ECE5463: Introduction to Robotics

Lecture Note 8: Inverse Kinematics

Prof. Wei Zhang

Department of Electrical and Computer Engineering
Ohio State University
Columbus, Ohio, USA

Spring 2018

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 θ that satisfy

$$T(\theta) = X$$

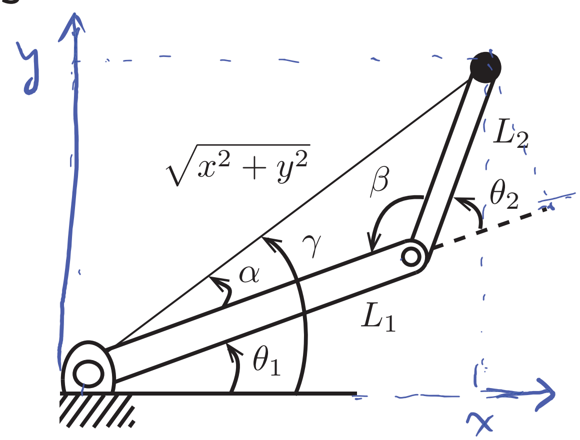
- Multiple solutions may exist; they are challenging to characterize in general

- 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

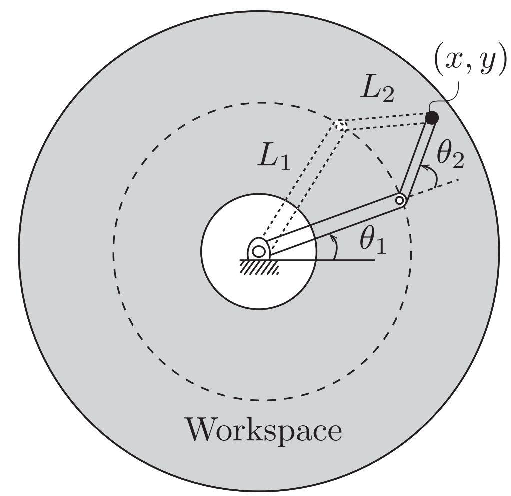
- 2-link planar open chain: considering only the end-effector position and ignoring its orientation, the forward kinematics is

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



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

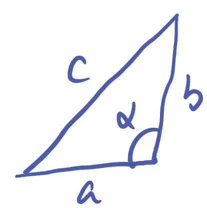
$$\begin{cases} \text{Righty Solution:} & \theta_1 = \gamma - \alpha, \quad \theta_2 = \pi - \beta \\ \text{Lefty Solution:} & \theta_1 = \gamma + \alpha, \quad \theta_2 = \beta - \pi \end{cases}$$



where

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

$$\alpha = \cos^{-1}\left(\frac{x^2 + y^2 + L_1^2 - L_2^2}{2L_1\sqrt{(x^2+y^2)}}\right)$$



Recall : law of cosines: $c^2 = a^2 + b^2 - 2ab\cos(\alpha)$

Analytical Inverse Kinematics: PUMA-Type Arm

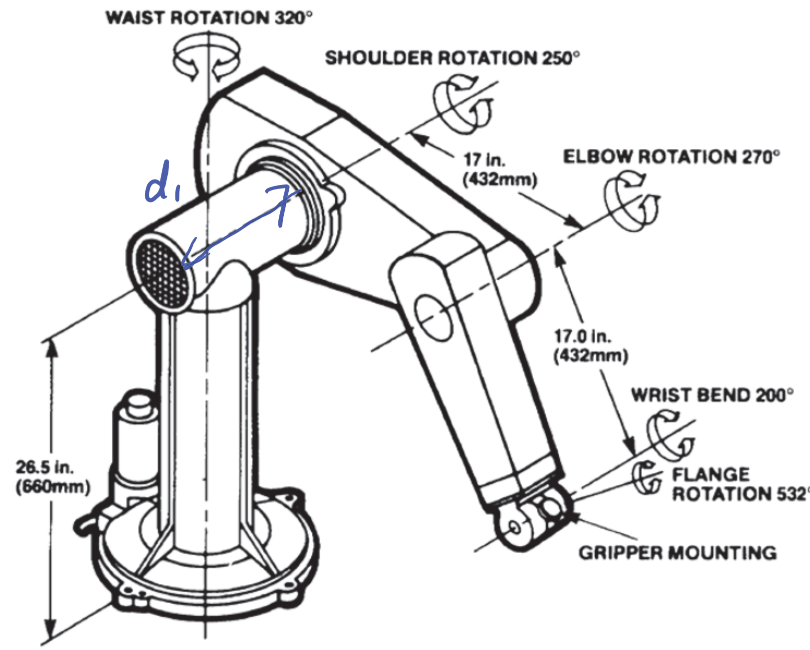
6R arm of PUMA type:

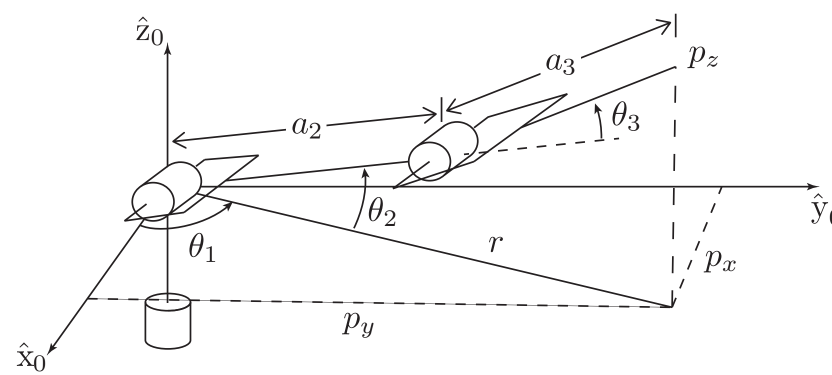
- ~~Two~~ The first shoulder joint axes intersect orthogonally at a common point

- 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

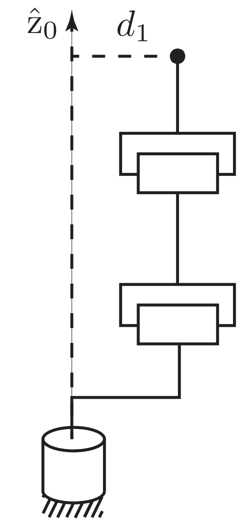




no offset $d_1 = 0$

PUMA-Type Arm: Inverse Position Subproblem

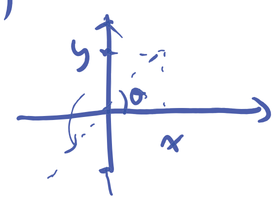
- Given desired configuration $X = (R, p) \in SE(3)$. Clearly, $p = (p_x, p_y, p_z)$ depends only on $\theta_1, \theta_2, \theta_3$. Solving for $(\theta_1, \theta_2, \theta_3)$ based on given (p_x, p_y, p_z) is the inverse position problem.



we need to project p onto x_0-y_0 plane

- Assume that p_x, p_y not both equal to zero. They can be used to determine two solutions of θ_1

– C-style
$\theta \Leftarrow \text{atan2}(y, x)$



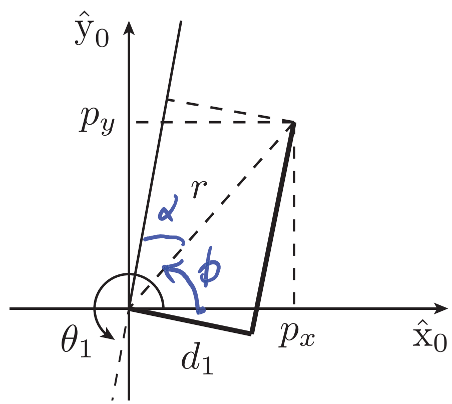
- Solutions: define: $\phi = \text{atan2}(p_y, p_x)$, $r^2 = p_x^2 + p_y^2$

$\alpha = \text{atan2}(d_1, \sqrt{p_x^2 + p_y^2 - d_1^2})$

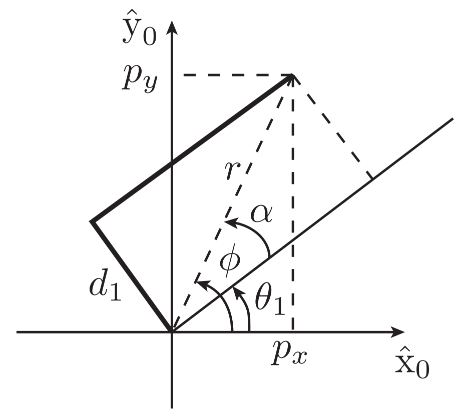
lefty sln: $\theta_1 = \phi - \alpha$

righty sln: $\theta_1 = \pi + \phi + \alpha$

See textbook

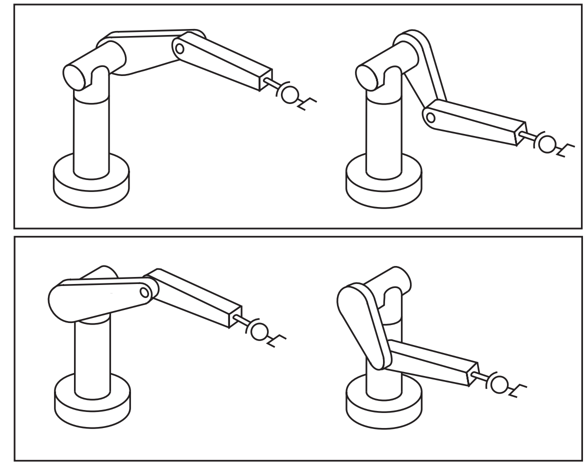


righty - sln



lefty - sln

PUMA-Type Arm: Inverse Position Subproblem



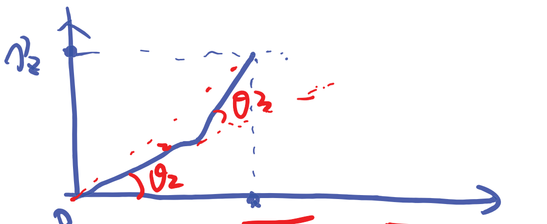
- Determining θ_2 and θ_3 is inverse kinematics problem for a planar two-link chain

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

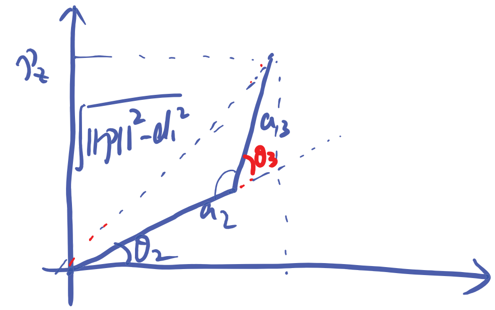
$$\theta_3 = \text{atan2}\left(\pm\sqrt{1-D^2}, D\right)$$

Fix θ_1 :

side view:



$\sqrt{r^2-d_1^2} = \sqrt{p_x^2+p_y^2-d_1^2}$



- The solutions of θ_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

Recall: we are given the target configuration

$$X = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$$

- The forward kinematics can be written as:

$$e^{[\mathcal{S}_4]\theta_4} e^{[\mathcal{S}_5]\theta_5} e^{[\mathcal{S}_6]\theta_6} = \boxed{\left(e^{\cancel{-}[\mathcal{S}_1]\theta_1} e^{\cancel{-}[\mathcal{S}_2]\theta_2} e^{\cancel{-}[\mathcal{S}_3]\theta_3}\right)^{-1} X M^{-1}}$$

$$e^{[S_1]\theta_1} \cdots e^{[S_4]\theta_4} \cdots e^{[S_6]\theta_6} M = X$$

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 ω_i components of $\mathcal{S}_4, \mathcal{S}_5, \mathcal{S}_6$ 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 $\underset{\in \mathbb{R}^n}{\theta_d}$ such that $\underset{\in \mathbb{R}^m}{x_d} = f(\theta_d)$.

- Taylor expansion around initial guess θ^0:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} f_1(\theta_1, \cdots, \theta_n) \\ f_2(\theta_1, \cdots, \theta_n) \\ \vdots \\ f_m(\theta_1, \cdots, \theta_n) \end{bmatrix}$$

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

- Let $J(\theta^0) = \left.\frac{\partial f}{\partial \theta}\right|_{\theta^0}$ $\in \mathbb{R}^{m \times n}$ and drop the h.o.t., we can compute $\Delta\theta$ as

$$J_{ij}(\theta^0) = \frac{\partial f_i}{\partial \theta_j}$$

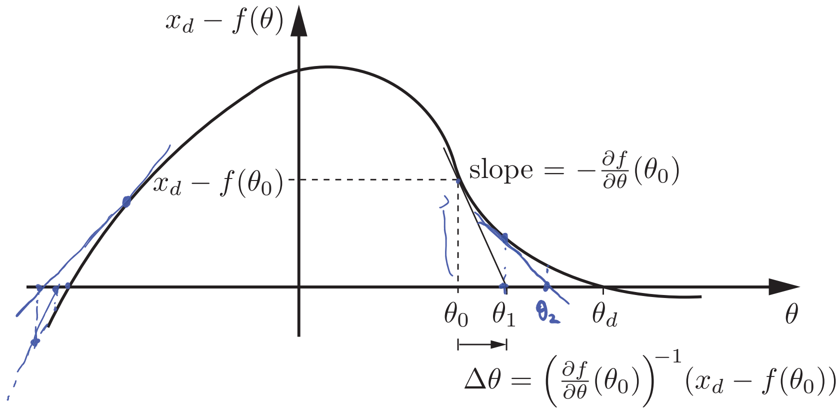
$$\Delta\theta = \underbrace{J^\dagger(\theta^0)}_{A}\overbrace{(x_d - f(\theta^0))}^{b}$$

  - $J^\dagger$ denotes the Moore-Penrose pseudoinverse
  - For any linear equation: $b = Az$, the solution $z^* = A^\dagger b$ falls into the following two categories:
    1. $Az^* = b$: z^* is the smallest norm sln : i.e. $\underbrace{A\hat{z} = b}_{\text{for any } \hat{z}}$ then $\|z^*\| \le \|\hat{z}\|$
    2. $\|Az^* - b\| \le \|Az - b\|, \forall z \in \mathbb{R}^n$ $Az^* \ne b$

Newton-Raphason 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 θ^0

From Newton Method to Inverse Kinematics Solution

- Given desired configuration $X = \overset{T_{sd}}{\cancel{T_{sb}}} \in SE(3)$, we want to find $\theta_d \in \mathbb{R}^n$ such that $T_{sb}(\theta_d) = T_{sd}$ constant matrix
- At the ith iteration, we want to move towards the desired position: $\notin SE(3)$
  - In vector case, the direction to move is $e = x_d - f(\theta^i)$ $\Rightarrow$ becomes $T_{sd} - T_{sb}(\theta^i)$
  - Meaning: e is the velocity vector which, if followed for unit time, would cause a motion from $f(\theta^i)$ to x_d current $x^i = f(\theta^i) \Rightarrow x^i + (x_d - f(\theta^i)) = x_d$
  - 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 $\Rightarrow e^{[\mathcal{V}_b]} = T_{bd}(\theta^i)$

$$[\mathcal{V}_b] = \log\left(T_{bd}(\theta^i)\right), \qquad \text{where } T_{bd}(\theta^i) = T_{sb}^{-1}(\theta^i) T_{sd}$$

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 θ^0
    - Set $i = 0$ and select a small error tolerance $\epsilon > 0$
  - Set $[\mathcal{V}_b] = \log T_{bd}(\theta^i)$. While $\|\mathcal{V}_b\| > \epsilon$:
    1. Set $\theta^{i+1} = \theta^i + J_b^\dagger(\theta^i)\mathcal{V}_b$
    2. Increment i
- An equivalent algorithm can be developed in the space frame, using the space Jacobian J_s and the spatial twist $\mathcal{V}_s = [\mathrm{Ad}_{T_{sb}}]\,\mathcal{V}_b$

More Discussions

-