# Jacobian and Inverse Kinematics

CS 6301 Special Topics: Introduction to Robot Manipulation and Navigation

Professor Yu Xiang

The University of Texas at Dallas

# Manipulator Jacobian

- Space Jacobian $\mathcal{V}_s = \begin{bmatrix} J_{s1} & J_{s2}(\theta) & \cdots & J_{sn}(\theta) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} = J_s(\theta)\dot{\theta}.$

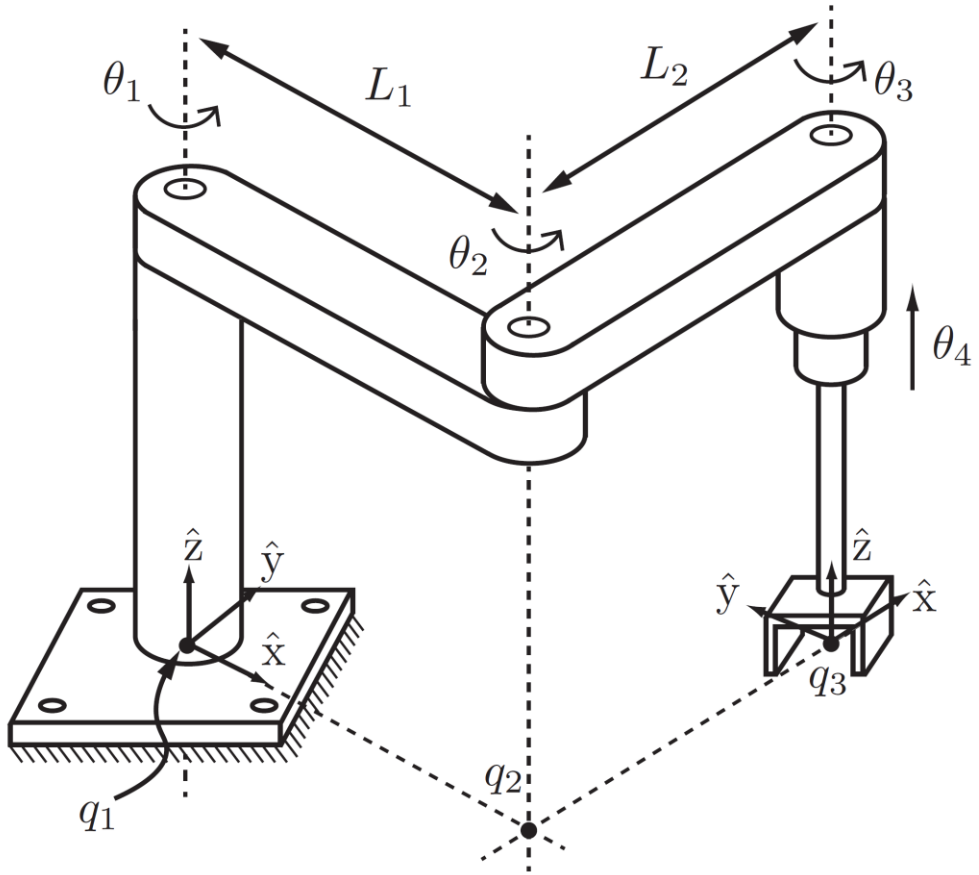
$$J_{s1} = \mathcal{S}_1 \qquad J_{si}(\theta) = \mathrm{Ad}_{e^{[\mathcal{S}_1]\theta_1} \cdots e^{[\mathcal{S}_{i-1}]\theta_{i-1}}}(\mathcal{S}_i) \qquad i = 2, \ldots, n,$$

- Body Jacobian

$$\mathcal{V}_b = \begin{bmatrix} J_{b1}(\theta) & \cdots & J_{bn-1}(\theta) & J_{bn} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} = J_b(\theta)\dot{\theta}$$

$$J_{bn} = \mathcal{B}_n \quad J_{bi}(\theta) = \mathrm{Ad}_{e^{-[\mathcal{B}_n]\theta_n} \cdots e^{-[\mathcal{B}_{i+1}]\theta_{i+1}}}(\mathcal{B}_i) \qquad i = n-1, \ldots, 1$$

# Manipulator Jacobian




a spatial RRRP chain

$$J_s(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & L_1 s_1 & L_1 s_1 + L_2 s_{12} & 0 \\ 0 & -L_1 c_1 & -L_1 c_1 - L_2 c_{12} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J_b(\theta) = [Ad_{T_{bs}}] J_s(\theta)$$

$$[\mathrm{Ad}_T] = \begin{bmatrix} R & 0 \\ [p]R & R \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

# Manipulator Jacobian

- Up to now, we represent the end-effector configuration by $T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$
- Then

$$\dot{T} = \begin{bmatrix} \dot{R} & \dot{p} \\ 0 & 0 \end{bmatrix} \qquad \begin{aligned} [\mathcal{V}_s] &= \dot{T}T^{-1} \\ [\mathcal{V}_b] &= T^{-1}\dot{T} \end{aligned} \qquad \begin{aligned} \mathcal{V}_s &= J_s(\theta)\dot{\theta} \\ \mathcal{V}_b &= J_b(\theta)\dot{\theta} \end{aligned}$$

How to use twist?

For constant twist $T(t) = e^{[\mathcal{V}_s]t} \cdot T(0)$ $\qquad$ $T(t) = T(0) \cdot e^{[\mathcal{V}_b]t}$

For non-constant twist (First-order integration)

$$T(k+1) = e^{[\mathcal{V}_s(k)]\Delta t} \cdot T(k)$$

$$T(k+1) = T(k) \cdot e^{[\mathcal{V}_b(k)]\Delta t}$$

# Analytic Jacobian

- What if we want to use 6 variables for the end-effector configuration?
  - Translation $x, y, z$
  - Rotation: exponential coordinates $r = \hat{\omega}\theta$

$$R(\theta) = e^{[\hat{\omega}]\theta} \qquad R(\theta) = I + \sin\theta[\hat{\omega}] + (1 - \cos\theta)[\hat{\omega}]^2$$

a minimum set of coordinates $$q = \begin{bmatrix} r_x \\ r_y \\ r_z \\ x \\ y \\ z \end{bmatrix}$$

# Analytic Jacobian

- Can we find the Jacobian for $\dot{q} = J_a(\theta)\dot{\theta}$ $\quad q = \begin{bmatrix} r \\ x \end{bmatrix} \in \mathbb{R}^6$

In space frame

- Let's start with the body Jacobian $\mathcal{V}_b = J_b(\theta)\dot{\theta}$

$$\mathcal{V}_b = \begin{bmatrix} \omega_b \\ v_b \end{bmatrix} = J_b(\theta)\dot{\theta} = \begin{bmatrix} J_\omega(\theta) \\ J_v(\theta) \end{bmatrix} \dot{\theta}$$

Previously lectures $v_b = R^T \dot{p}$

$$\dot{x} = R_{sb}(\theta) v_b = R_{sb}(\theta) J_v(\theta)\dot{\theta}$$

$$\dot{r} = A^{-1}(r)\omega_b = A^{-1}(r) J_\omega(\theta)\dot{\theta}$$

$$\begin{aligned} \omega_b &= A(r)\dot{r}, \\ A(r) &= I - \frac{1 - \cos\|r\|}{\|r\|^2}[r] + \frac{\|r\| - \sin\|r\|}{\|r\|^3}[r]^2 \end{aligned}$$

Exercise 5.10 Lynch & Park

# Analytic Jacobian

- Can we find the Jacobian for $\dot{q} = J_a(\theta)\dot{\theta}$ $\quad q = \begin{bmatrix} r \\ x \end{bmatrix} \in \mathbb{R}^6$

In space frame

$$\dot{r} = A^{-1}(r)\omega_b = A^{-1}(r)J_\omega(\theta)\dot{\theta}$$

$$\dot{x} = R_{sb}(\theta)v_b = R_{sb}(\theta)J_v(\theta)\dot{\theta}$$

$$\dot{q} = \begin{bmatrix} \dot{r} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} A^{-1}(r) & 0 \\ 0 & R_{sb} \end{bmatrix} \begin{bmatrix} \omega_b \\ v_b \end{bmatrix}$$

$$J_a(\theta) = \begin{bmatrix} A^{-1}(r) & 0 \\ 0 & R_{sb}(\theta) \end{bmatrix} \begin{bmatrix} J_\omega(\theta) \\ J_v(\theta) \end{bmatrix} = \begin{bmatrix} A^{-1}(r) & 0 \\ 0 & R_{sb}(\theta) \end{bmatrix} J_b(\theta)$$

Can we do $q(t) = q(0) + \dot{q}\,t$ ? No $R(t) = e^{[r(t)]}$

# Inverse Kinematics

- Calculation of the joint coordinates given the position and orientation of its end-effector

End-effector transformation

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \Rightarrow \text{Joint coordinates} \quad \theta$$

# Inverse Kinematics

- For a n degree-of-freedom open chain with forward kinematics
  $T(\theta) \quad \theta \in \mathbb{R}^n$

- Given a homogenous transformation $X \in SE(3)$

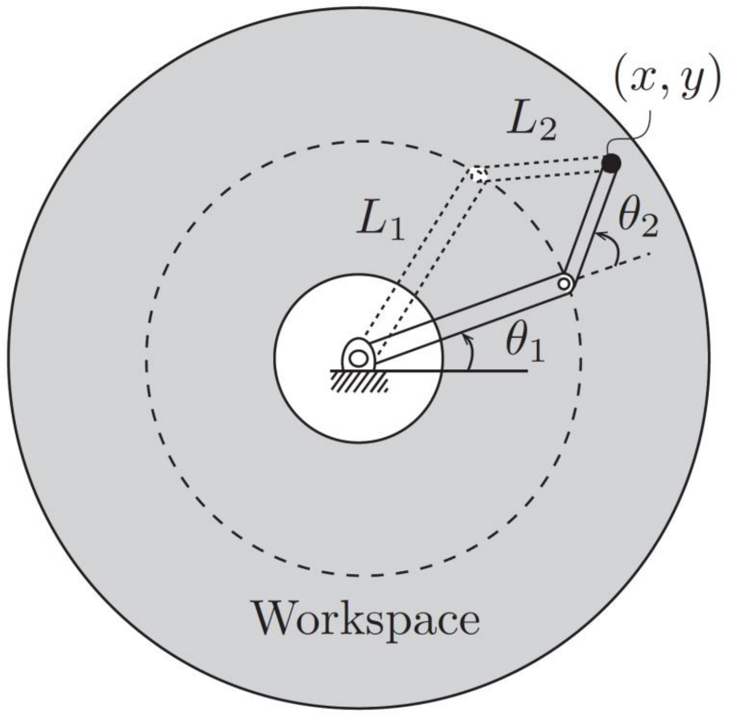
- Find solutions $\theta$ such that $T(\theta) = X$

# Inverse Kinematics

- IK can have multiple solutions
- FK only has a single solution
- Find solutions $\theta$ such that

$$T(\theta) = X$$

- Finding the roots of a nonlinear equation




Analytic Inverse Kinematics

# Numerical Inverse Kinematics Algorithm

- Forward kinematics $x = f(\theta)$ $\quad f : \mathbb{R}^n \rightarrow \mathbb{R}^m$
- Desired end-effector coordinates $x_d$
- Objective function for the Newton-Raphson method

$$g(\theta) = x_d - f(\theta)$$

- Goal

$$g(\theta_d) = x_d - f(\theta_d) = 0$$

- Initial guess $\theta^0$

# Newton-Raphson Method

- Taylor expansion

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

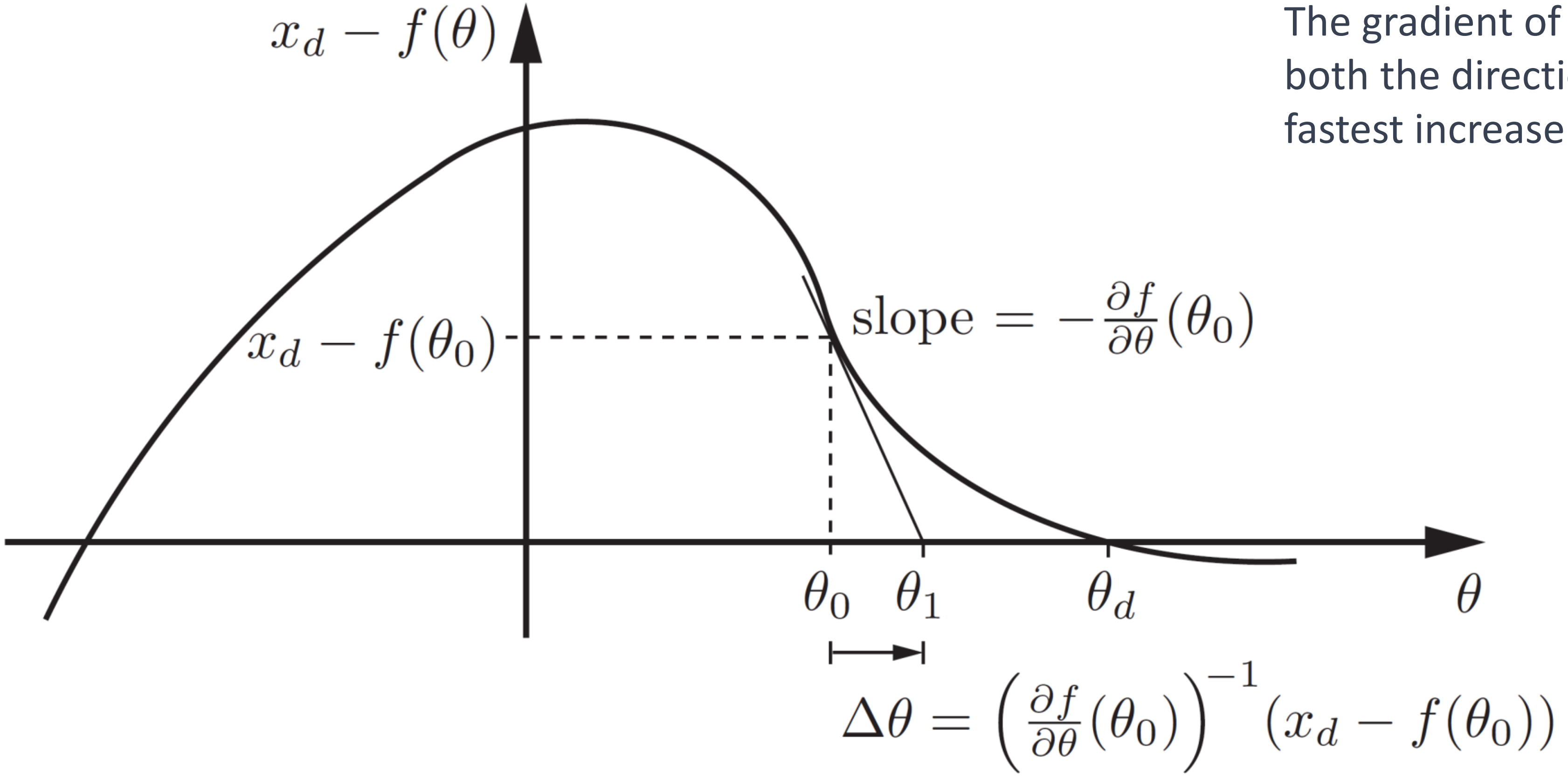
$J(\theta^0) \in \mathbb{R}^{m \times n}$ Jacobian

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

Whe $J(\theta^0)$ is square and invertible $\Delta\theta = J^{-1}(\theta^0)\left(x_d - f(\theta^0)\right)$

$$\theta^1 = \theta^0 + \Delta\theta$$

# Numerical Inverse Kinematics Algorithm




The gradient of a function represents both the direction and rate of the fastest increase of that function.

# Numerical Inverse Kinematics Algorithm

- When J is not invertible, use pseudoinverse $J^\dagger$

$$Jy = z \qquad J \in \mathbb{R}^{m\times n},\ y \in \mathbb{R}^n,\ \text{and}\ z \in \mathbb{R}^m$$

$$y^* = J^\dagger z$$

$$J^\dagger = J^{\mathrm{T}}(JJ^{\mathrm{T}})^{-1} \qquad \text{if } J \text{ is fat, } n > m \text{ (called a right inverse since } JJ^\dagger = I)$$
$$J^\dagger = (J^{\mathrm{T}}J)^{-1}J^{\mathrm{T}} \qquad \text{if } J \text{ is tall, } n < m \text{ (called a left inverse since } J^\dagger J = I).$$

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

# Numerical Inverse Kinematics Algorithm

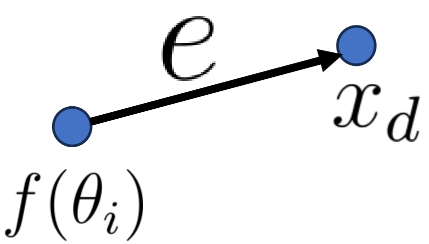
- Newton-Raphson iterative algorithm for inverse kinematics

- Initialization: given $x_d \in \mathbb{R}^m$ , initial guess $\theta^0 \in \mathbb{R}^n$

- Set $e = x_d - f(\theta^i)$ While $||e|| > \epsilon$ for some small $\epsilon$:
  - Set $\theta^{i+1} = \theta^i + J^\dagger(\theta^i)e$
  - Increment $i$

# Numerical Inverse Kinematics Algorithm

- How to achieve a desired end-effector configuration $T_{sd} \in SE(3)$
- Given current configuration $T_{sb}(\theta^i)$
- We cannot simply compute error as $T_{sd} - T_{sb}(\theta^i)$
- Consider error $e = x_d - f(\theta^i)$ as a velocity vector




- Similarly, a body twist $\mathcal{V}_b \in \mathbb{R}^6$ will cause a motion $T_{sb}(\theta^i)$ to $T_{sd}$

# Numerical Inverse Kinematics Algorithm

- How to achieve a desired end-effector configuration $T_{sd} \in SE(3)$
- Current configuration $T_{sb}(\theta^i)$

- Desired configuration $T_{bd}(\theta^i) = T_{sb}^{-1}(\theta^i)T_{sd} = T_{bs}(\theta^i)T_{sd}$
- Body twist $[\mathcal{V}_b] = \log T_{bd}(\theta^i)$ Lynch & Park 3.3.3.2 Matrix Logarithm of Rigid-Body Motions

- Updating rule

Pseudoinverse of the body Jacobian

$$\theta^{i+1} = \theta^i + J_b^{\dagger}(\theta^i)\mathcal{V}_b$$

# Inverse Velocity Kinematics

- Find the joint velocity $\dot{\theta}$ to follow a desired end-effector trajectory $T_{sd}(t)$
- Method 1: uses inverse kinematics to compute $\theta_d(k\Delta t)$

Joint velocity $\dot{\theta} = \big(\theta_d(k\Delta t) - \theta((k-1)\Delta t)\big)/\Delta t$

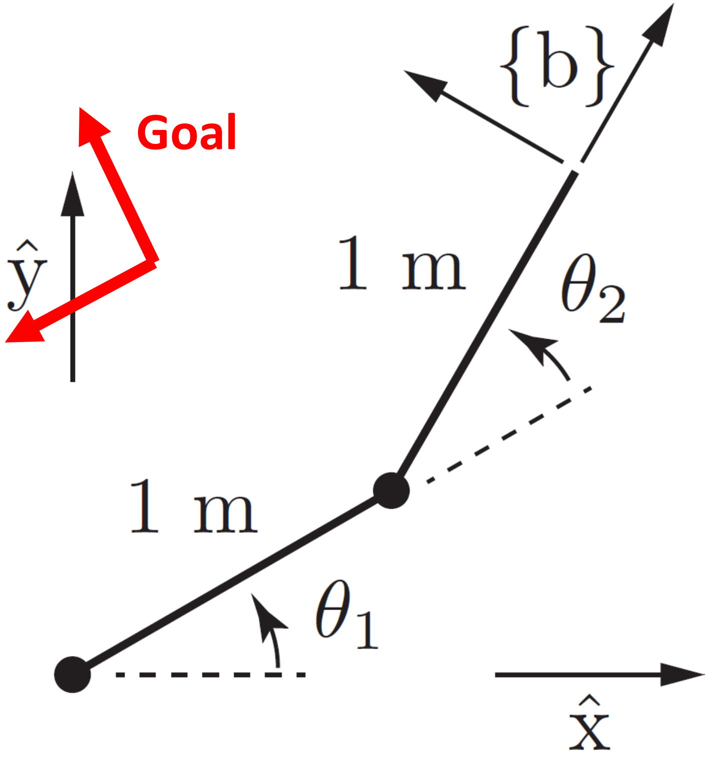
interval $[(k-1)\Delta t, k\Delta t]$

- Method 2: uses $J\dot{\theta} = \mathcal{V}_d$

$$\dot{\theta} = J^{\dagger}(\theta)\mathcal{V}_d$$

$$[\mathcal{V}_d] = \log\left(T_{sd}((k-1)\Delta t)^{-1} T_{sd}(k\Delta t)\right)$$

# Numerical Inverse Kinematics



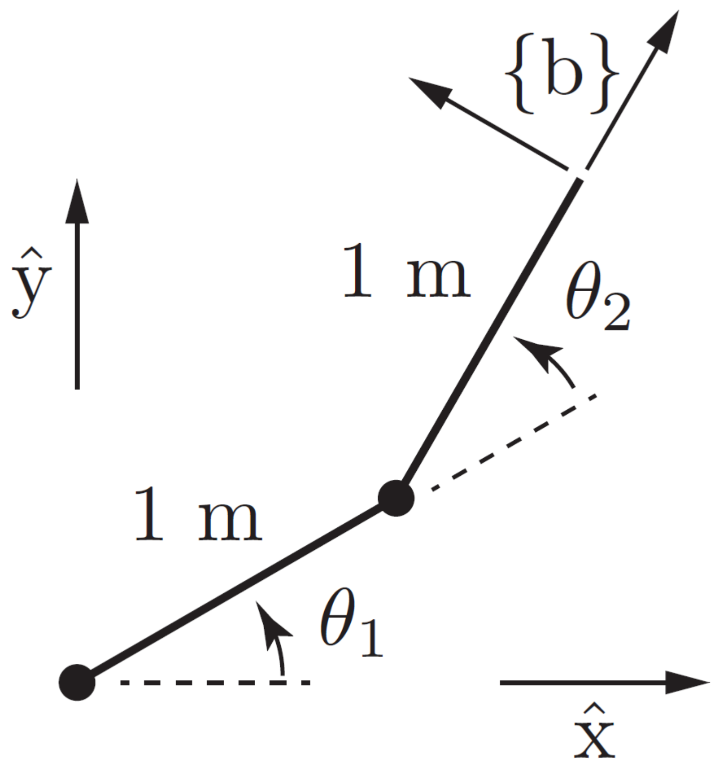

A 2R robot

Goal

$$(x, y) = (0.366\ \mathrm{m}, 1.366\ \mathrm{m})$$

$$T_{sd} = \begin{bmatrix} -0.5 & -0.866 & 0 & 0.366 \\ 0.866 & -0.5 & 0 & 1.366 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Numerical Inverse Kinematics



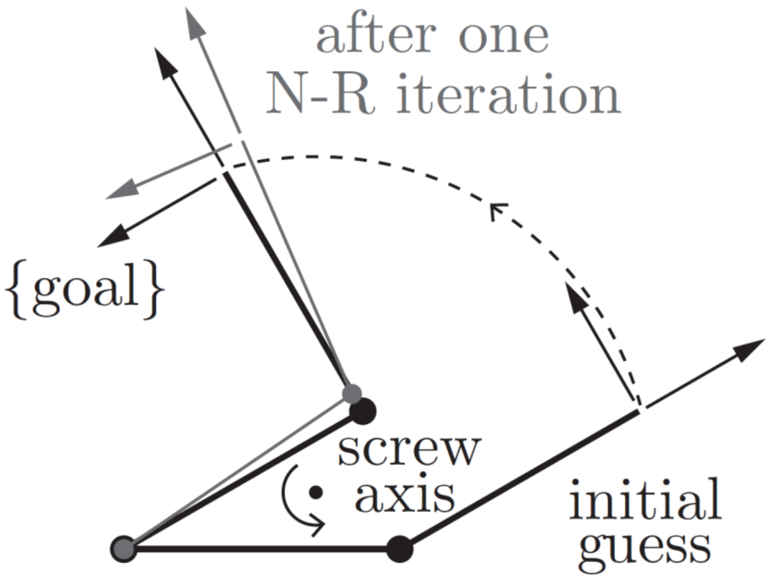

- Forward kinematics

$$M = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathcal{B}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \end{bmatrix} \quad \mathcal{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$T(\theta) = Me^{[\mathcal{B}]_1\theta_1}e^{[\mathcal{B}]_2\theta_2}$$

- Initial guess $\theta^0 = (0, 30^\circ)$

# Numerical Inverse Kinematics




- Compute body twist

$$T_{bd}(\theta^i) = T_{sb}^{-1}(\theta^i)T_{sd} = T_{bs}(\theta^i)T_{sd}$$

$$[\mathcal{V}_b] = \log T_{bd}(\theta^i)$$

- Compute body Jacobian $J_b(\theta) \in \mathbb{R}^{6\times n}$

$$J_{bi}(\theta) = \mathrm{Ad}_{e^{-[\mathcal{B}_n]\theta_n}\cdots e^{-[\mathcal{B}_{i+1}]\theta_{i+1}}}(\mathcal{B}_i)$$

- Update $\theta^{i+1} = \theta^i + J_b^{\dagger}(\theta^i)\mathcal{V}_b$

| $i$ | $(\theta_1, \theta_2)$ | $(x, y)$ | $\mathcal{V}_b = (\omega_{zb}, v_{xb}, v_{yb})$ | $\lVert\omega_b\rVert$ | $\lVert v_b\rVert$ |
|---|---|---|---|---|---|
| 0 | $(0.00, 30.00°)$ | $(1.866, 0.500)$ | $(1.571, 0.498, 1.858)$ | 1.571 | 1.924 |
| 1 | $(34.23°, 79.18°)$ | $(0.429, 1.480)$ | $(0.115, -0.074, 0.108)$ | 0.115 | 0.131 |
| 2 | $(29.98°, 90.22°)$ | $(0.363, 1.364)$ | $(-0.004, 0.000, -0.004)$ | 0.004 | 0.004 |
| 3 | $(30.00°, 90.00°)$ | $(0.366, 1.366)$ | $(0.000, 0.000, 0.000)$ | 0.000 | 0.000 |

$$\theta_d = (30°, 90°)$$

# Summary

- Inverse kinematics
- Newton-Raphson Method
- Numerical Inverse Kinematics Algorithm
- Inverse Velocity Kinematics

# Further Reading

- Chapter 6 and Appendix D in Kevin M. Lynch and Frank C. Park. Modern Robotics: Mechanics, Planning, and Control. 1st Edition, 2017.