{J}^{\dagger}(\mathbf{x}_k)\left(\boldsymbol{\mu} - \mathbf{f}^{\mathrm{ee}}(\mathbf{x}_k)\right),$$

where $\mathbf{J}^{\dagger}$ denotes the pseudoinverse of the Jacobian $\mathbf{J}(\mathbf{x})$.

Control-Law Formulation: Alternatively, the update can be expressed in continuous-time form as a velocity-based controller:

$$\dot{\mathbf{x}} = \mathbf{J}^{\dagger}(\mathbf{x})\left(\boldsymbol{\mu} - \mathbf{f}^{\mathrm{ee}}(\mathbf{x})\right) \cdot \frac{\alpha}{\Delta t},$$

where:

- $\alpha \in \mathbb{R}_+$ is a scalar gain,
- Δt is the time step.

Tasks

1. Implement the forward kinematics function $\mathbf{f}^{ee}(\mathbf{x})$ to compute the end-effector's position and orientation from joint angles.

2. Implement the residual vector using the logarithmic map for the angular error.

3. **Derive the analytical Jacobian $\mathbf{J}(\mathbf{x})$** for the 3-link planar robot. Express each row of the Jacobian in terms of the partial derivatives of the end-effector pose with respect to the joint angles.

4. Implement the inverse kinematics loop using the pseudoinverse $\mathbf{J}^{\dagger}$ and iteratively move the end-effector to the target.

5. Visualize the manipulator's joint positions and trajectory toward the target in 2D space.

6. **Numerical Jacobian estimation (Section 5.1):** Implement the numerical Jacobian using finite differences:
$$J_{i,j} \approx \frac{f_i(x^{(j)}) - f_i(x)}{\Delta}, \quad x_k^{(j)} = \begin{cases} x_k + \Delta, & \text{if } k = j \\ x_k, & \text{otherwise} \end{cases}$$
Compare it to your derived analytical Jacobian using the Frobenius norm $\|J_{\text{num}} - J_{\text{ana}}\|_F$. Try this at several random configurations.

7. **Task-priority IK using nullspace projection (Section 5.2):** Use a task-priority controller:
$$\dot{x} = \mathbf{J}^{\dagger}(x)\,\dot{f} + \mathbf{N}(x)\,g(x), \quad \mathbf{N}(x) = \mathbf{I} - \mathbf{J}^{\dagger}(x)\mathbf{J}(x)$$
Choose a secondary task $g(x) = -\nabla x_1$ to minimize the first joint angle. Visualize and compare the trajectory with and without the secondary task.