

Ch7: Inverse Kinematics

Inverse Kinematics

Inverse Kinematics

The inverse kinematics of a robot refers to the calculation of the joint coordinates θ from the position and orientation (**pose**) of its end-effector frame.

- “Geometric” inverse kinematics:

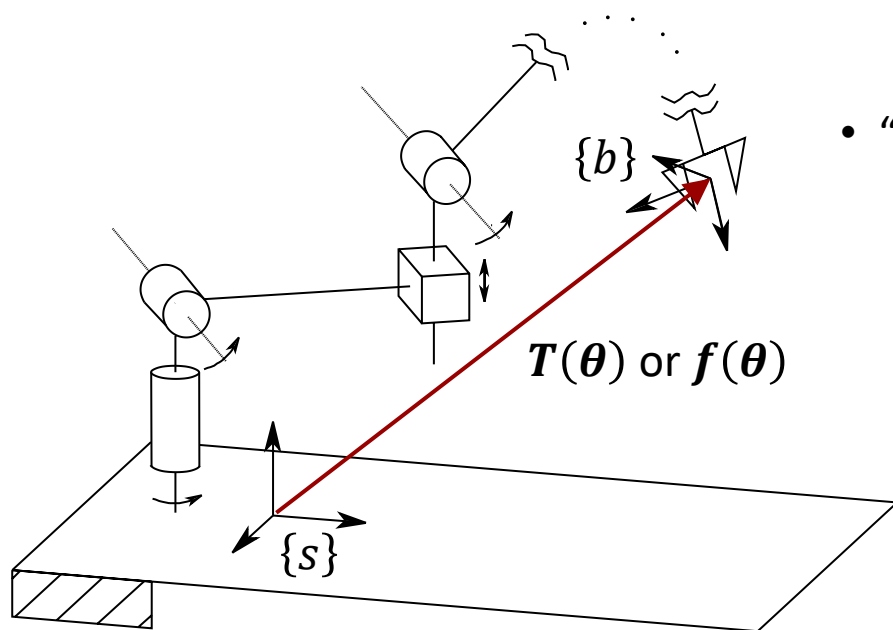
Given $T_{sb} = T(\theta) \in SE(3)$, Find $\theta \in \mathbb{R}^n$

$$T: \mathbb{R}^n \rightarrow SE(3)$$

- “Minimum-Coordinate” inverse kinematics:

Given $x = f(\theta) \in \mathbb{R}^m$, Find $\theta \in \mathbb{R}^n$

$$(m \leq n) \quad f: \mathbb{R}^n \rightarrow \mathbb{R}^m$$



Complexities of Inverse Kinematics

- The equations to solve are in general nonlinear. Thus, it is not always possible to find a closed-form solution.
- Multiple (finite) solutions may exist.
- Infinite solutions may exist (e.g., in the case of a kinematically redundant manipulator).
- There might be no admissible solutions (e.g., when the given EE pose does not belong to the manipulator dexterous workspace.).

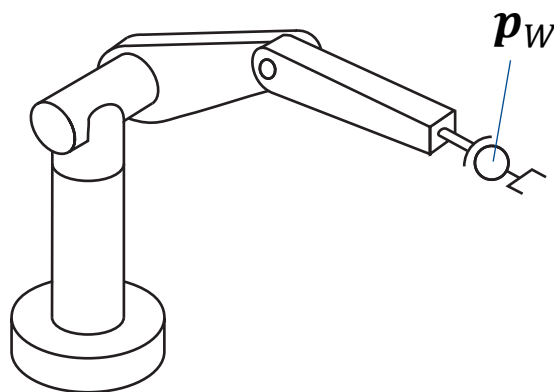
► Solving Inverse Kinematics Problems:

- **Analytic Methods:** Finding closed-form solutions using algebraic intuition or geometric intuition.
- **Iterative Numerical Methods:** When there are no (or it is difficult to find) closed-form solutions.

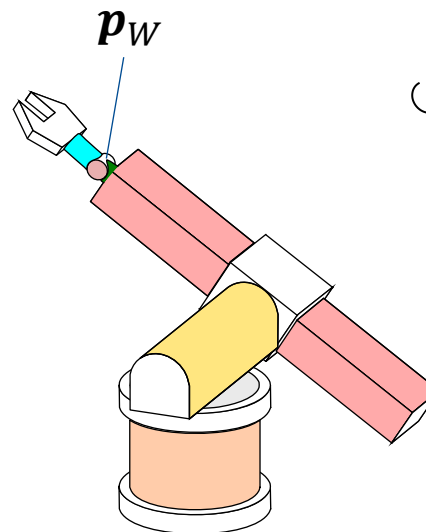
Analytic Inverse Kinematics

Analytic Inverse Kinematics

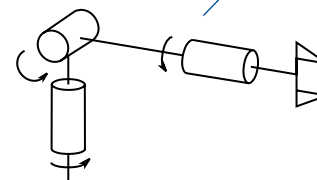
Most of the existing manipulators are typically formed by an **arm** and a **spherical wrist** (i.e., three consecutive revolute joint axes intersect at a common point). Thus, we can decouple the solution for the position (e.g., point p_W at the intersection of the three revolute axes) from that for the orientation.



PUMA Arm (6R)



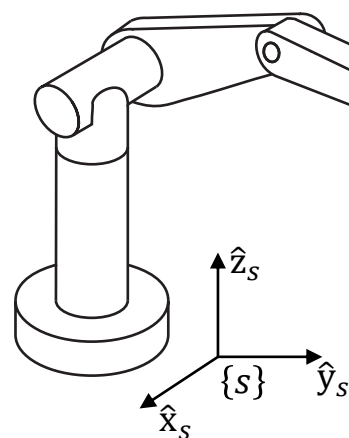
Stanford Arm (RRPRRR)



3R Planar Arm

* Therefore, it is possible to solve the inverse kinematics for the arm separately from the inverse kinematics for the spherical wrist.

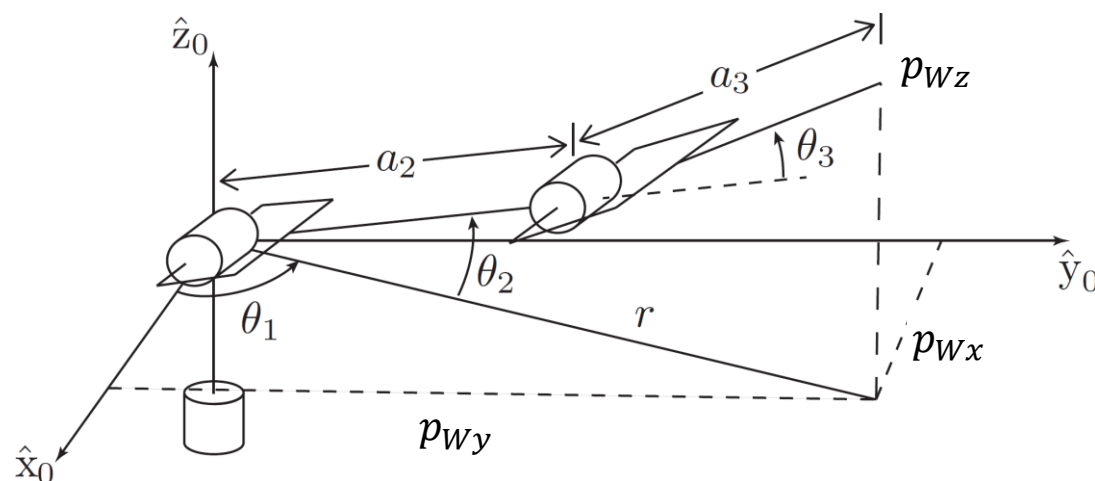
6R PUMA-Type Arms



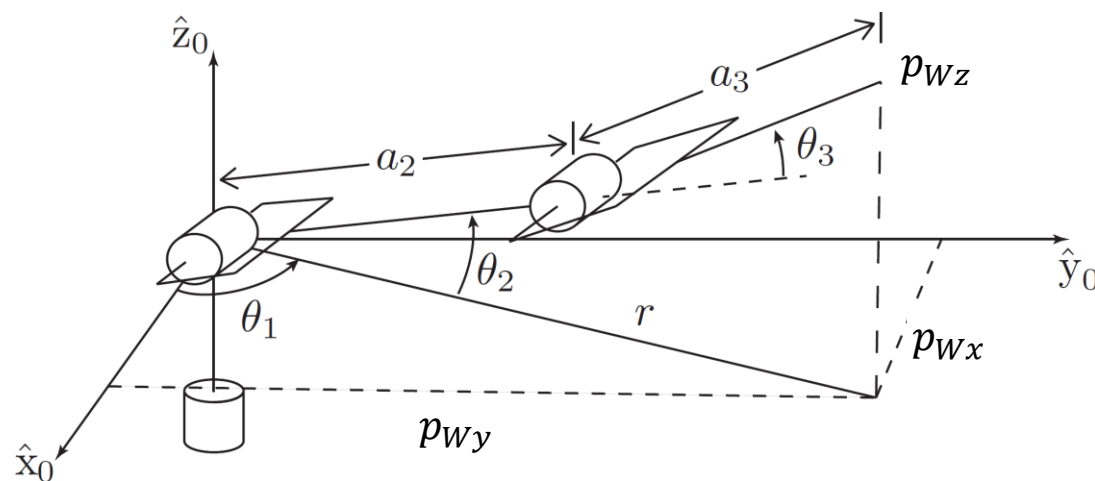
Wrist joints intersect orthogonally at a common point \mathbf{p}_W

$$\mathbf{T}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{R}_{sb} & \mathbf{p}_b \\ \mathbf{0} & 1 \end{bmatrix} = e^{[s_1]\theta_1} e^{[s_2]\theta_2} e^{[s_3]\theta_3} e^{[s_4]\theta_4} e^{[s_5]\theta_5} e^{[s_6]\theta_6} \mathbf{M}$$

By having \mathbf{p}_b , we can find $\mathbf{p}_W = (p_{Wx}, p_{Wy}, p_{Wz})$



6R PUMA-Type Arms (cont.)



$$\begin{aligned} p_{Wx} &= c_1(a_2c_2 + a_3c_{23}) = c_1r \\ p_{Wy} &= s_1(a_2c_2 + a_3c_{23}) = s_1r \\ p_{Wz} &= a_2s_2 + a_3s_{23} \end{aligned}$$

❖ Inverse position problem of finding $(\theta_1, \theta_2, \theta_3)$ using algebraic intuition:

$$p_{Wx}^2 + p_{Wy}^2 + p_{Wz}^2 = a_2^2 + a_3^2 + 2a_2a_3c_3$$

$$c_3 = \frac{p_{Wx}^2 + p_{Wy}^2 + p_{Wz}^2 - a_2^2 - a_3^2}{2a_2a_3}$$

$$s_3 = \pm \sqrt{1 - c_3^2}$$



$$\theta_3 = \text{atan2}(s_3, c_3)$$



$$\begin{aligned} \theta_{3,I} &\in [-\pi, \pi] \\ \theta_{3,II} &= -\theta_{3,I} \end{aligned}$$

6R PUMA-Type Arms (cont.)

$$\begin{aligned}
 p_{Wx}^2 + p_{Wy}^2 &= (a_2 c_2 + a_3 c_{23})^2 \longrightarrow a_2 c_2 + a_3 \mathbf{c}_{23} = \pm \sqrt{p_{Wx}^2 + p_{Wy}^2} = \pm r \\
 p_{Wz} &= a_2 s_2 + a_3 \mathbf{s}_{23} \\
 \mathbf{s}_{23} &= s_2 c_3 + s_3 c_2 \\
 \mathbf{c}_{23} &= c_2 c_3 - s_2 s_3
 \end{aligned}
 \quad \left. \vphantom{\begin{aligned} p_{Wx}^2 + p_{Wy}^2 &= (a_2 c_2 + a_3 c_{23})^2 \\ p_{Wz} &= a_2 s_2 + a_3 \mathbf{s}_{23} \\ \mathbf{s}_{23} &= s_2 c_3 + s_3 c_2 \\ \mathbf{c}_{23} &= c_2 c_3 - s_2 s_3 \end{aligned}} \right\} \Rightarrow$$

$$\begin{aligned}
 c_2 &= \frac{\pm \sqrt{p_{Wx}^2 + p_{Wy}^2} (a_2 + a_3 c_3) + p_{Wz} a_3 s_3}{a_2^2 + a_3^2 + 2a_2 a_3 c_3} \\
 s_2 &= \frac{p_{Wz} (a_2 + a_3 c_3) - \left(\pm \sqrt{p_{Wx}^2 + p_{Wy}^2} a_3 s_3 \right)}{a_2^2 + a_3^2 + 2a_2 a_3 c_3}
 \end{aligned}$$



$$\theta_2 = \text{atan2}(s_2, c_2)$$

For each θ_3 , we have two solutions for θ_2 :

$$\left\{ \begin{array}{l} \theta_{3,I} \\ \theta_{3,II} \end{array} \right\} \left\{ \begin{array}{ll} \theta_{2,I} & (+r) \\ \theta_{2,II} & (-r) \\ \theta_{2,III} & (+r) \\ \theta_{2,IV} & (-r) \end{array} \right.$$

$$\begin{aligned}
 \theta_{3,I} &\rightarrow (\theta_{2,I}, \theta_{2,II}) \\
 \theta_{3,II} &\rightarrow (\theta_{2,III}, \theta_{2,IV})
 \end{aligned}$$

6R PUMA-Type Arms (cont.)

$$p_{Wx} = c_1(a_2c_2 + a_3c_{23})$$

$$p_{Wy} = s_1(a_2c_2 + a_3c_{23})$$

$$a_2c_2 + a_3c_{23} = \pm \sqrt{p_{Wx}^2 + p_{Wy}^2} = \pm r$$



$$p_{Wx} = \pm c_1 \sqrt{p_{Wx}^2 + p_{Wy}^2}$$

$$p_{Wy} = \pm s_1 \sqrt{p_{Wx}^2 + p_{Wy}^2}$$



$$\theta_{1,I} = \text{atan2}(p_{Wy}, p_{Wx})$$

$$\theta_{1,II} = \text{atan2}(-p_{Wy}, -p_{Wx})$$

Thus, in total, there exist four solutions:

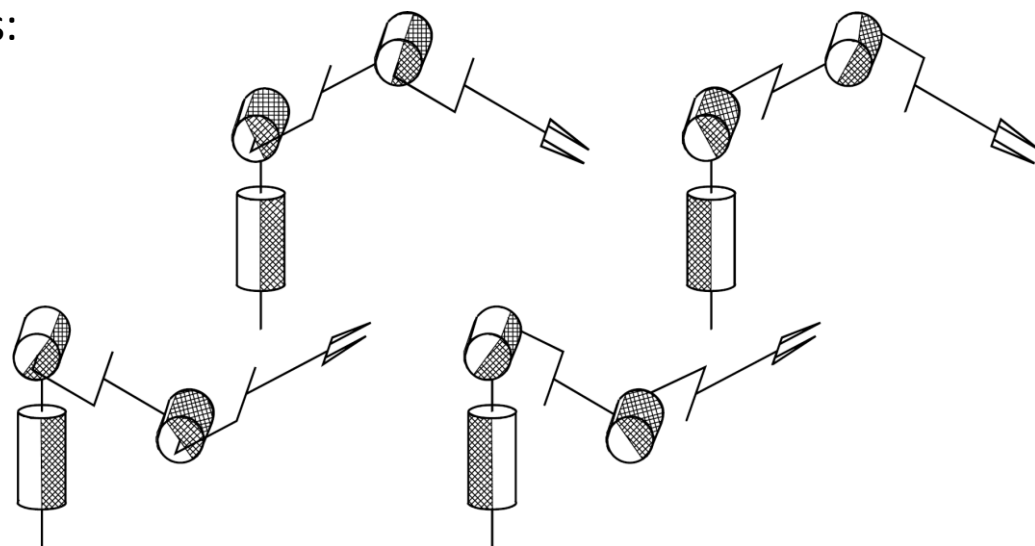
$$\left\{ \begin{array}{l} \theta_{3,I} \left\{ \begin{array}{ll} \theta_{2,I} \quad (+r) & \rightarrow \theta_{1,I} \\ \theta_{2,II} \quad (-r) & \rightarrow \theta_{1,II} \end{array} \right. \\ \theta_{3,II} \left\{ \begin{array}{ll} \theta_{2,III} \quad (+r) & \rightarrow \theta_{1,I} \\ \theta_{2,IV} \quad (-r) & \rightarrow \theta_{1,II} \end{array} \right. \end{array} \right.$$

$$(\theta_{1,I}, \theta_{2,I}, \theta_{3,I})$$

$$(\theta_{1,I}, \theta_{2,III}, \theta_{3,II})$$

$$(\theta_{1,II}, \theta_{2,II}, \theta_{3,I})$$

$$(\theta_{1,II}, \theta_{2,IV}, \theta_{3,II})$$



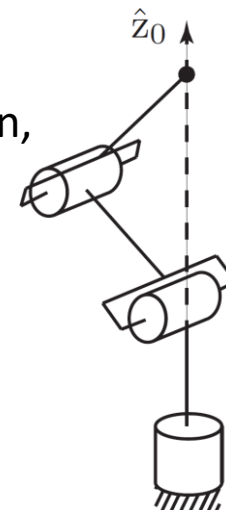
6R PUMA-Type Arms (cont.)

Note: When $p_{Wx} = p_{Wy} = 0$, the arm is in a kinematically singular configuration, and there are infinitely many possible solutions for θ_1 .

❖ Inverse orientation problem of finding $(\theta_4, \theta_5, \theta_6)$:

$$e^{[S_4]\theta_4} e^{[S_5]\theta_5} e^{[S_6]\theta_6} = e^{-[S_3]\theta_3} e^{-[S_2]\theta_2} e^{-[S_1]\theta_1} \mathbf{T}(\boldsymbol{\theta}) \mathbf{M}^{-1} = \mathbf{T}'$$

known



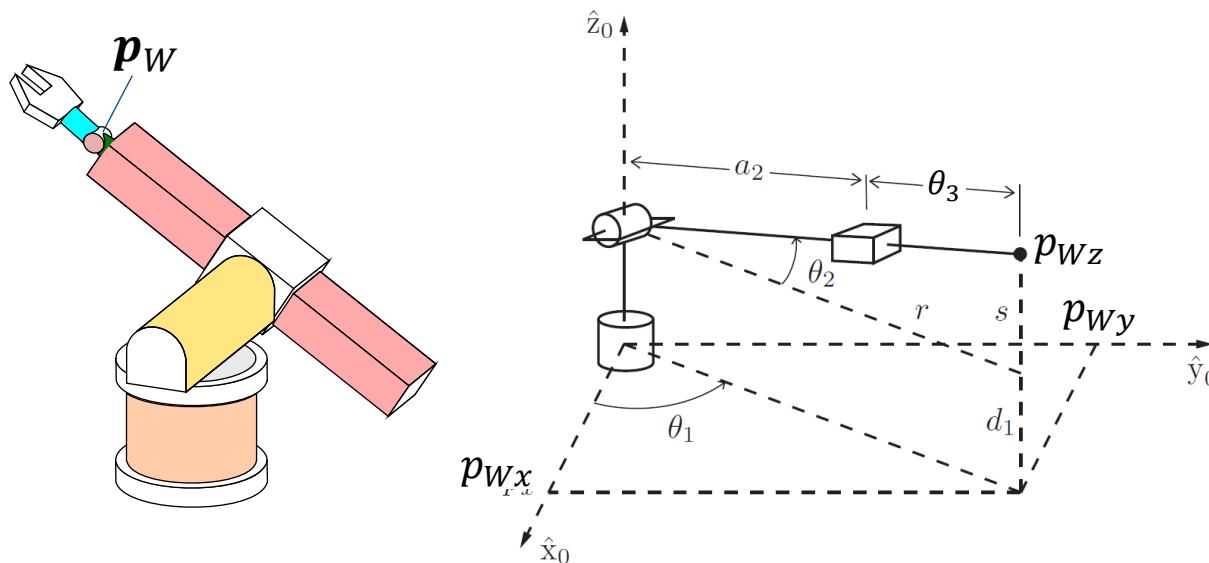
Assume that the joint axes (S_4, S_5, S_6) of the spherical wrist are aligned in the $(\hat{z}_s, \hat{y}_s, \hat{x}_s)$ directions, respectively:

$$\begin{aligned} S_{\omega_4} &= (0, 0, 1) \\ S_{\omega_5} &= (0, 1, 0) \\ S_{\omega_6} &= (1, 0, 0) \end{aligned} \quad \Rightarrow \quad \text{Rot}(\hat{z}, \theta_4) \text{Rot}(\hat{y}, \theta_5) \text{Rot}(\hat{x}, \theta_6) = \mathbf{R}' \quad \Rightarrow \quad \begin{array}{l} \text{This corresponds to} \\ \text{the ZYX Euler angles.} \end{array}$$

$$\mathbf{T}' = (\mathbf{R}', \mathbf{p}')$$

$\Rightarrow (\theta_4, \theta_5, \theta_6)$

Stanford-Type Arms



$$r^2 = p_{Wx}^2 + p_{Wy}^2$$

$$s = p_{Wz} - d_1$$

❖ Inverse position problem of finding $(\theta_1, \theta_2, \theta_3)$ using geometric intuition:

$$\text{If } p_{Wx}, p_{Wy} \neq 0: \begin{cases} \theta_1 = \text{atan2}(p_{Wy}, p_{Wx}) \\ \theta_2 = \text{atan2}(s, r) \end{cases}, \quad \begin{cases} \theta_1 = \pi + \text{atan2}(p_{Wy}, p_{Wx}) \\ \theta_2 = \pi - \text{atan2}(s, r) \end{cases}$$

$$(\theta_3 + a_2)^2 = r^2 + s^2 \quad \longrightarrow \quad \theta_3 = \sqrt{r^2 + s^2} - a_2 = \sqrt{p_{Wx}^2 + p_{Wy}^2 + (p_{Wz} - d_1)^2} - a_2$$

\Rightarrow Thus, there are 2 solutions to the inverse kinematics problem.

❖ Inverse orientation problem of finding $(\theta_4, \theta_5, \theta_6)$ is similar to PUMA.

Numerical Inverse Kinematics

Newton–Raphson Method

Newton–Raphson Method is an iterative method for numerically finding the roots of a nonlinear equation $f(\theta) = 0$ where $f: \mathbb{R} \rightarrow \mathbb{R}$ is differentiable.

If θ^0 is an initial guess for the solution, Taylor expansion of $f(\theta)$ at θ^0 is

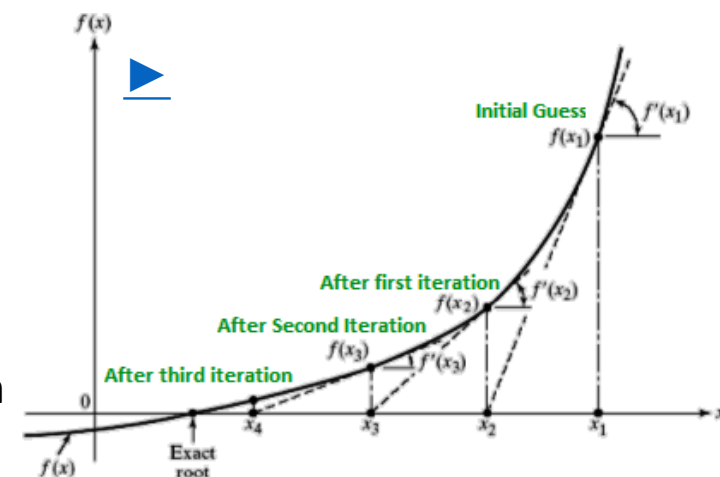
$$f(\theta) = f(\theta^0) + \frac{\partial f}{\partial \theta}(\theta^0)(\theta - \theta^0) + \text{higher-order terms (h.o.t)} \xrightarrow{f(\theta) = 0} \theta = \theta^0 - \left(\frac{\partial f}{\partial \theta}(\theta^0) \right)^{-1} f(\theta^0)$$

Using θ as the new guess for the solution and repeating:

$$\theta^{k+1} = \theta^k - \left(\frac{\partial f}{\partial \theta}(\theta^k) \right)^{-1} f(\theta^k)$$

The iteration is repeated until some stopping criterion is satisfied, e.g., $\frac{|f(\theta^k) - f(\theta^{k+1})|}{|f(\theta^k)|} \leq \epsilon$

ϵ : a given threshold value



Inverse Kinematics Based on Newton–Raphson Method (Minimum-Coordinate IK)

Assume that the EE pose is represented by the minimum number of coordinates, i.e., $\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}) \in \mathbb{R}^m$, $\boldsymbol{\theta} \in \mathbb{R}^n$. Thus, given a desired EE pose \mathbf{x}_d , the goal is to find joint coordinates $\boldsymbol{\theta} = \boldsymbol{\theta}_d$ such that

$$\mathbf{x}_d = \mathbf{f}(\boldsymbol{\theta}_d) \quad (\text{Assumption: } \mathbf{f} \text{ is differentiable})$$

- We use a method similar to the Newton–Raphson method for nonlinear root-finding: Given an initial guess $\boldsymbol{\theta}^0$ which is “close to” a solution $\boldsymbol{\theta}_d$, and using the Taylor expansion:

$$\mathbf{x}_d = \mathbf{f}(\boldsymbol{\theta}) = \mathbf{f}(\boldsymbol{\theta}^0) + \underbrace{\left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}^0}}_{\mathbf{J}(\boldsymbol{\theta}^0) \in \mathbb{R}^{m \times n}} \underbrace{(\boldsymbol{\theta} - \boldsymbol{\theta}^0)}_{\Delta \boldsymbol{\theta}} + \text{h.o.t.}$$

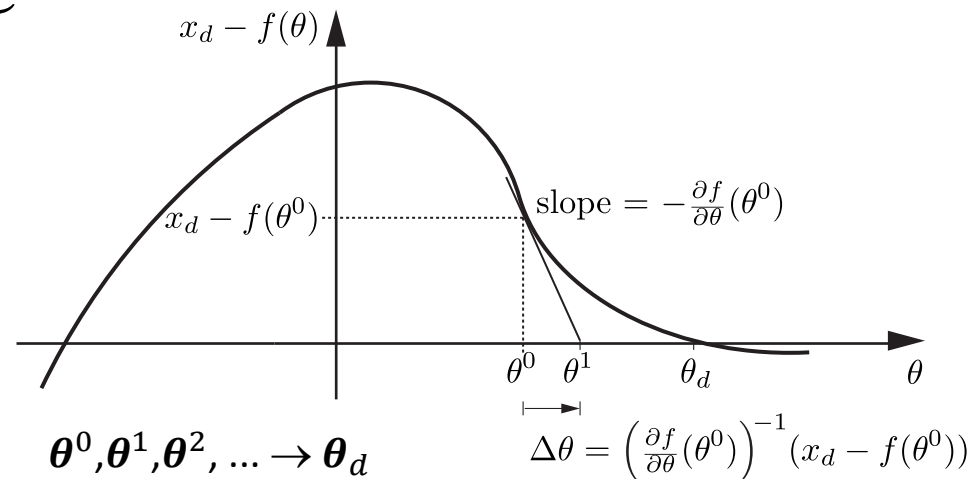
Analytical Jacobian at $\boldsymbol{\theta}^0$

Approximately:

$\xrightarrow{\text{(h.o.t.} = 0\text{)}} \mathbf{J}(\boldsymbol{\theta}^0) \Delta \boldsymbol{\theta} = \mathbf{x}_d - \mathbf{f}(\boldsymbol{\theta}^0)$

* If \mathbf{J} is square ($m = n$) and invertible:

$$\Delta \boldsymbol{\theta} = \mathbf{J}^{-1}(\boldsymbol{\theta}^0)(\mathbf{x}_d - \mathbf{f}(\boldsymbol{\theta}^0))$$



Inverse Kinematics Based on Newton–Raphson Method (Minimum Coordinate IK) (cont.)

* If J is not invertible, either because it is not square or because it is singular,:

$$\Delta\theta = J^{\dagger}(\theta^0)(x_d - f(\theta^0)) \quad J^{\dagger}: \text{Moore–Penrose pseudoinverse}$$

In this case, if J is full rank ($\text{rank}(J) = \min(m, n)$), i.e., the robot is not at a singularity:

- If $n > m$ (the robot has more joints n than end-effector coordinates m):
(Fat Matrix)

$$J^{\dagger} = J^T(JJ^T)^{-1}$$

Note: If there are multiple inverse kinematics solutions, the iterative process tends to converge to the solution that is “closest” to the initial guess θ^0 .

Note: Methods of optimization are needed in situations where an exact solution may not exist and we seek the closest approximate solution; or, conversely, an infinity of inverse kinematics solutions exists (i.e., if the robot is kinematically redundant) and we seek a solution that is optimal with respect to some criterion.

Algorithm for Minimum Coordinate Representation

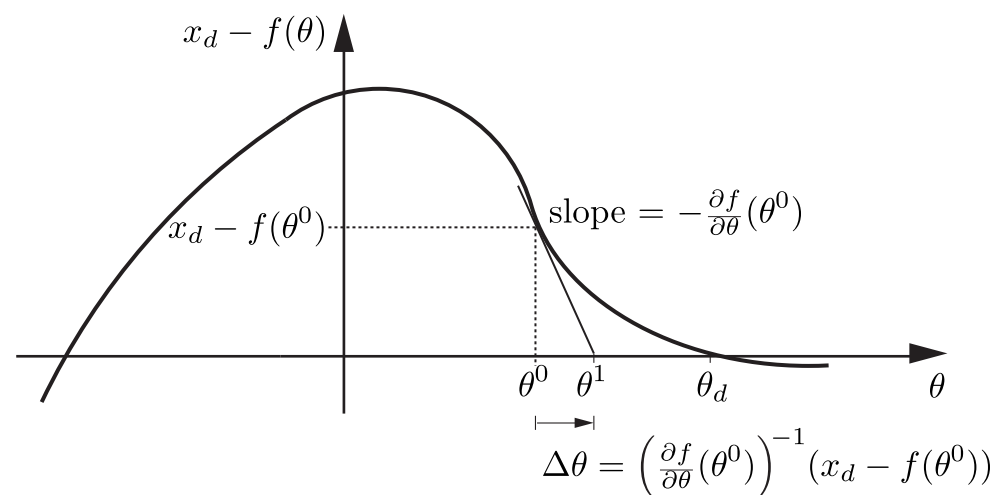
a) Initialization: Given $x_d \in \mathbb{R}^m$ and an initial guess $\theta^0 \in \mathbb{R}^n$, set $i = 0$.

b) Iteration: Set $e = x_d - f(\theta^i)$. While $\|e\| > \epsilon$ for some small $\epsilon \in \mathbb{R}$:

- Set $\theta^{i+1} = \theta^i + J^\dagger(\theta^i)e$.
- Increment i .

Algorithm in MATLAB:

```
max_iterations = 20;  
i = 0;  
Theta = Theta_0;  
e = X_d - FK(Theta);  
while norm(e) > epsilon && i < max_iterations  
    Theta = Theta + pinv(J(Theta)) * e;  
    i = i + 1;  
    e = X_d - FK(Theta);  
end
```



Inverse Kinematics Based on Newton–Raphson Method (Geometric IK)

Assume that the EE pose is represented by a Transformation Matrix, i.e., $T_{sb} = T(\theta) \in SE(3)$, $\theta \in \mathbb{R}^n$. Thus, given a desired EE pose T_{sd} , the goal is to find joint coordinates $\theta = \theta_d$ such that

$$T_{sd} = T(\theta_d)$$

Algorithm for Matrix Transformation Representation:

Algorithm in Body Frame:

a) Initialization: Given $T_{sd} \in SE(3)$ and an initial guess $\theta^0 \in \mathbb{R}^n$, set $i = 0$.

b) Iteration: Set $[\mathbf{v}_b] = \log(T_{bd}(\theta^i)) = \log(T_{sb}^{-1}(\theta^i)T_{sd})$. While $\|\omega_b\| > \epsilon_\omega$ or $\|\mathbf{v}_b\| > \epsilon_v$ for some small $\epsilon_\omega, \epsilon_v \in \mathbb{R}$:

- Set $\theta^{i+1} = \theta^i + J_b^\dagger(\theta^i)\mathbf{v}_b$.

- Increment i .

(\mathbf{v}_b is the twist that takes T_{sb} to T_{sd} in 1s)

(ϵ_ω has the unit of radian and the dimension of ϵ_v is length)

Algorithm in Space Frame:

a) Initialization: Given $T_{sd} \in SE(3)$ and an initial guess $\theta^0 \in \mathbb{R}^n$, set $i = 0$.

b) Iteration: Set $[\mathbf{v}_s] = [\text{Ad}_{T_{sb}}] \log(T_{bd}(\theta^i)) = [\text{Ad}_{T_{sb}}] \log(T_{sb}^{-1}(\theta^i)T_{sd})$. While $\|\omega_s\| > \epsilon_\omega$ or $\|\mathbf{v}_s\| > \epsilon_v$ for some small $\epsilon_\omega, \epsilon_v \in \mathbb{R}$:

- Set $\theta^{i+1} = \theta^i + J_s^\dagger(\theta^i)\mathbf{v}_s$.

- Increment i .

(\mathbf{v}_s is the twist that takes T_{sb} to T_{sd} in 1s)

