Outline

• Inverse Kinematics Problem

• Analytical Solution for PUMA-Type Arm

• Numerical Inverse Kinematics
Inverse Kinematics Problem

- **Inverse Kinematics Problem**: Given the forward kinematics $T(\theta), \theta \in \mathbb{R}^n$ and the target homogeneous transform $X \in SE(3)$, find solutions $\theta$ that satisfy

$$T(\theta) = X$$

- Multiple solutions may exist; they are challenging to characterize in general

- This lecture will focus on:
  - Simple illustrating example
  - Analytical solution for PUMA-type arm
  - Numerical solution using the Newton-Raphson method
Example: 2-Link Planar Open Chain

- 2-link planar open chain: considering only the end-effector position and ignoring its orientation, the forward kinematics is

\[
\begin{bmatrix}
  x \\
y
\end{bmatrix} = \begin{bmatrix}
  L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \\
  L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2)
\end{bmatrix} = f(\theta_1, \theta_2)
\]

- Inverse Kinematics Problem: Given \((x, y)\), find \((\theta_1, \theta_2) = f^{-1}(x, y)\)

- Inverse Kinematics Solution:

\[
\begin{align*}
\text{Righty Solution:} & \quad \theta_1 = \gamma - \alpha, \quad \theta_2 = \pi - \beta \\
\text{Lefty Solution:} & \quad \theta_1 = \gamma + \alpha, \quad \theta_2 = \beta - \pi
\end{align*}
\]

where

\[
\gamma = \text{atan2}(y, x), \quad \beta = \cos^{-1}\left(\frac{L_1^2 + L_2^2 - x^2 - y^2}{2L_1L_2}\right)
\]

\[
\alpha = \cos^{-1}\left(\frac{x^2 + y^2 + L_1^2 - L_2^2}{2L_1\sqrt{x^2 + y^2}}\right)
\]
Analytical Inverse Kinematics: PUMA-Type Arm

6R arm of PUMA type:

- The first two shoulder joint axes intersect orthogonally at a common point
- Joint axis 3 lies in $\hat{x}_0 - \hat{y}_0$ plane and is aligned with joint axis 2
- Joint axes 4,5,6 (wrist joints) intersect orthogonally at a common point (the wrist center)
- For PUMA-type arms, the inverse Kinematics problem can be decomposed into inverse position and inverse orientation subproblems
PUMA-Type Arm: Inverse Position Subproblem

- Given desired configuration \( X = (R, p) \in SE(3) \). Clearly, \( p = (p_x, p_y, p_z) \) depends only on \( \theta_1, \theta_2, \theta_3 \). Solving for \( (\theta_1, \theta_2, \theta_3) \) based on given \( (p_x, p_y, p_z) \) is the inverse position problem.

- Assume that \( p_x, p_y \) not both equal to zero. They can be used to determine two solutions of \( \theta_1 \)

- Solutions:
PUMA-Type Arm: Inverse Position Subproblem

- Determining $\theta_2$ and $\theta_3$ is inverse kinematics problem for a planar two-link chain

$$\cos(\theta_3) = \frac{\|p\|^2 - d_1^2 - a_2^2 - a_3^2}{2a_2a_3} = D$$

$$\theta_3 = \text{atan2} \left( \pm \sqrt{1 - D^2}, D \right)$$

- The solutions of $\theta_3$ corresponds to the “elbow-up” and “elbow-down” configurations for the two-link planar arm.

- Similarly, we can find:

$$\theta_2 = \text{atan2} \left( p_z, \sqrt{p_x^2 + p_y^2 - d_1^2} \right) - \text{atan2} \left( a_3 \sin \theta_3, a_2 + a_3 \cos \theta_3 \right)$$
PUMA-Type Arm: Inverse Orientation Subproblem

• Now we have found \((\theta_1, \theta_2, \theta_3)\), we can determine \((\theta_4, \theta_5, \theta_6)\) given the end-effector orientation.

• The forward kinematics can be written as:

\[
e^{[S_4] \theta_4} e^{[S_5] \theta_5} e^{[S_6] \theta_6} = e^{-[S_3] \theta_3} e^{-[S_2] \theta_2} e^{-[S_1] \theta_1} X M^{-1}
\]

where the right-hand side is now known (denoted by \(\tilde{R}\)).

• For simplicity, we assume joint axes of 4,5,6 are aligned in the \(\hat{z}_0, \hat{y}_0\) and \(\hat{x}_0\) directions, respectively; Hence, the \(\omega_i\) components of \(S_4, S_5, S_6\) are:

\[
\omega_4 = (0, 0, 1), \omega_4 = (0, 1, 0), \omega_6 = (1, 0, 0)
\]

• Therefore, the wrist angles can be determined as the solution to

\[
Rot(\hat{z}, \theta_4) Rot(\hat{y}, \theta_5) Rot(\hat{x}, \theta_6) = \tilde{R}
\]

• This corresponds to solving for ZYX Euler angles given \(\tilde{R} \in SO(3)\), whose analytical solution can be found in Appendix B.1.1 of the textbook.
Numerical Inverse Kinematics

- Inverse kinematics problem can be viewed as finding roots of a nonlinear equation:

\[
T(\theta) = X
\]

- Many numerical methods exist for finding roots of nonlinear equations.

- For inverse kinematics problem, the target configuration \( X \in SE(3) \) is a homogeneous matrix. We need to modify the standard root finding methods. But the main idea is the same.

- We first recall the standard Newton-Raphson method for solving \( x = f(\theta) \), where \( \theta \in \mathbb{R}^n \) and \( x \in \mathbb{R}^m \). Then we will discuss how to modify the method to numerically solve the inverse kinematics problem.
Newton-Raphson Method

- Given $f : \mathbb{R}^n \to \mathbb{R}^m$, we want to find $\theta_d$ such that $x_d = f(\theta_d)$.

- Taylor expansion around initial guess $\theta^0$:

$$x_d = f(\theta_d) = f(\theta^0) + \frac{\partial f}{\partial \theta} \bigg|_{\theta^0} (\theta_d - \theta^0) + \text{h.o.t.}$$

- Let $J(\theta^0) = \frac{\partial f}{\partial \theta} \bigg|_{\theta^0}$ and drop the h.o.t., we can compute $\Delta \theta$ as

$$\Delta \theta = J^\dagger(\theta^0)(x_d - f(\theta^0))$$

- $J^\dagger$ denotes the Moore-Penrose pseudoinverse

- For any linear equation: $b = Az$, the solution $z^* = A^\dagger b$ falls into the following two categories:

  1. $Az^* = b$

  2. $\|Az^* - b\| \leq \|Az - b\|, \forall z \in \mathbb{R}^n$
Newton-Raphson Method

Algorithm:

- Initialization: Given \( x_d \in \mathbb{R}^m \) and an initial guess \( \theta^0 \in \mathbb{R}^n \), set \( i = 0 \) and select tolerance \( \epsilon > 0 \).
- Set \( e = x_d - f(\theta^i) \). While \( \| e \| > \epsilon \):
  1. Set \( \theta^{i+1} = \theta^i + J^\dagger(\theta^i)e \)
  2. Increment \( i \).

- If \( f(\theta) \) is a linear function, the algorithm will converge to solution in one-step.

- If \( f \) is nonlinear, there may be multiple solutions. The algorithm tends to converge to the solution that is the "closest" to the initial guess \( \theta^0 \).
From Newton Method to Inverse Kinematics Solution

- Given desired configuration $X = T_{sd} \in SE(3)$, we want to find $\theta_d \in \mathbb{R}^n$ such that $T_{sb}(\theta_d) = T_{sd}$

- At the $i$th iteration, we want to move towards the desired position:
  - In vector case, the direction to move is $e = x_d - f(\theta^i)$
  - Meaning: $e$ is the velocity vector which, if followed for unit time, would cause a motion from $f(\theta^i)$ to $x_d$
  - Thus, we should look for a body twist $\mathcal{V}_b$ which, if followed for unit time, would cause a motion from $T_{sb}(\theta^i)$ to the desired configuration $T_{sd}$.
  - Such a body twist is given by
    \[
    [\mathcal{V}_b] = \log T_{bd}(\theta^i), \quad \text{where} \quad T_{bd}(\theta^i) = T_{sb}^{-1}(\theta^i)T_{sd}
    \]

To achieve a desired body twist, we need the joint rate vector:

\[
\Delta \theta = J_b^\dagger(\theta^i) \mathcal{V}_b
\]
Numerical Inverse Kinematics Algorithm

- **Algorithm:**
  - Initialization:
    - Given: $T_{sd}$ and initial guess $\theta^0$
    - Set $i = 0$ and select a small error tolerance $\epsilon > 0$
    - Set $[V_b] = \log T_{bd}(\theta^i)$. While $\|V_b\| > \epsilon$:
      1. Set $\theta^{i+1} = \theta^i + J_b^\dagger(\theta^i)V_b$
      2. Increment $i$

- An equivalent algorithm can be developed in the space frame, using the space Jacobian $J_s$ and the spatial twist $V_s = [\text{Ad}_{T_{sb}}]V_b$
More Discussions

•