Inverse Kinematics
逆向运动学 (IK)

北京大学 前沿计算研究中心

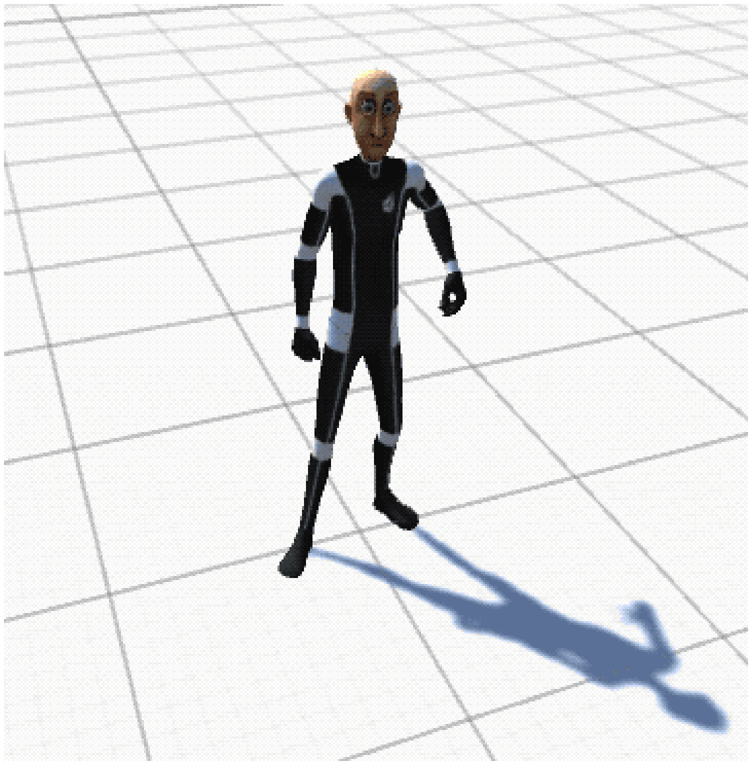
刘利斌

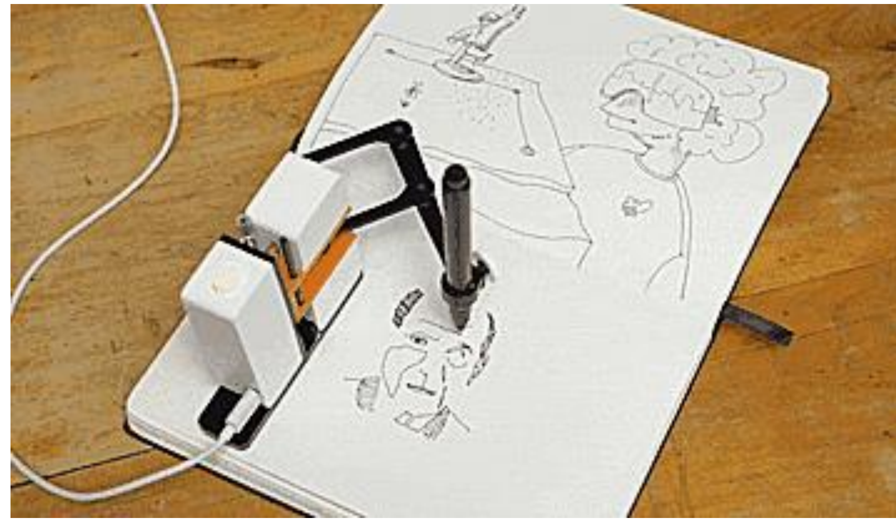
前向运动学 FK vs. 逆向运动学 IK

- 前向运动学
  - 给出关节旋转，计算末端肢体 (End-Effectors) 的位置、朝向等全局信息

- 逆向运动学
  - 给出末端肢体的**目标位置**，计算相应的关节旋转，使得末端肢体到达目标位置

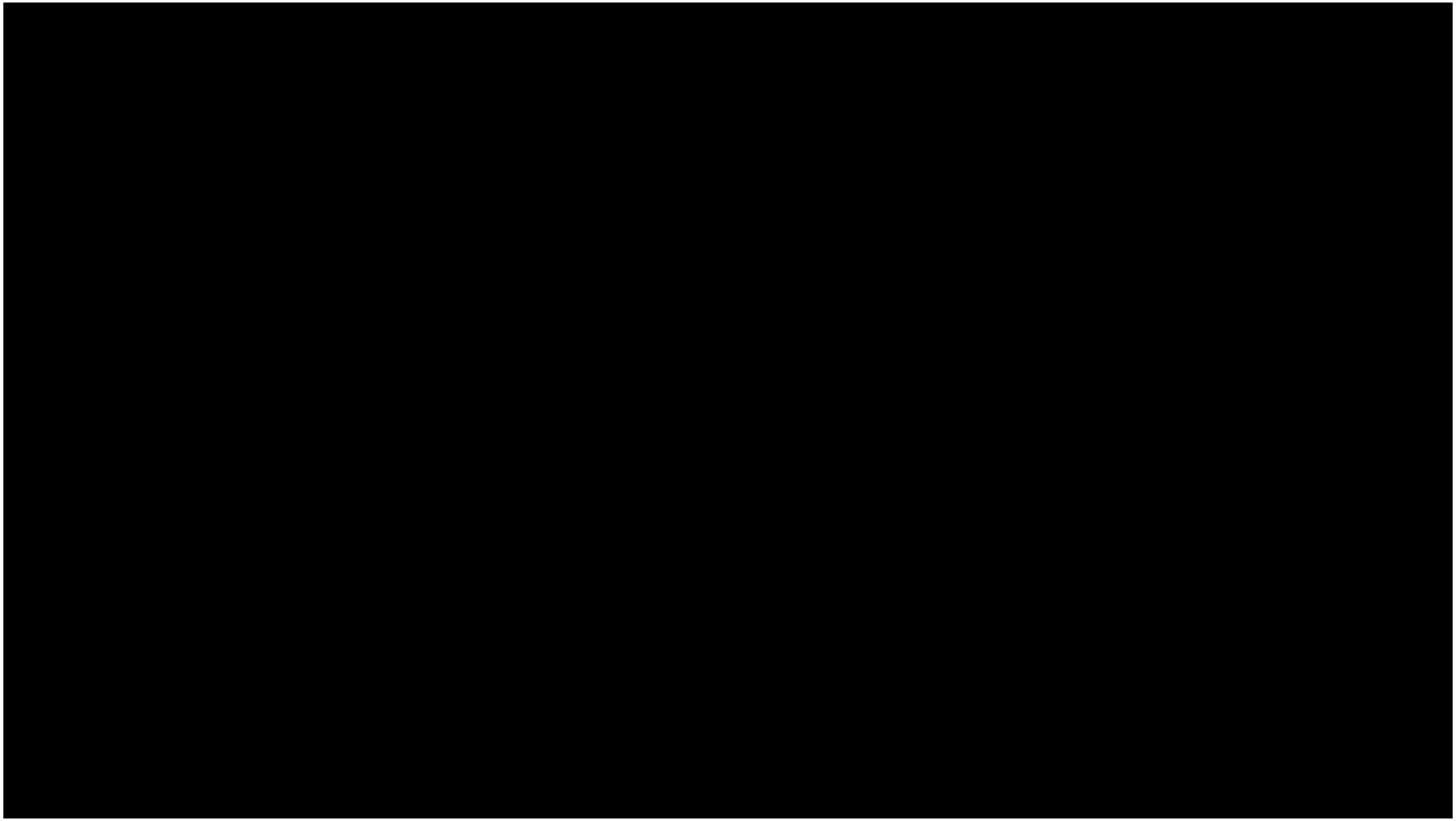
IK应用的例子





- 虚拟角色姿态控制
- 机械臂控制
- etc

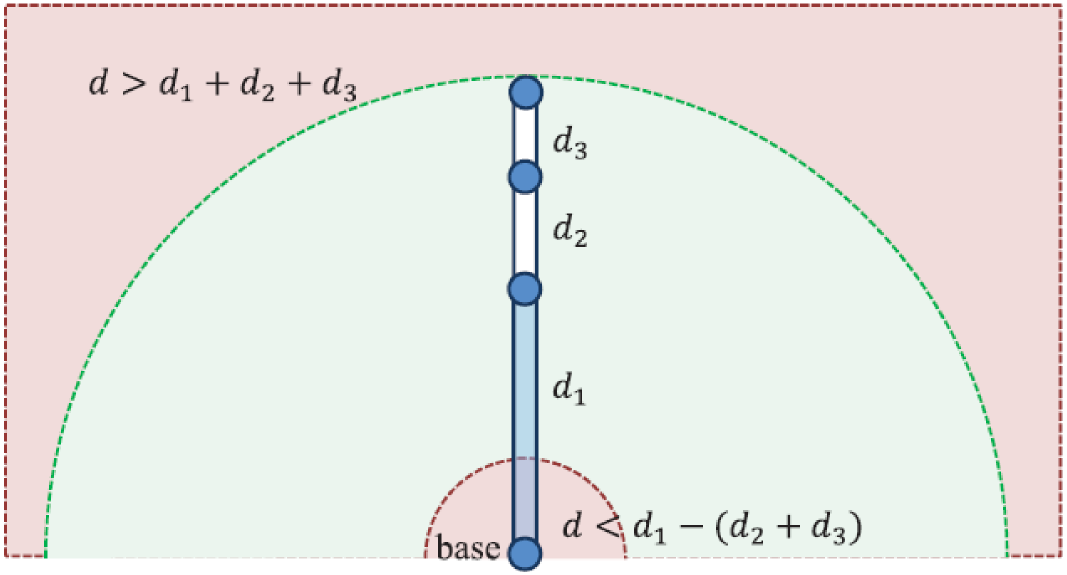
IK应用的例子



Human Body Rig in Blender
https://www.youtube.com/watch?v=MAM7mF2v7dE

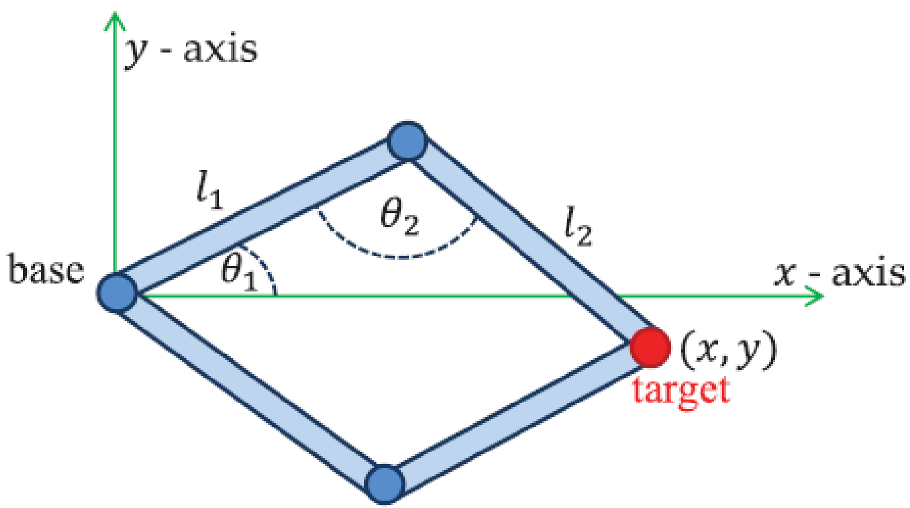
IK问题的可行区域

- Workspace：机械臂的工作空间
- 超出这个范围, 机械臂无法到达
- 此时IK无解



IK问题的多解性

- 可能若干种不同旋转方式
- 可以到达同一个位置
- 最好能够比较平滑，自然



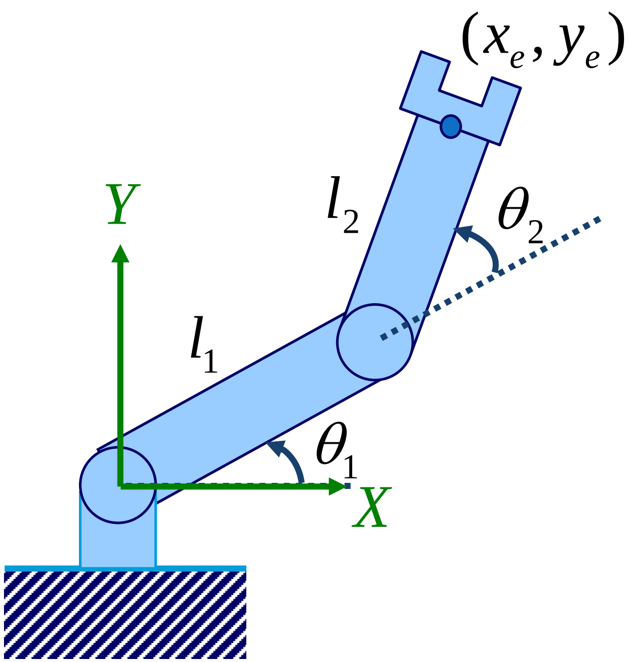
IK问题的解法

- 两关节 (Two-link) IK问题的解析解
- 一般性问题的数值解
  - 启发式方法：
    - CCD， FABRIK
  - 基于Jacobian矩阵的方法
    - Jacobian inverse
    - Jacobian transpose
  - 基于数据的方法

A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. 2018. **Inverse Kinematics Techniques in Computer Graphics: A Survey**. *Computer Graphics Forum* 37, 6 (September 2018), 35–58. https://onlinelibrary.wiley.com/doi/10.1111/cgf.13310

两关节 (Two-link) IK问题

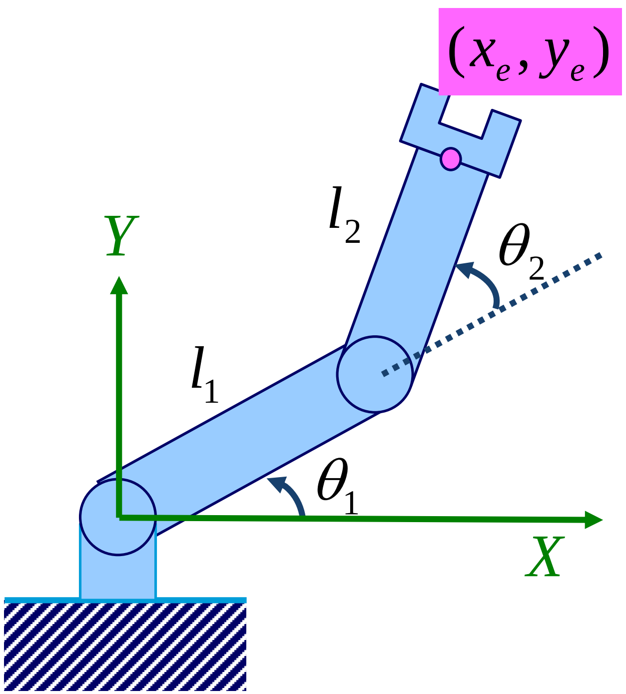
- 只有两个关节的机械臂
  - 前向(FK)问题：给出关节转角，求末端点位置



$$x_e = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y_e = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

两关节 (Two-link) IK问题

- 只有两个关节的机械臂
  - 逆向(FK)问题：给出末端点位置，求关节转角

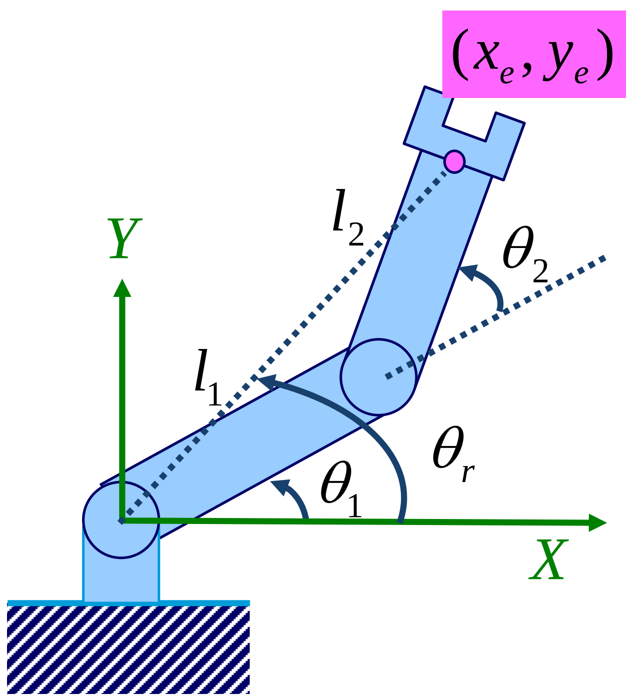


$$\theta_1 = ??$$

$$\theta_2 = ??$$

两关节 (Two-link) IK问题

- 只有两个关节的机械臂
  - 逆向(FK)问题：给出末端点位置，求关节转角



$$\cos(\theta_r) = \frac{x_e}{\sqrt{x_e^2 + y_e^2}}$$

$$\theta_r = \cos^{-1}\left(\frac{x_e}{\sqrt{x_e^2 + y_e^2}}\right)$$

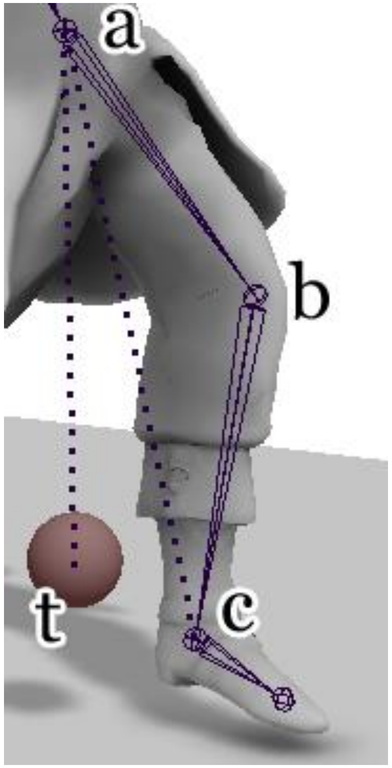
$$\cos(\theta_r - \theta_1) = \frac{l_1^2 + x_e^2 + y_e^2 - l_2^2}{2l_1\sqrt{x_e^2 + y_e^2}}$$

$$\theta_1 = \theta_r - \cos^{-1}\left(\frac{l_1^2 + x_e^2 + y_e^2 - l_2^2}{2l_1\sqrt{x_e^2 + y_e^2}}\right)$$

$$\cos(\pi - \theta_2) = \frac{l_1^2 + l_2^2 - x_e^2 - y_e^2}{2l_1 l_2}$$

$$\theta_2 = \pi - \cos^{-1}\left(\frac{l_1^2 + l_2^2 - x_e^2 - y_e^2}{2l_1 l_2}\right)$$

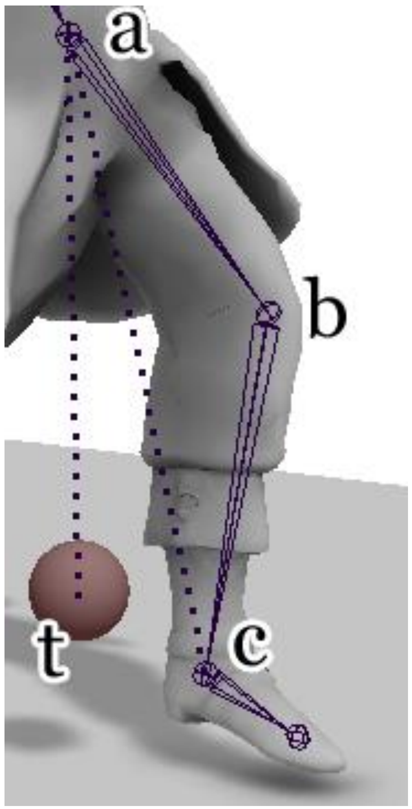
两关节 (Two-link) IK问题 – 3D情况

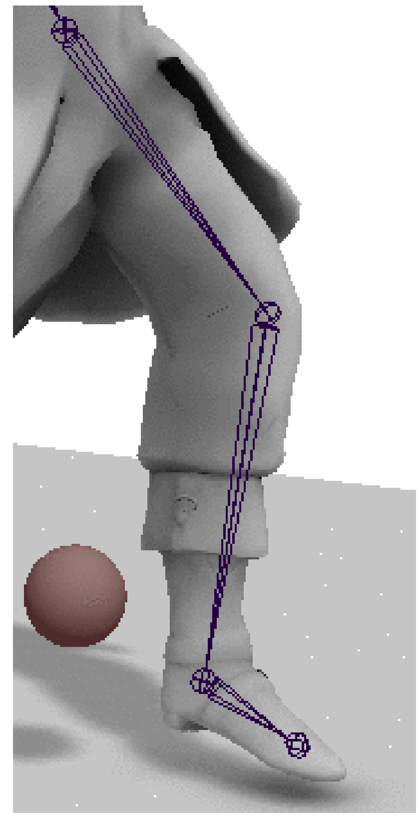


http://theorangeduck.com/page/simple-two-joint

- 问题描述
- 我们希望旋转关节a, b,
- 使c能够到达目标位置t
- 两个关节的情况有解析解

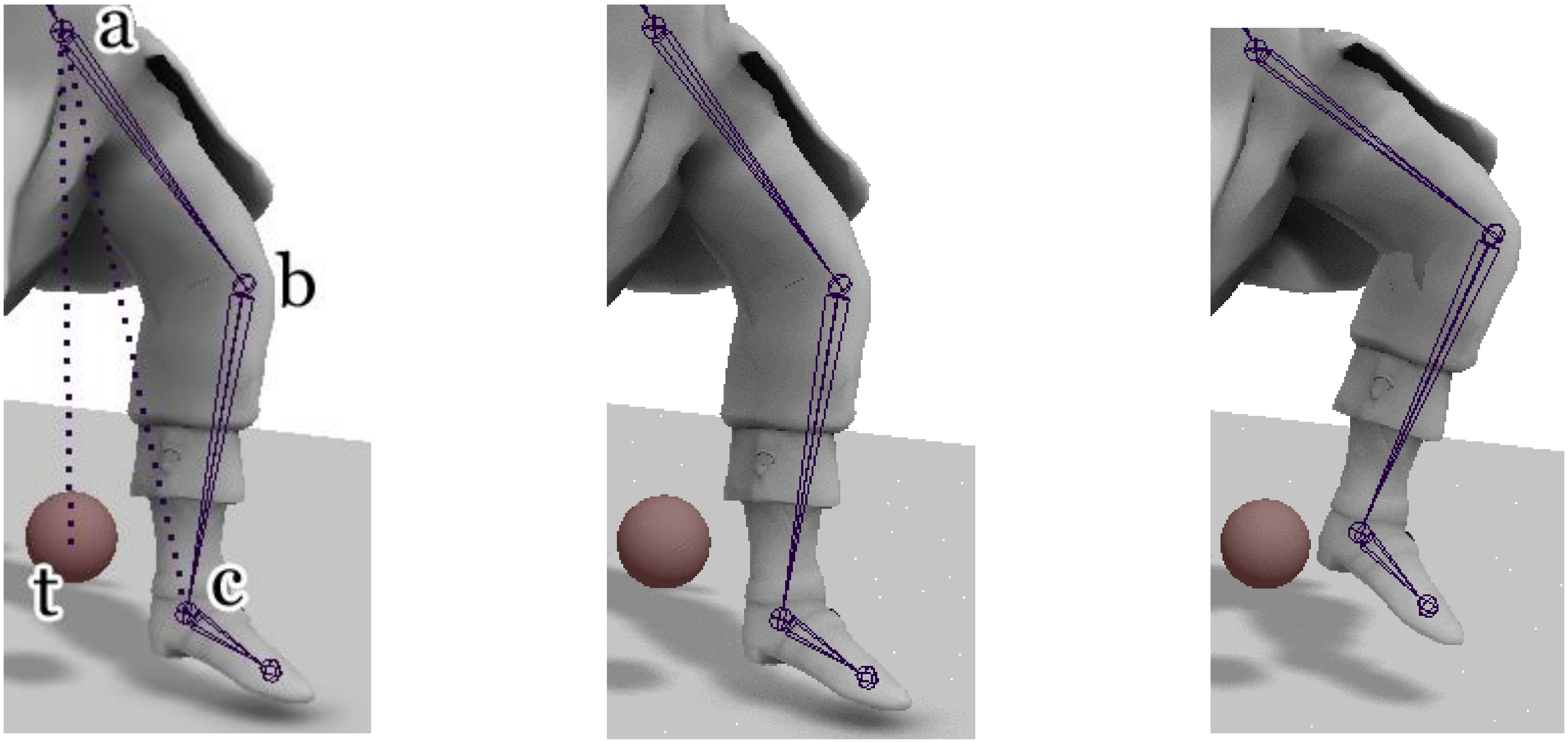
两关节 (Two-link) IK问题 – 3D情况





1. 旋转膝盖, 使得$ac = at$

两关节 (Two-link) IK问题 – 3D情况



旋转髋关节, 使c与目标t的位置重合

一般性问题的数值解

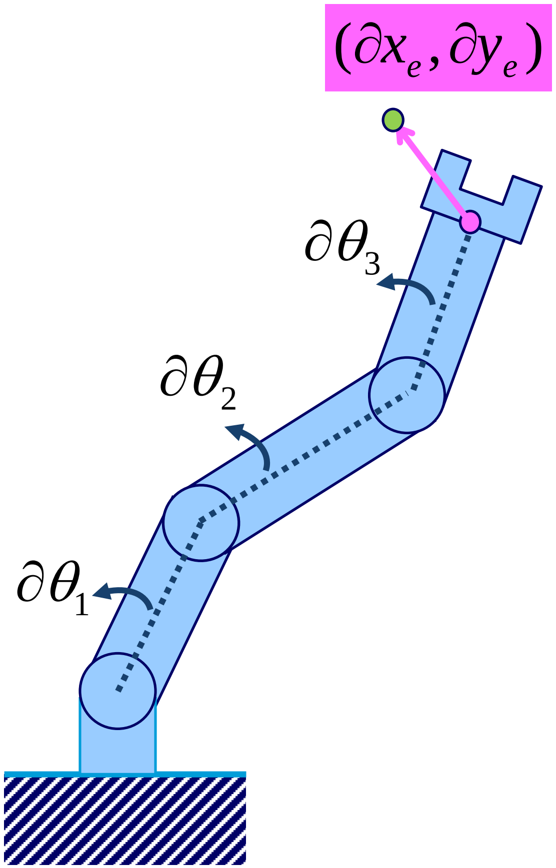
- 关节数 > 2 的IK问题一般没有解析解
- 数值求解 IK 问题
  - 基本思想：迭代

从初始状态出发

Loop：

计算/猜测关节角的更新

更新关节角，计算误差



启发式的IK方法

Heuristic IK Algorithms

一般性问题的数值解

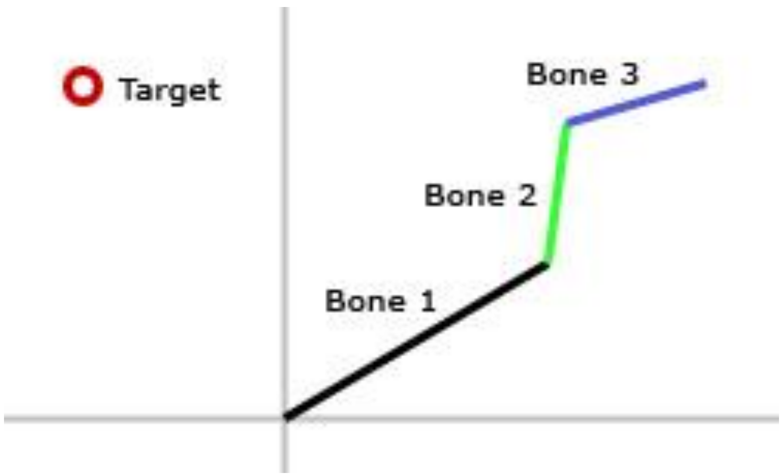
- 启发式方法：
  - CCD
  - FABRIK
- 基于Jacobian矩阵的方法
  - Jacobian inverse
  - Jacobian transpose
- 基于数据的方法

A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. 2018. **Inverse Kinematics Techniques in Computer Graphics: A Survey**. *Computer Graphics Forum* 37, 6 (September 2018), 35–58. https://onlinelibrary.wiley.com/doi/10.1111/cgf.13310

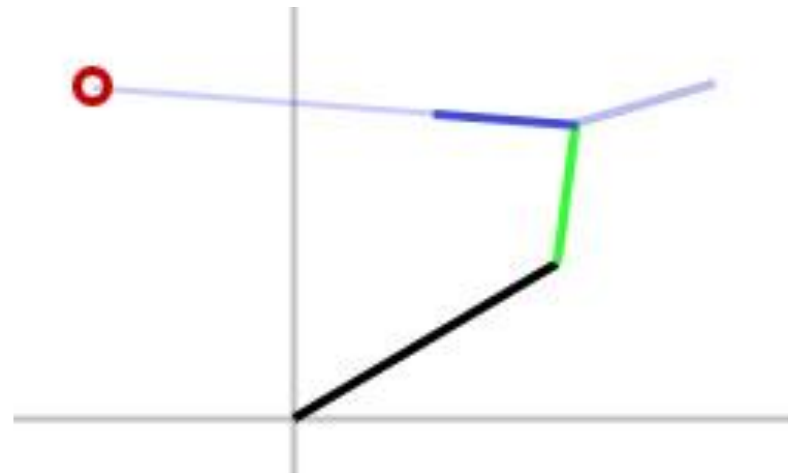
循环坐标下降法
Cyclic Coordinate Descent (CCD)

- 非常常用的 IK 算法
- 基本思想：
  - 从链条末端开始，依次让每个关节与末端的连线指向目标
  - 循环若干次，直到链条末端与目标重合

循环坐标下降法
Cyclic Coordinate Descent (CCD)

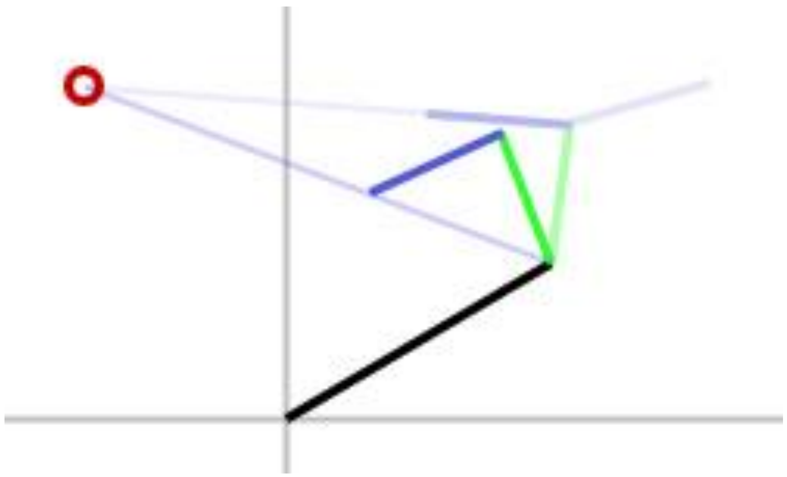


初始状态

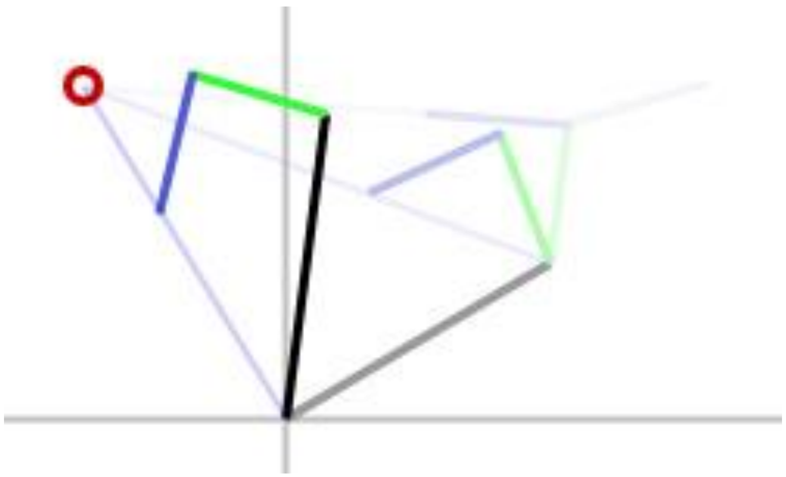


让蓝色关节与末端的连线指向目标

循环坐标下降法
Cyclic Coordinate Descent (CCD)

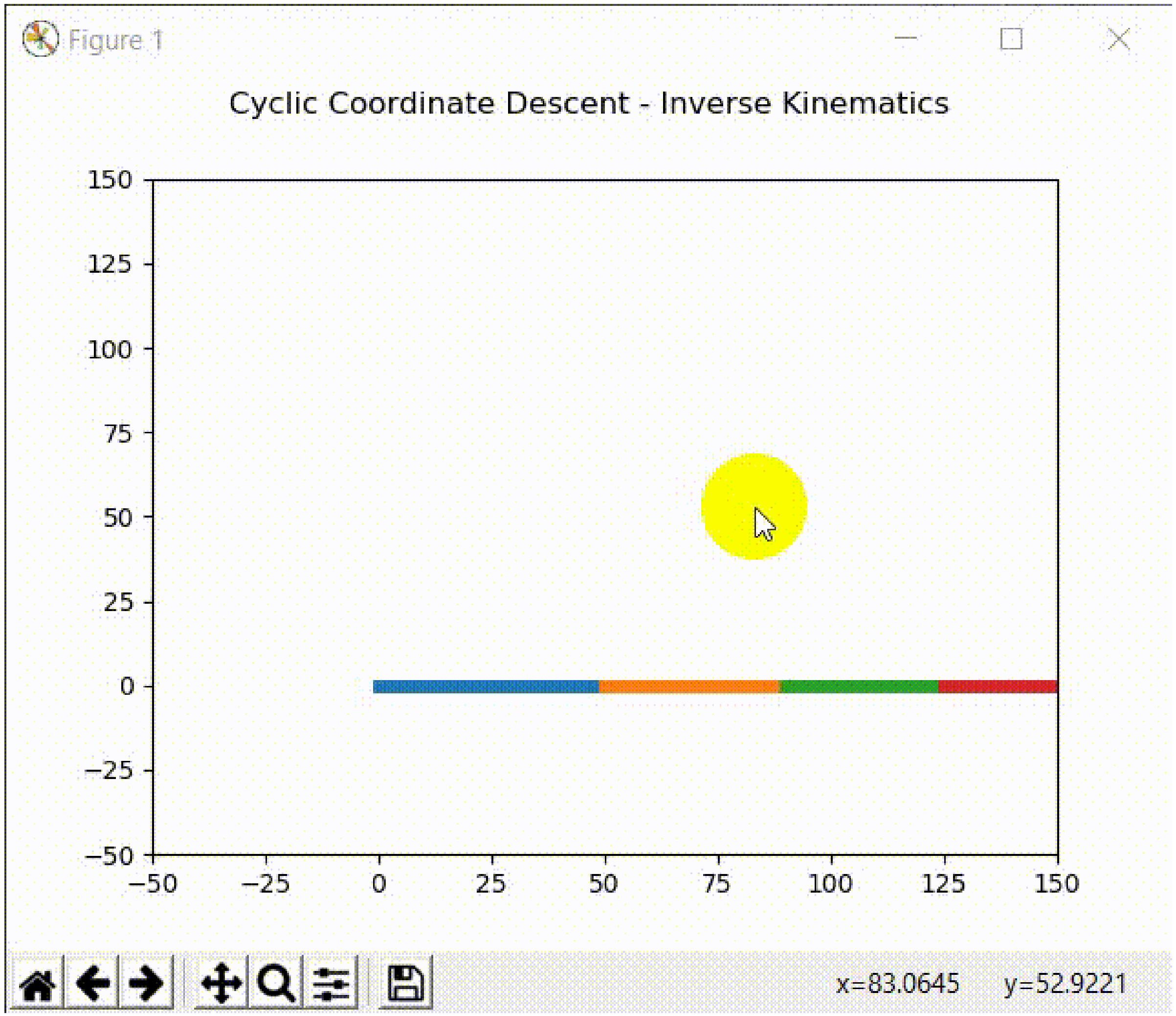


让绿色关节与末端的连线指向目标



让黑色关节与末端的连线指向目标

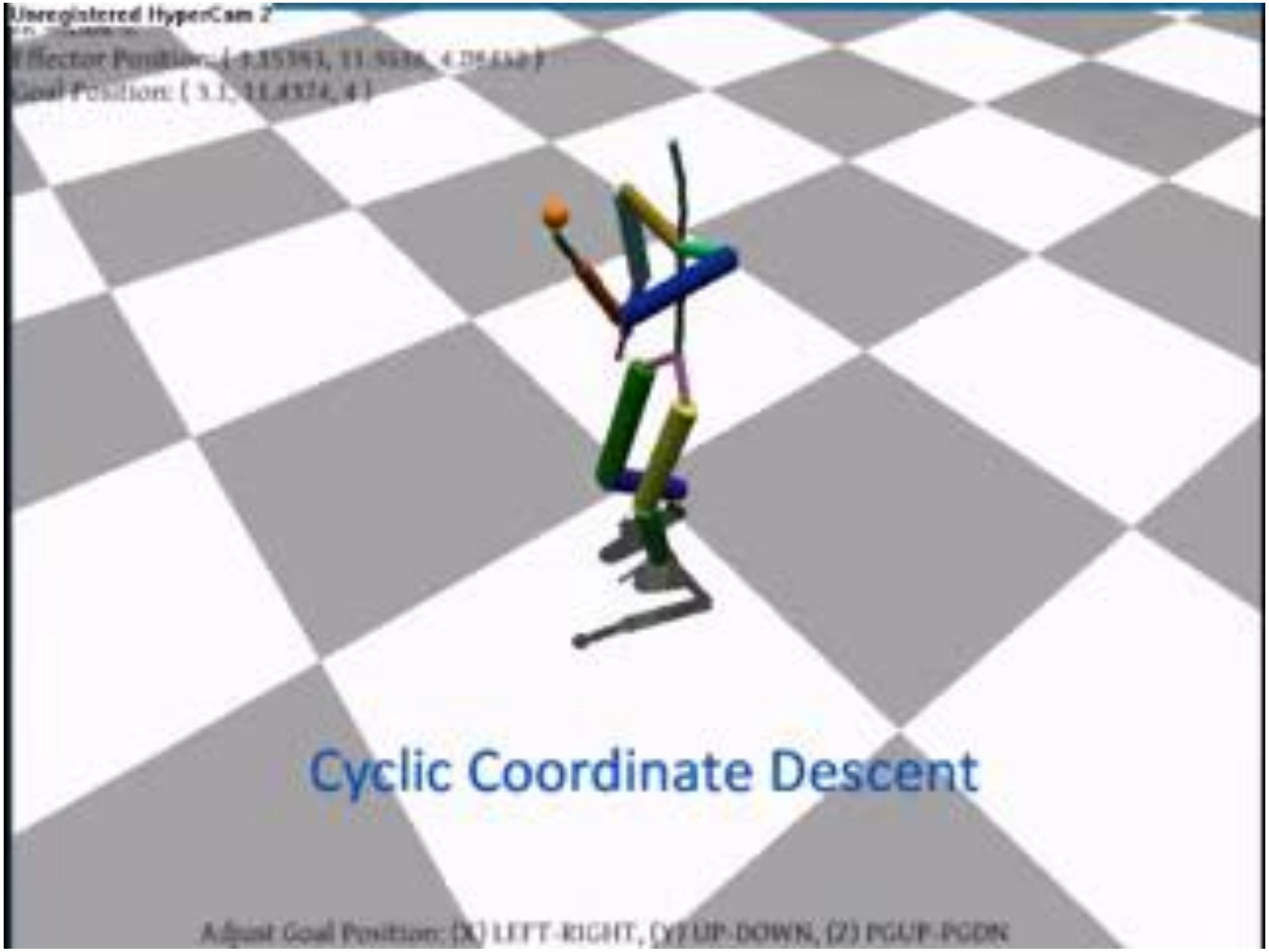
循环坐标下降法
Cyclic Coordinate Descent (CCD)



循环坐标下降法
Cyclic Coordinate Descent (CCD)

- 优点
  - 简单
  - 快
- 缺点
  - 末端关节运动较大
  - 可能收敛较慢甚至无法收敛
  - 跟踪连续目标是可能动作不稳定

循环坐标下降法
Cyclic Coordinate Descent (CCD)

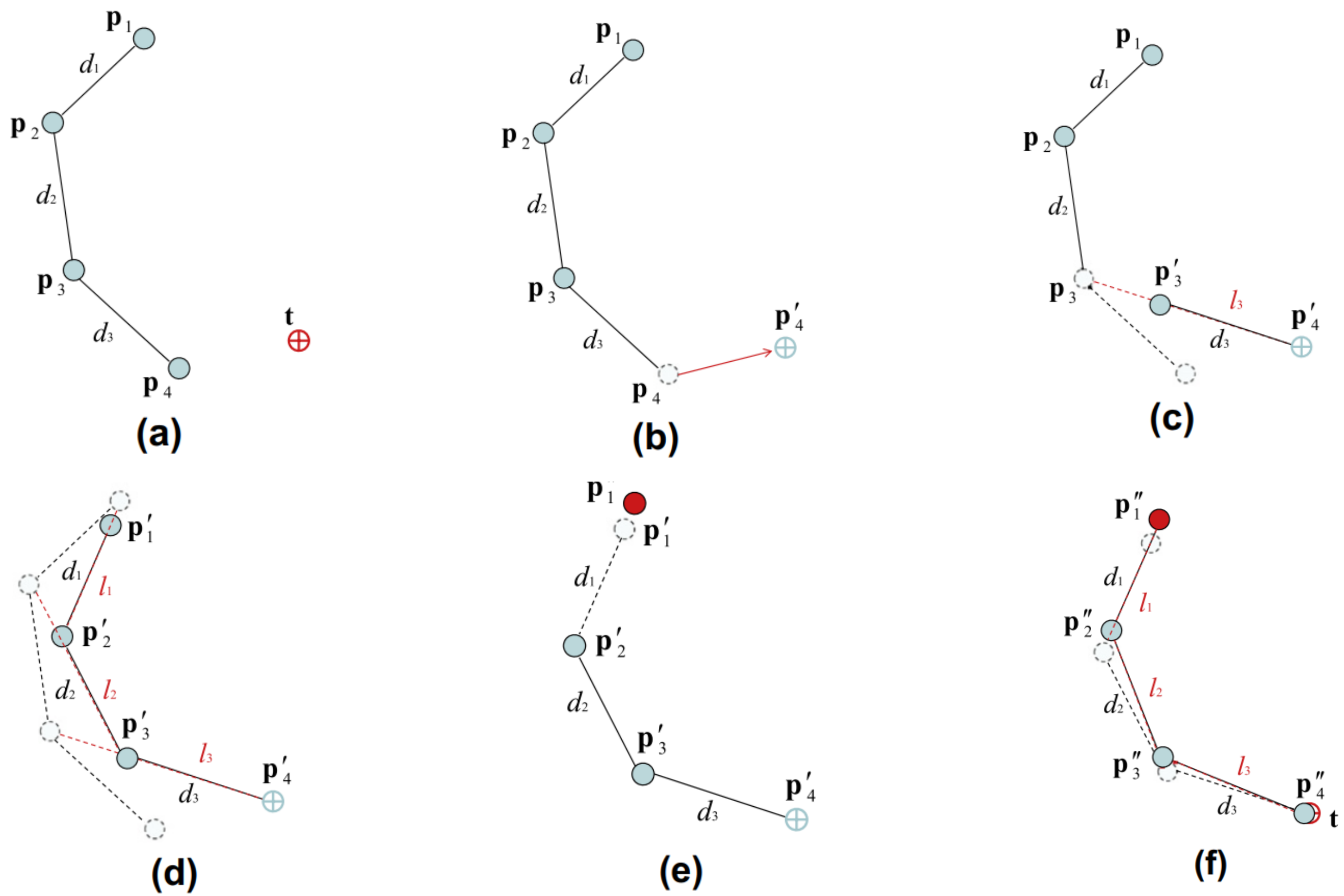


FABRIK - Forward and Backward Reaching IK

- 结合前向与后向过程的方法
- 基本思想：
  - 后向过程
    - 从末端关节开始，依次将每个关节放置在合适的位置
  - 前向过程
    - 从起点关节开始，依次修正关节的错位

Andreas Aristidou and Joan Lasenby. 2011. **FABRIK: A fast, iterative solver for the Inverse Kinematics problem**. *Graphical Models* 73, 5 (September 2011), 243–260. https://www.sciencedirect.com/science/article/pii/S1524070311000178

FABRIK - Forward and Backward Reaching IK



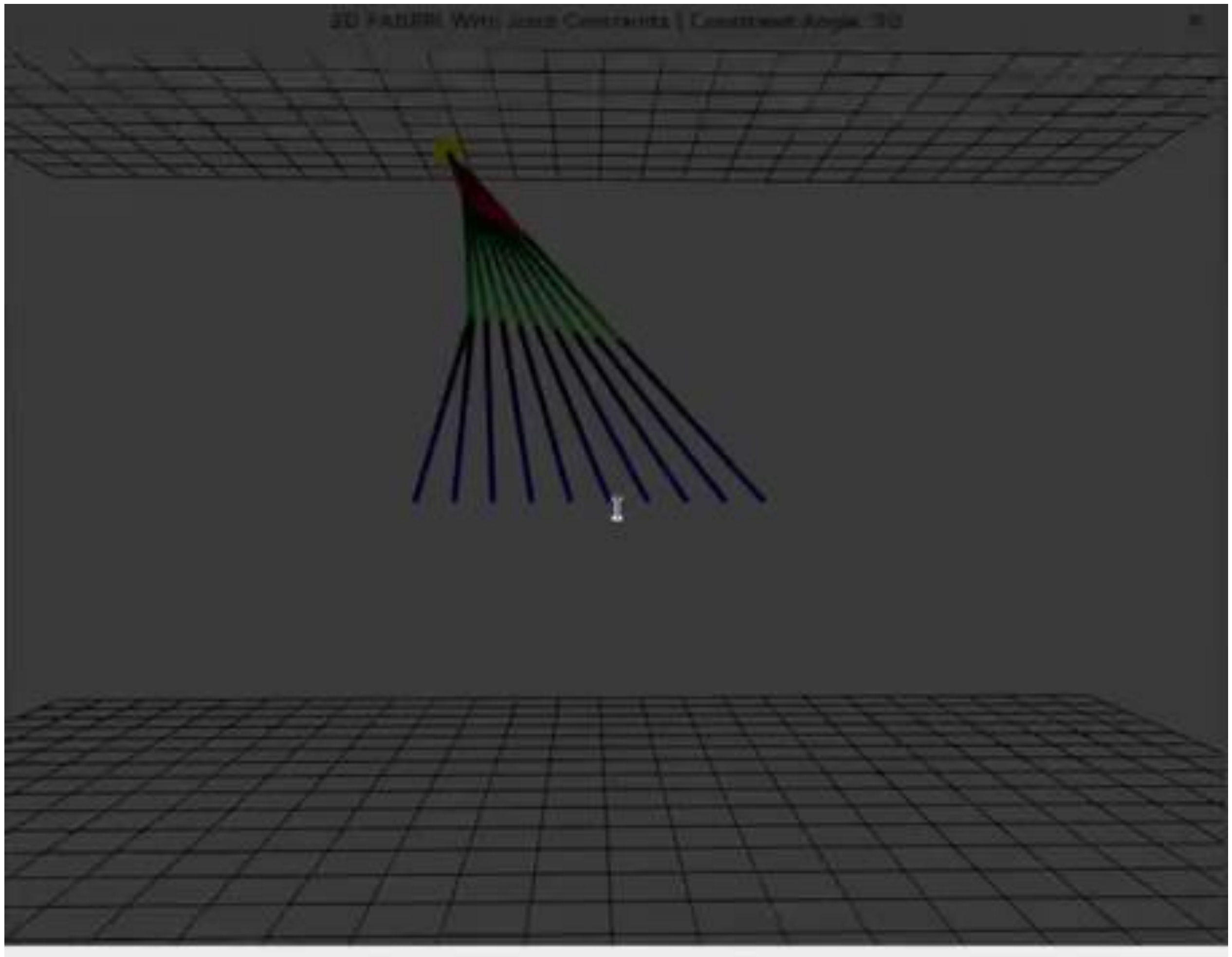
Andreas Aristidou and Joan Lasenby. 2011. **FABRIK: A fast, iterative solver for the Inverse Kinematics problem**. *Graphical Models* 73, 5 (September 2011), 243–260

FABRIK - Forward and Backward Reaching IK

- 优点
  - 简单快速
  - 有解时保证收敛
  - 可以处理关节约束
    - 但是不保证收敛性

- 缺点
  - 基于关节位置的方法
    需要额外步骤计算关节角度
  - 不保证时间连续

FABRIK - Forward and Backward Reaching IK



FABRIK Inverse Kinematics Implementation in 3D with Joint Constraints
https://www.youtube.com/watch?v=dWb-ke_-JXI

基于Jacobian的方法

Jacobian-based IK Methods

一般IK问题的数值解

- 启发式方法：
  - CCD
  - FABRIK
- 基于Jacobian矩阵的方法
  - Jacobian inverse
  - Jacobian transpose
- 基于数据的方法

A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. 2018. **Inverse Kinematics Techniques in Computer Graphics: A Survey**. *Computer Graphics Forum* 37, 6 (September 2018), 35–58. https://onlinelibrary.wiley.com/doi/10.1111/cgf.13310

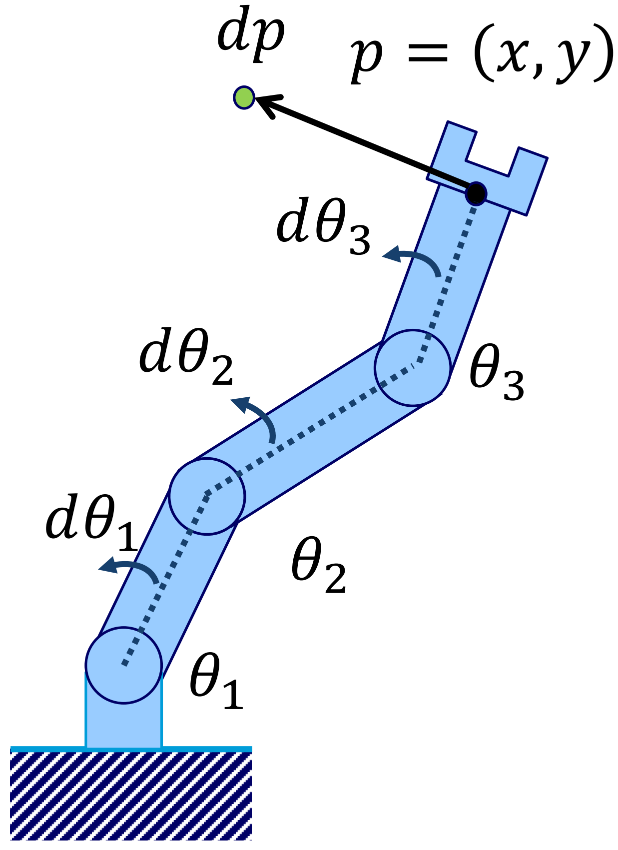
Jacobian矩阵

- 末端肢体的位置可以写成转角的函数

$$p = f(\theta_0, \theta_1, \theta_2)$$

- 两边求导，可以得到

$$\boldsymbol{dp} = \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \frac{\partial f_x}{\partial \theta_1} & \frac{\partial f_x}{\partial \theta_2} & \frac{\partial f_x}{\partial \theta_3} \\ \frac{\partial f_y}{\partial \theta_1} & \frac{\partial f_y}{\partial \theta_2} & \frac{\partial f_y}{\partial \theta_3} \end{pmatrix} \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \end{pmatrix} = \mathrm{J}\boldsymbol{d\theta}$$



Jacobian矩阵

- 更一般的情况：

$$\boldsymbol{f} = (f_1, f_2, \dots, f_m)^T \qquad \boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_n)^T$$

$$J = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} = \begin{array}{cc} \text{自由度 } n & \\ \begin{pmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \cdots & \frac{\partial f_1}{\partial \theta_n} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \cdots & \frac{\partial f_2}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial \theta_1} & \frac{\partial f_m}{\partial \theta_2} & \cdots & \frac{\partial f_m}{\partial \theta_n} \end{pmatrix} & \text{约束数量 } m \end{array}$$

如何计算Jacobian矩阵？

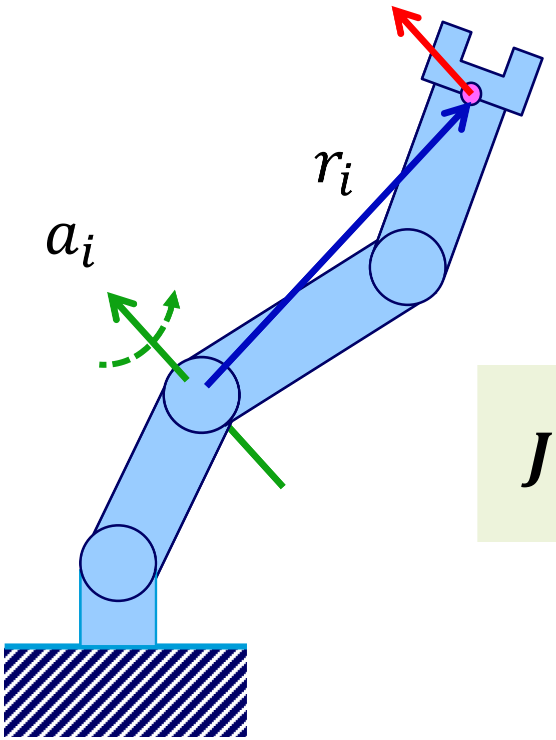
- 解析求导
  - 利用pytorch/tensorflow等自动求导工具

- 有限差分

$$\boldsymbol{J}_{ij}=\frac{\partial f_i}{\partial \theta_j}=\frac{f_i(\theta_1,\theta_2\cdots,\theta_j+\varepsilon,\cdots)-f_i(\theta_1,\theta_2\cdots,\theta_j,\cdots)}{\varepsilon}$$

- 几何方法 -- 叉乘

用几何方法计算Jacobian矩阵

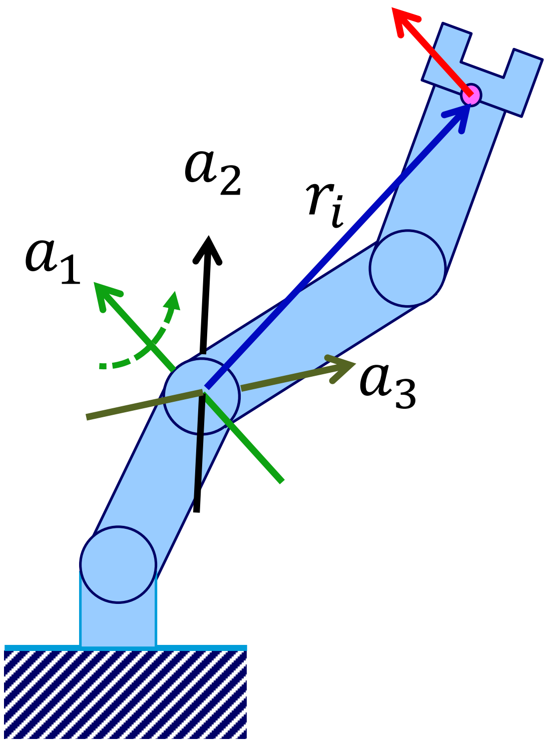


$$J = (\boldsymbol{a}_1 \times \boldsymbol{r}_1 \quad \boldsymbol{a}_2 \times \boldsymbol{r}_2 \quad \cdots \quad \boldsymbol{a}_n \times \boldsymbol{r}_n)$$

用几何方法计算Jacobian矩阵

如何处理多于一个自由度的关节?

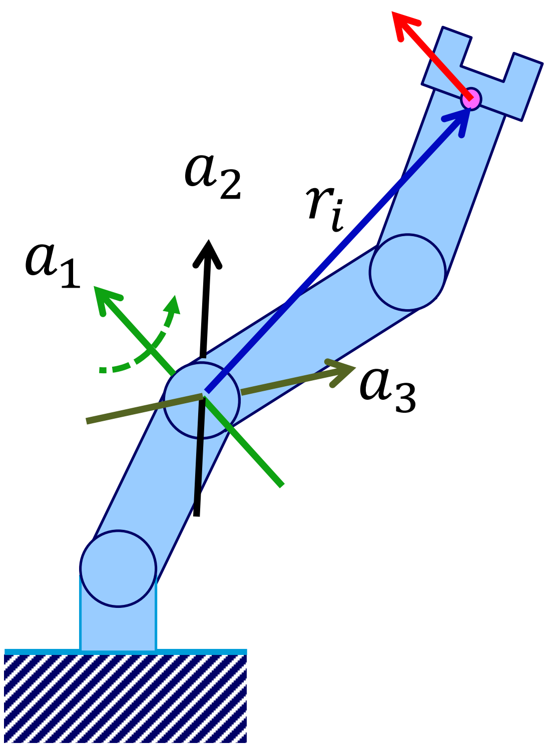
→ 可看作多个同位置的单自由度关节



用几何方法计算Jacobian矩阵

如何处理多于一个自由度的关节?

→ 可看作多个同位置的单自由度关节



用欧拉角表示旋转

$$R = R_x(\theta)R_y(\varphi)R_z(\psi)$$

$$\boldsymbol{a}_1 = \boldsymbol{e}_x$$

$$\boldsymbol{a}_2 = \boldsymbol{e}_y$$

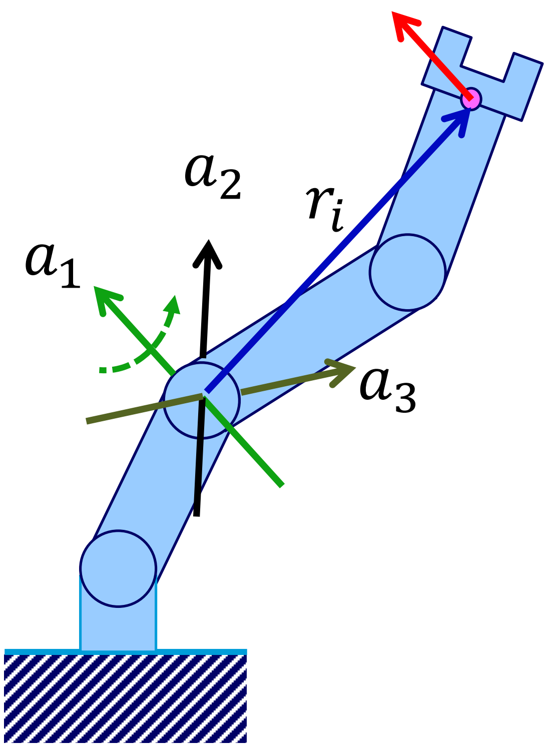
$$\boldsymbol{a}_3 = \boldsymbol{e}_z$$

?

用几何方法计算Jacobian矩阵

如何处理多于一个自由度的关节?

→ 可看作多个同位置的单自由度关节



用欧拉角表示旋转

$$R = R_x(\theta)R_y(\varphi)R_z(\psi)$$

$$\boldsymbol{a}_1 = \boldsymbol{e}_x$$

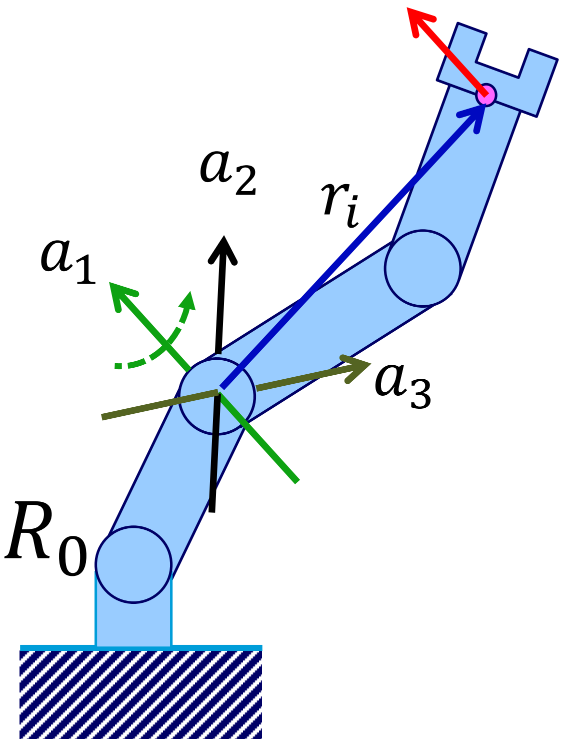
$$\boldsymbol{a}_2 = R_x\boldsymbol{e}_y$$

$$\boldsymbol{a}_3 = R_xR_y\boldsymbol{e}_z$$

用几何方法计算Jacobian矩阵

如何处理多于一个自由度的关节?

→ 可看作多个同位置的单自由度关节



用欧拉角表示旋转

$$R = R_x(\theta)R_y(\varphi)R_z(\psi)$$

$$\boldsymbol{a}_1 = R_0\boldsymbol{e}_x$$

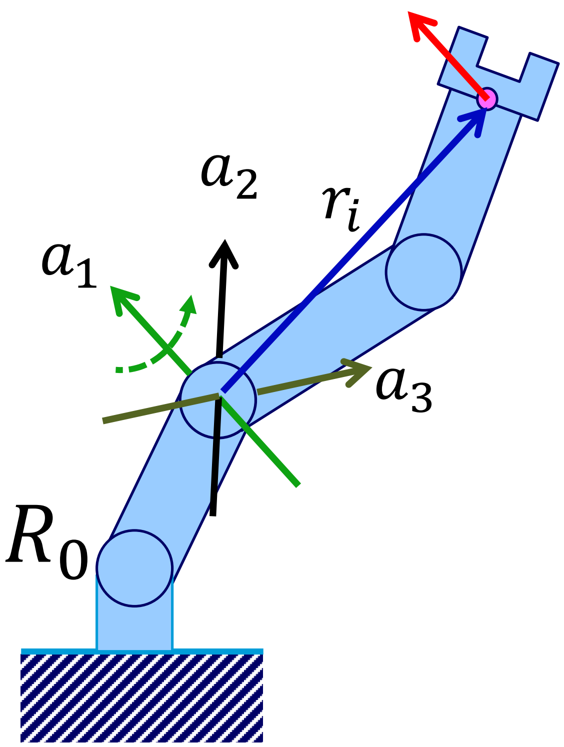
$$\boldsymbol{a}_2 = R_0R_x\boldsymbol{e}_y$$

$$\boldsymbol{a}_3 = R_0R_xR_y\boldsymbol{e}_z$$

用几何方法计算Jacobian矩阵

如何处理多于一个自由度的关节?

→ 可看作多个同位置的单自由度关节



用欧拉角表示旋转

$$R = R_x(\theta)R_y(\varphi)R_x(\psi)$$

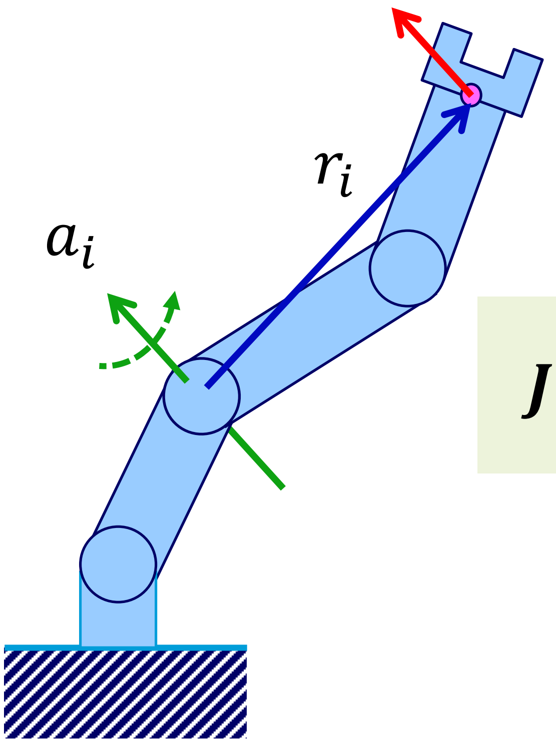
$$\boldsymbol{a}_1 = R_0\boldsymbol{e}_x$$

$$\boldsymbol{a}_2 = R_0R_x\boldsymbol{e}_y$$

$$\boldsymbol{a}_3 = R_0R_xR_y\boldsymbol{e}_x$$

用几何方法计算Jacobian矩阵

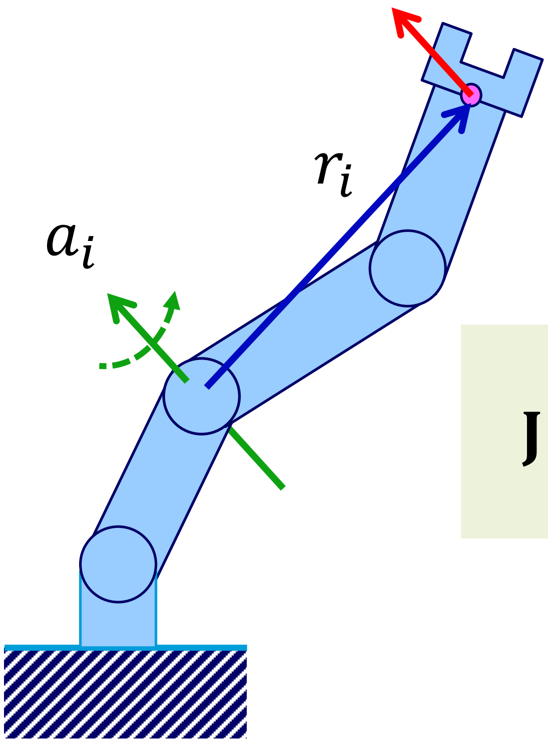
- 需要同时控制末端点朝向？



$$\boldsymbol{J} = (\boldsymbol{a}_1 \times \boldsymbol{r}_1 \quad \boldsymbol{a}_2 \times \boldsymbol{r}_2 \quad \cdots \quad \boldsymbol{a}_n \times \boldsymbol{r}_n)$$

用几何方法计算Jacobian矩阵

- 需要同时控制末端点朝向？

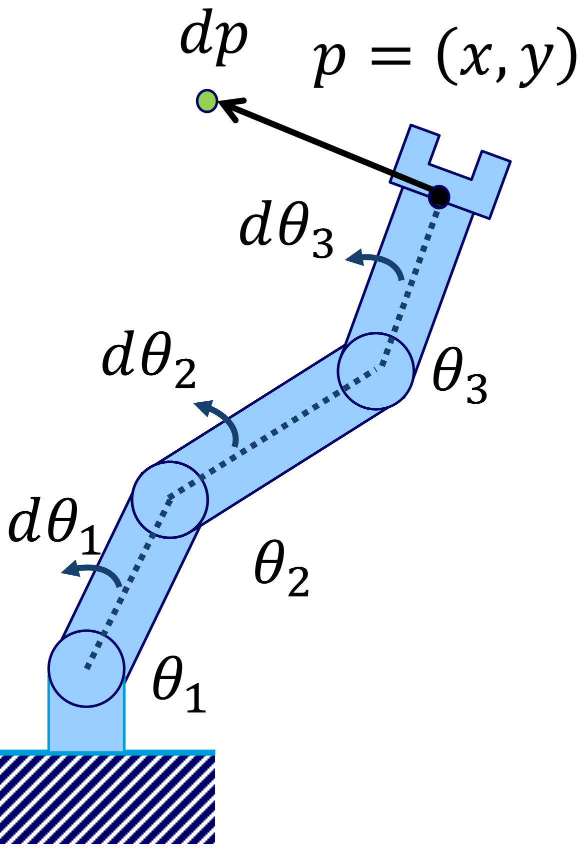


$$\mathbf{J} = \begin{pmatrix} \mathbf{a}_1 \times \mathbf{r}_1 & \mathbf{a}_2 \times \mathbf{r}_2 & \cdots & \mathbf{a}_n \times \mathbf{r}_n \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{pmatrix}$$

利用Jacobian矩阵求解IK问题

$$\boldsymbol{p} = f(\theta_0, \theta_1, \theta_2)$$

$$d\boldsymbol{p} = J d\boldsymbol{\theta}$$



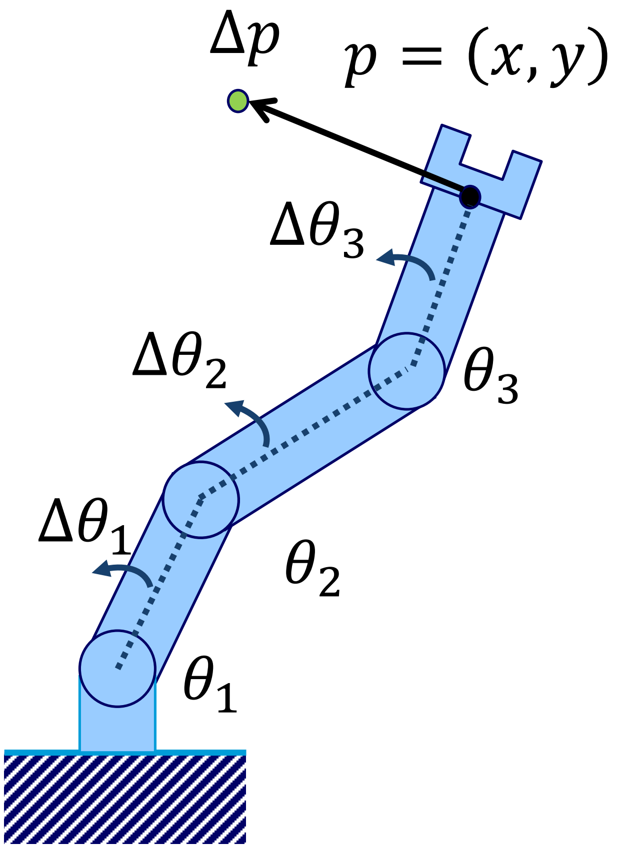
利用Jacobian矩阵求解IK问题

$$\boldsymbol{p} = f(\theta_0, \theta_1, \theta_2)$$

$$d\boldsymbol{p} = J d\boldsymbol{\theta}$$

- 线性近似

$$\Delta\boldsymbol{p} = J\Delta\boldsymbol{\theta}$$



利用Jacobian矩阵求解IK问题

$$\boldsymbol{p} = f(\theta_0, \theta_1, \theta_2)$$

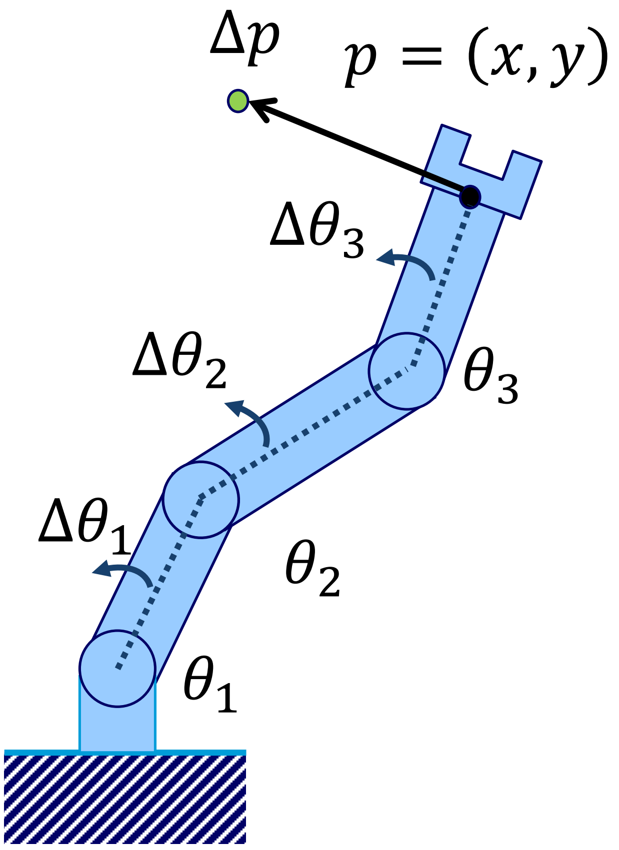
$$d\boldsymbol{p} = J d\boldsymbol{\theta}$$

- 线性近似

$$\Delta\boldsymbol{p} = J\Delta\boldsymbol{\theta}$$

于是

$$\Delta\boldsymbol{\theta} = J^{-1}\Delta\boldsymbol{p}$$



利用Jacobian矩阵求解IK问题

$$\boldsymbol{p} = f(\theta_0, \theta_1, \theta_2)$$

$$d\boldsymbol{p} = J d\boldsymbol{\theta}$$

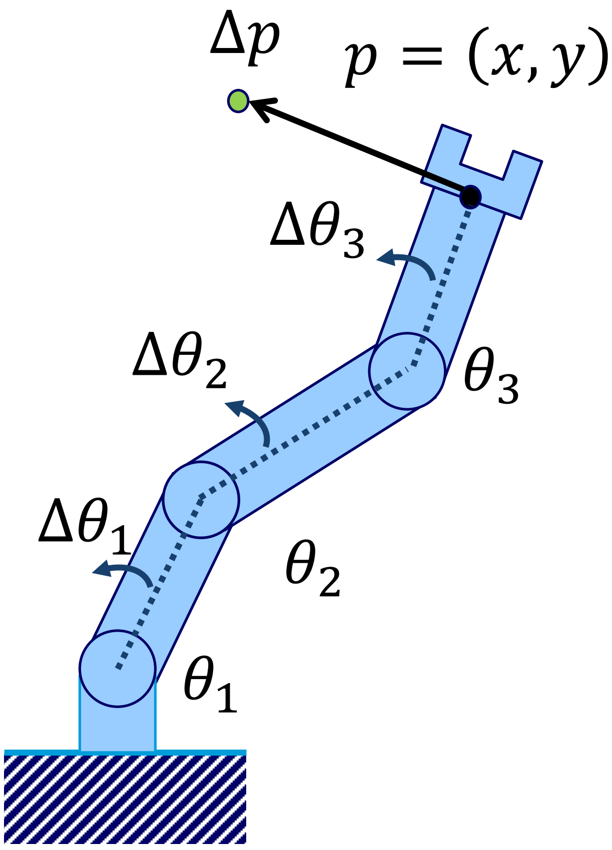
- 线性近似

$$\Delta\boldsymbol{p} = J\Delta\boldsymbol{\theta}$$

于是

$$\Delta\boldsymbol{\theta} = J^{-1}\Delta\boldsymbol{p}$$

对不对?

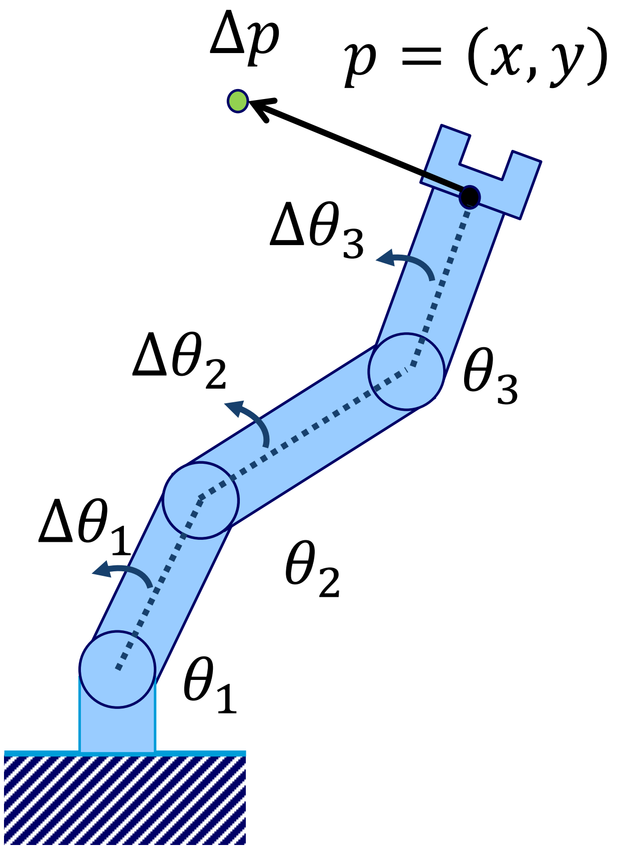


利用Jacobian矩阵求解IK问题

$$\Delta \boldsymbol{p} = J\Delta\boldsymbol{\theta}$$

$$\Delta \boldsymbol{p} = \boxed{\quad \boldsymbol{J} \quad} \ \Delta\boldsymbol{\theta}$$

$\boldsymbol{J}$ 通常不是方阵



线性方程

$$\mathbf{Ax} = \mathbf{b}$$

$$a_{00}x_0 + a_{01}x_1 + \cdots + a_{0n}x_n = b_0$$
$$a_{10}x_0 + a_{11}x_1 + \cdots + a_{1n}x_n = b_1$$
$$\vdots$$
$$a_{n0}x_0 + a_{n1}x_1 + \cdots + a_{nn}x_n = b_n$$

线性方程的解

$$\mathbf{Ax} = \mathbf{b}$$

- A是可逆方阵 ➔ $\boldsymbol{x} = A^{-1}\boldsymbol{b}$
- 欠定/欠约束方程 → 方程数少于未知数
  - **A**是不可逆方阵，或者
  - **A**的行数少于列数

$$\boldsymbol{b} = \boxed{\boldsymbol{A}}\ \mathbf{x}$$

- 过定/过约束方程 → 方程数多于未知数
  - **A**的行数多余列数

$$\boldsymbol{b} = \boxed{\boldsymbol{A}}\ \mathbf{x}$$

过约束问题的最小二乘解

$$\mathbf{Ax} = \mathbf{b}$$

$$\boldsymbol{b} = \boxed{\boldsymbol{A}}\,\mathbf{x}$$

- 过约束问题通常无解
  - 即不存在 x 使 $Ax = b$ 成立

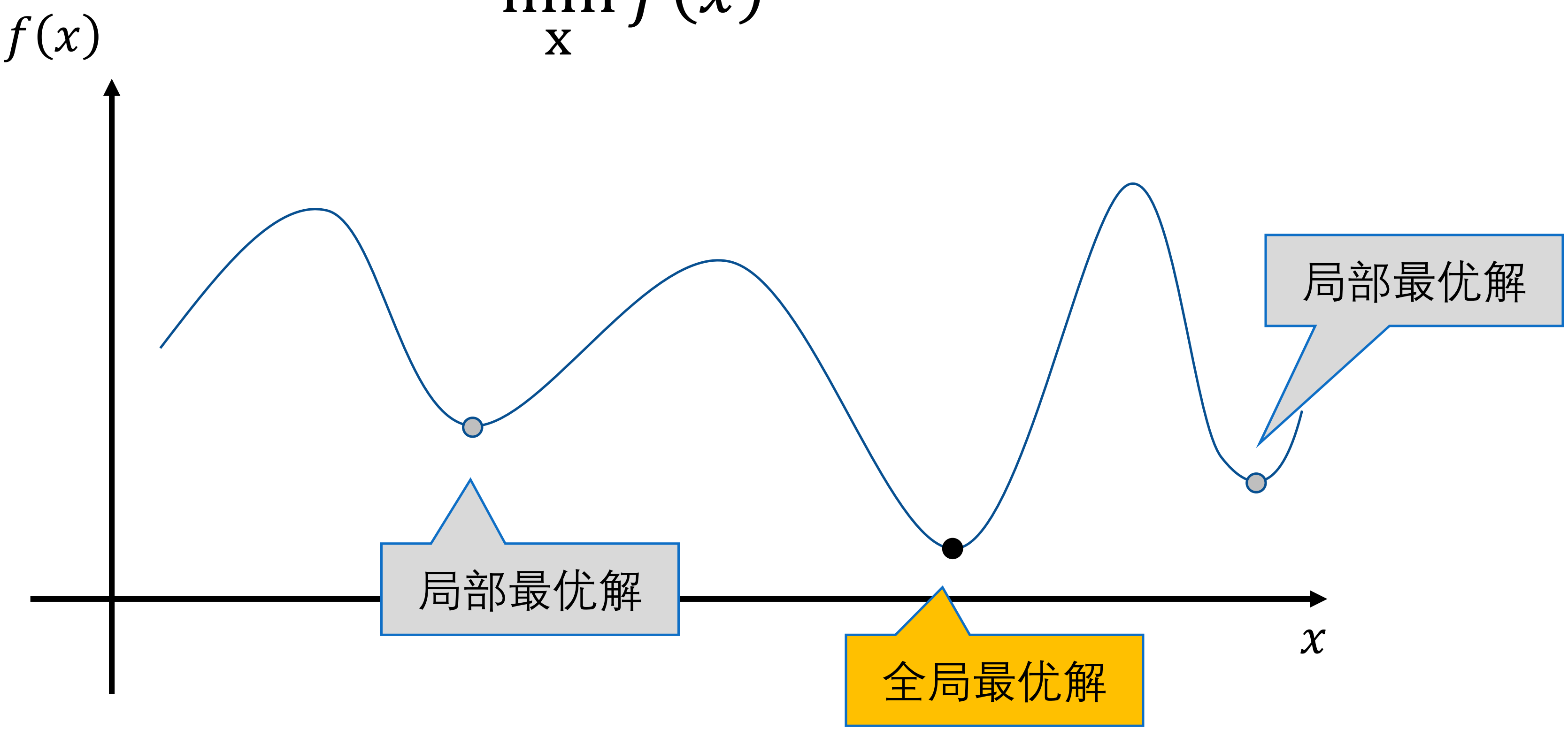
- 转化为优化问题

$$x = \underset{x}{\operatorname{argmin}} \|Ax - b\|_2^2$$

优化问题

求解变量 x 使得函数 $f(x)$ 取得最小值

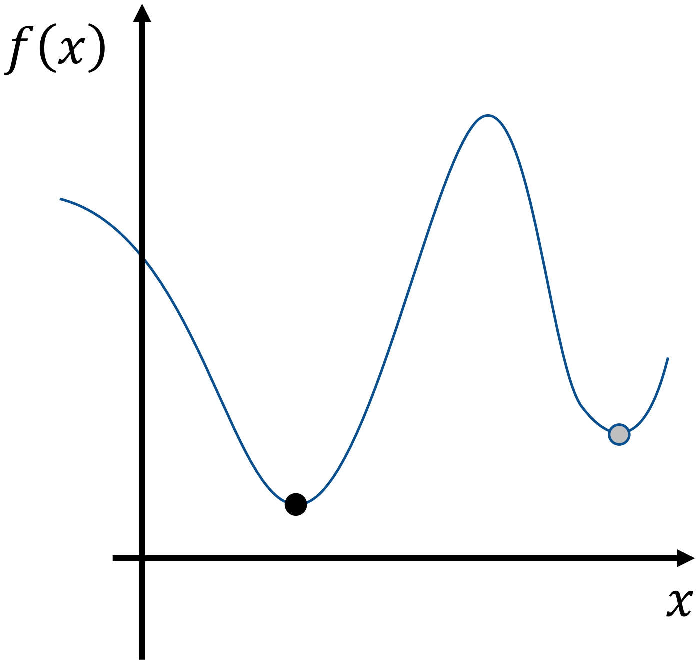
$$\min_{\mathrm{x}} f(x)$$



优化问题

求解变量 x 使得函数 $f(x)$ 取得最小值

$$\min_{\mathrm{x}} f(x)$$



极值条件

$$f'(x) = 0$$

优化问题

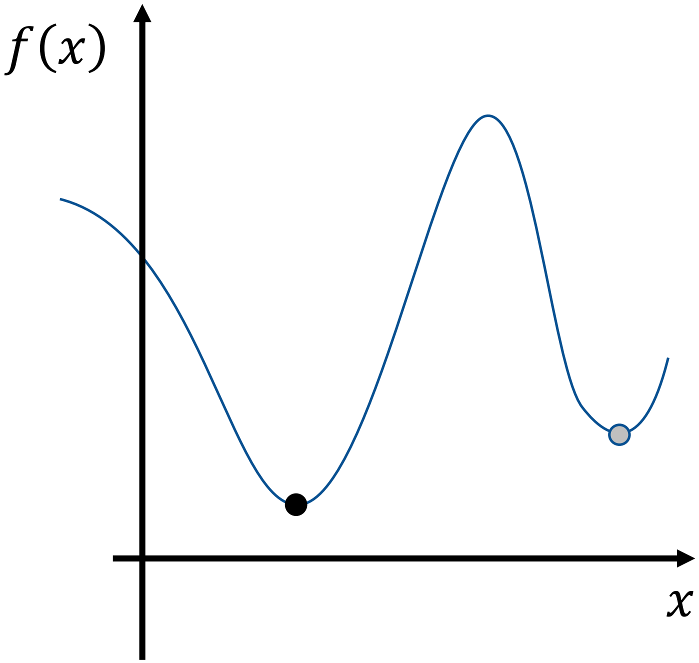
求解变量 x 使得函数 $f(x)$ 取得最小值

$$\min_{\mathrm{x}} f(x)$$

极值条件

$$f'(x) = 0$$

$$f''(x) \geq 0$$



过约束问题的最小二乘解

$$\mathbf{A}\mathbf{x} = \mathbf{b} \qquad \boldsymbol{b} = \boxed{\boldsymbol{A}}\ \mathbf{x}$$

- 优化问题

$$\boldsymbol{x} = \underset{x}{\operatorname{argmin}} \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2$$

$$\boldsymbol{x} = (A^T A)^{-1} A^T \boldsymbol{b}$$

过约束问题的最小二乘解

$$\mathbf{A}\mathbf{x} = \mathbf{b} \qquad\qquad \boldsymbol{b} = \boxed{\boldsymbol{A}}\,\mathbf{x}$$

- 优化问题

$$\boldsymbol{x} = \underset{x}{\operatorname{argmin}} \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2$$

$$f(x) = \|Ax - b\|_2^2$$
$$= (Ax-b)^T(Ax-b)$$
$$= x^TA^TAx - 2x^TA^Tb + b^Tb$$
$$\frac{\partial f}{\partial x} = 2A^TAx - 2A^Tb$$
$$A^TAx = A^Tb$$
$$x = (A^TA)^{-1}A^Tb$$

$$\boldsymbol{x} = \boxed{(A^TA)^{-1}A^T}\,\boldsymbol{b}$$

(Moore-Penrose) Pseudoinverse

伪逆 / 拟逆

(Moore-Penrose) Pseudoinverse

- 过约束问题（回归拟合、最小二乘……）

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

$$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Ax} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^+\mathbf{b}$$

$$\boxed{\mathbf{A}^+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T}$$

(Moore-Penrose) Pseudoinverse

- 过约束问题（回归拟合、最小二乘……）

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

$$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A}\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{+}\mathbf{b}$$

$$\boxed{\mathbf{A}^{+} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T}$$

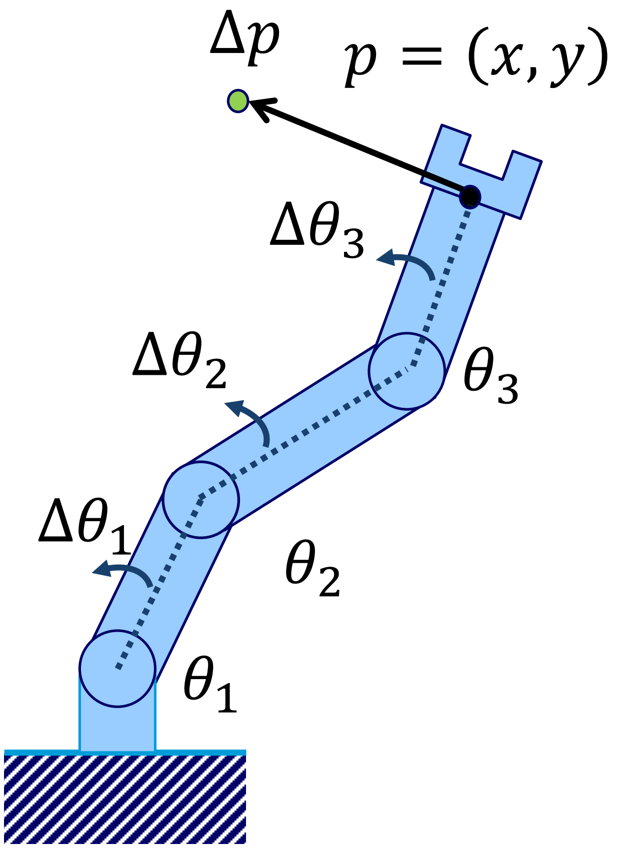
利用Jacobian矩阵求解IK问题

$$\Delta \boldsymbol{p} = J\Delta \boldsymbol{\theta}$$

$$\Delta \boldsymbol{p} = \boxed{\quad \boldsymbol{J} \quad} \Delta \boldsymbol{\theta}$$

$\boldsymbol{J}$ 通常不是方阵

通常是欠约束问题



欠约束问题

$$\mathbf{Ax} = \mathbf{b} \qquad \boldsymbol{b} = \boxed{\quad \boldsymbol{A} \quad} \; \mathbf{x}$$

- 欠约束问题通常解不唯一

欠约束问题

$$\mathbf{Ax = b} \qquad \boldsymbol{b} = \boxed{\quad \boldsymbol{A} \quad} \ \mathbf{x}$$

- 欠约束问题通常解不唯一
- 寻找“长度”最短的解

欠约束问题

$$\mathbf{Ax} = \mathbf{b} \qquad \boldsymbol{b} = \boxed{\boldsymbol{A}}\ \mathbf{x}$$

- 欠约束问题通常解不唯一
- 寻找“长度”最短的解

$$\min_{x} \|x\|_2^2$$

$$\text{s.t.}\ \ Ax = b$$

带约束的优化问题

$$\min_{x} f(x)$$

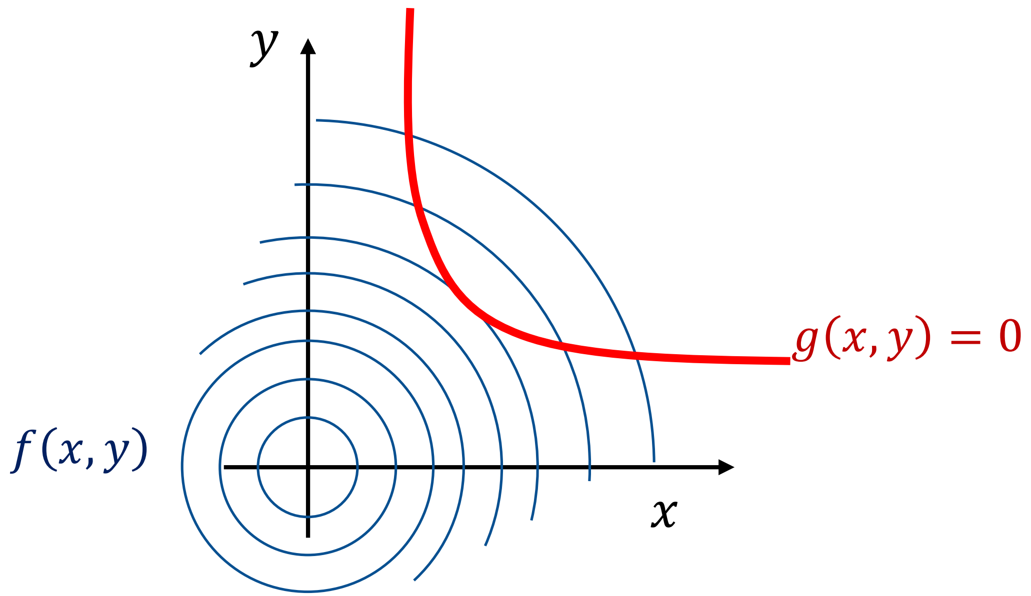
$$s.t.\quad g(x) = 0 \qquad \text{等式约束}$$

$$h(x) \geq 0 \qquad \text{不等式约束}$$

带约束的优化问题

$$\min_{x} f(x)$$

$$s.t.\ \ g(x) = 0$$ 等式约束



等值上的导为0，前进方向与导数方向垂直，当f导数与g(x,y)导数一样的时候，值最小

拉格朗日乘子法

$$\min_x f(x)$$
$$s.t.\ \ g(x) = 0$$

定义拉格朗日函数

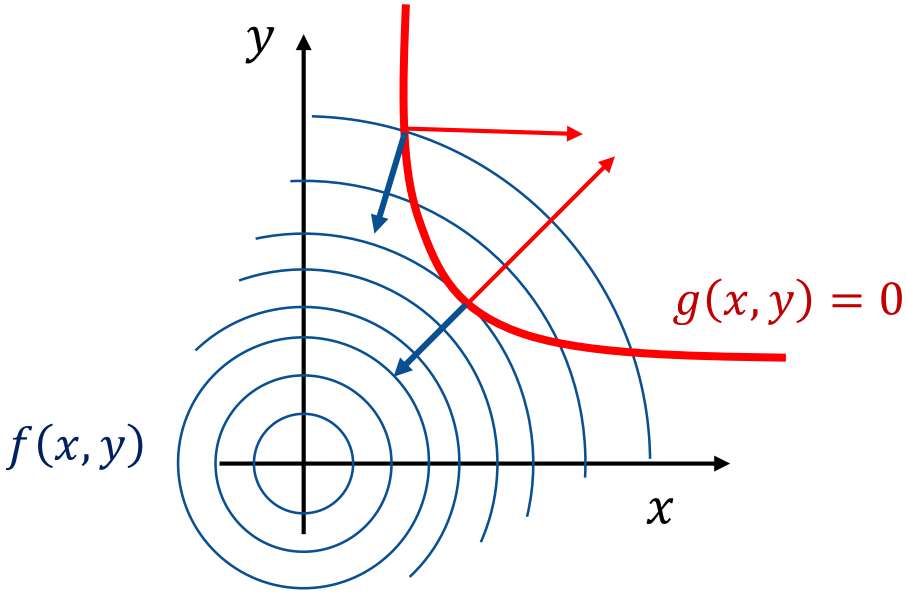
$$L(x,\lambda) = f(x) + \lambda^T g(x)$$

则优化问题的极值点满足条件

方向一样

$$\frac{dL}{dx} = f'^{(x)} + \lambda^T g'(x) = 0$$

$$\frac{dL}{d\lambda} = g(x) = 0$$



欠约束问题

$$\mathbf{Ax} = \mathbf{b} \qquad \boldsymbol{b} = \boxed{\boldsymbol{A}}\ \mathbf{x}$$

- 寻找“长度”最短的解

$$\min_x \|x\|_2^2$$
$$\text{s.t.}\ \ Ax = b$$

$$L(x, \lambda) = x^T x + \lambda^T (Ax - b)$$
$$\frac{\partial L}{\partial x} = 2x + \lambda^T A = 0$$
$$x = -\frac{1}{2}\lambda^T A$$
$$\frac{\partial L}{\partial \lambda} = Ax - b = 0$$

- 应用拉格朗日乘子法

$$x = A^T (AA^T)^{-1} b$$

欠约束问题

$$\mathbf{Ax} = \mathbf{b} \qquad \boldsymbol{b} = \boxed{\boldsymbol{A}}\ \mathbf{x}$$

- 寻找“长度”最短的解

$$\min_{x} \|x\|_2^2$$

$$\text{s.t. } Ax = b$$

- 应用拉格朗日乘子法

$$x = \boxed{A^T(AA^T)^{-1}} b$$

(Moore-Penrose) Pseudoinverse

(Moore-Penrose) Pseudoinverse

- 过约束问题（回归拟合、最小二乘……）
  - 左逆

$$\boxed{\mathbf{A}^{+} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}}$$

- 欠约束问题
  - 右逆

$$\boxed{\mathbf{A}^{+} = \mathbf{A}^{T}(\mathbf{A}\mathbf{A}^{T})^{-1}}$$

利用Jacobian矩阵求解IK问题

$$\boldsymbol{p} = f(\theta_0, \theta_1, \theta_2)$$

$$d\boldsymbol{p} = J d\boldsymbol{\theta}$$

- 线性近似

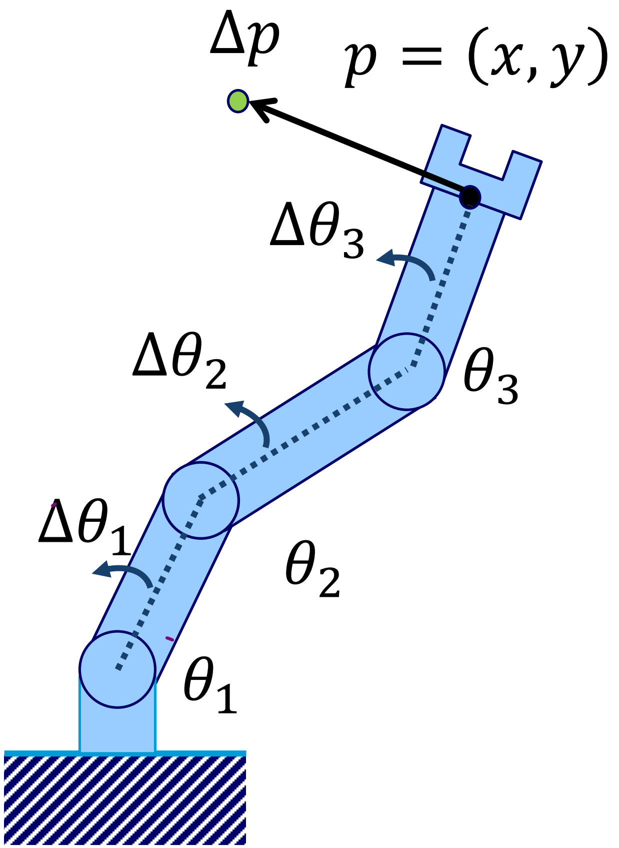
$$\Delta\boldsymbol{p} = J\Delta\boldsymbol{\theta}$$

于是

$$\cancel{\Delta\boldsymbol{\theta} = J^{-1}\Delta\boldsymbol{p}}$$

右逆？

$$\Delta\boldsymbol{\theta} = J^T(JJ^T)^{-1}\Delta\boldsymbol{p}$$



利用Jacobian矩阵求解IK问题

$$\boldsymbol{p} = f(\theta_0, \theta_1, \theta_2)$$

$$d\boldsymbol{p} = J d\boldsymbol{\theta}$$

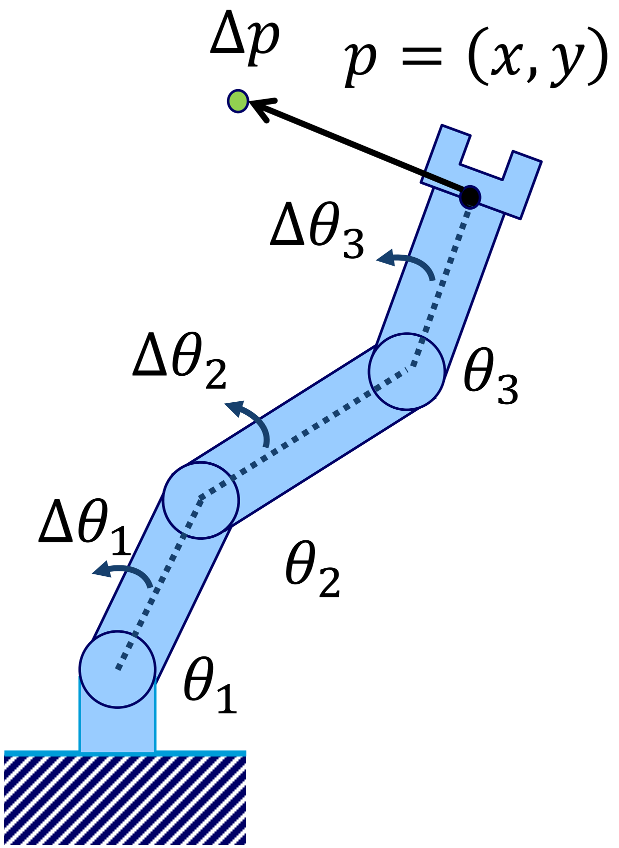
- 线性近似

$$\Delta\boldsymbol{p} = J\Delta\boldsymbol{\theta}$$

于是

$$\cancel{\Delta\boldsymbol{\theta} = J^{-1}\Delta\boldsymbol{p}}$$

$$\Delta\boldsymbol{\theta} = J^{+}\Delta\boldsymbol{p}$$



利用Jacobian矩阵求解IK问题

$$\boldsymbol{p} = f(\theta_0, \theta_1, \theta_2)$$

$$d\boldsymbol{p} = J d\boldsymbol{\theta}$$

- 线性近似

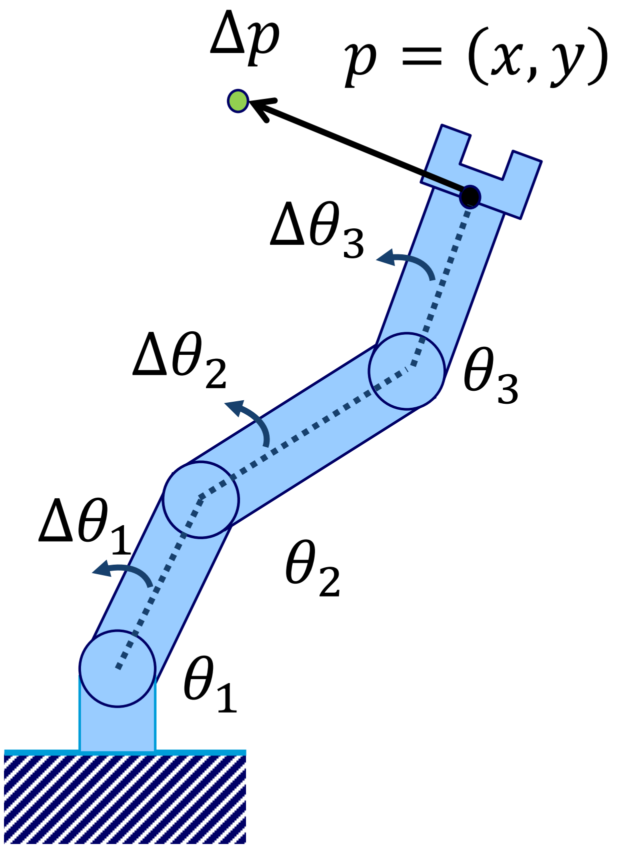
$$\Delta\boldsymbol{p} = J\Delta\boldsymbol{\theta}$$

于是

$$\cancel{\Delta\boldsymbol{\theta} = J^{-1}\Delta\boldsymbol{p}}$$

$$\Delta\boldsymbol{\theta} = J^{+}\Delta\boldsymbol{p}$$

需要迭代求解



伪逆与零空间（null-space）

- 欠约束问题

$$\mathbf{Ax} = \mathbf{b} \qquad \boldsymbol{b} = \boxed{\quad \boldsymbol{A} \quad} \ \mathbf{x}$$

的通解为

$$x = A^{+}b + (A^{+}A - I)z$$

其中 $\forall z \in \mathbb{R}^{n}$

- $(A^{+}A - I)$ 定义了 A 的零空间
- 更改 z 可以在满足约束的条件下完成其他任务

利用Jacobian矩阵求解IK问题

- 迭代求解

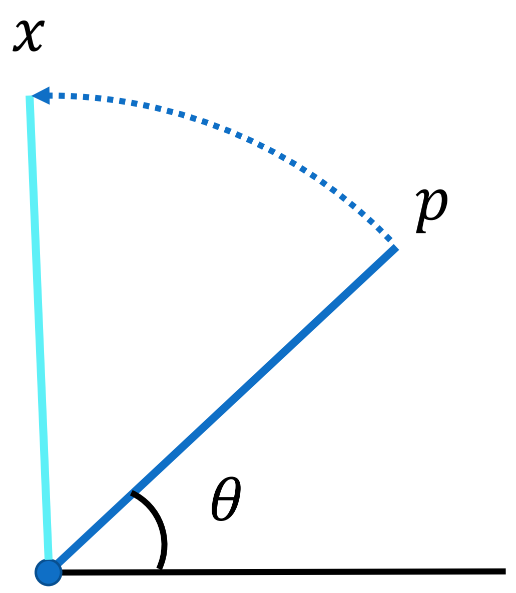
Loop:

$$\Delta p = x - f(\theta)$$

$$J = \frac{\partial f}{\partial \theta}(\theta)$$

$$\Delta\theta = J^{+}\Delta p$$

$$\theta \leftarrow \theta + \Delta\theta$$



利用Jacobian矩阵求解IK问题

- 迭代求解

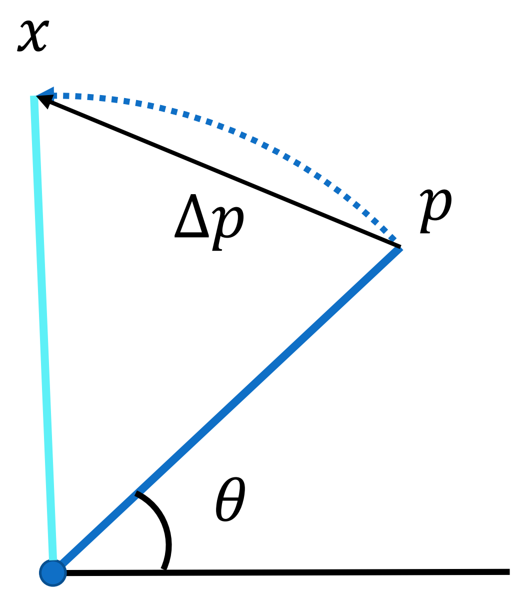
Loop:

$$\Delta p = x - f(\theta)$$

$$J = \frac{\partial f}{\partial \theta}(\theta)$$

$$\Delta \theta = J^{+} \Delta p$$

$$\theta \leftarrow \theta + \Delta \theta$$



利用Jacobian矩阵求解IK问题

- 迭代求解

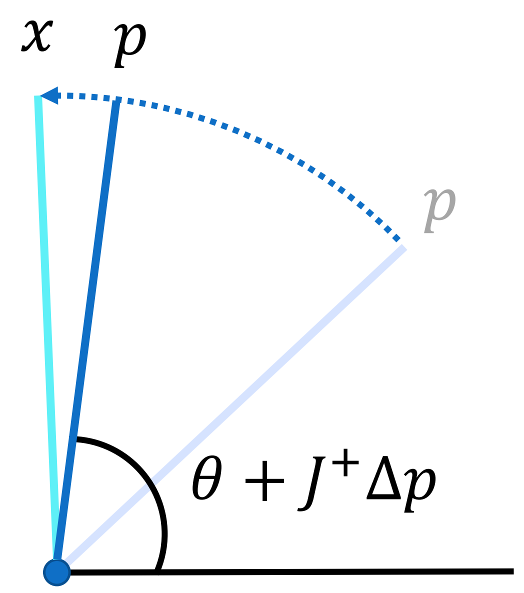
Loop:

$$\Delta p = x - f(\theta)$$

$$J = \frac{\partial f}{\partial \theta}(\theta)$$

$$\Delta \theta = J^{+} \Delta p$$

$$\theta \leftarrow \theta + \Delta \theta$$



利用Jacobian矩阵求解IK问题

- 迭代求解

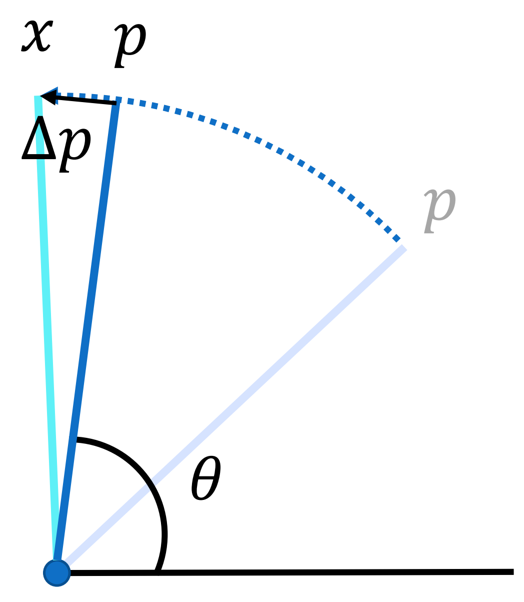
Loop:

$$\Delta p = x - f(\theta)$$

$$J = \frac{\partial f}{\partial \theta}(\theta)$$

$$\Delta\theta = J^{+}\Delta p$$

$$\theta \leftarrow \theta + \Delta\theta$$



利用Jacobian矩阵求解IK问题

- 迭代求解

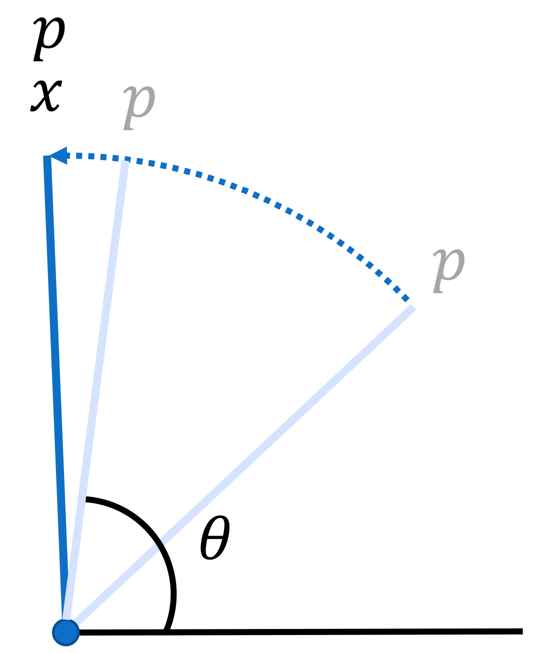
Loop **直到 $\boldsymbol{\Delta p < \varepsilon}$ 或 到达最大循环次数**

$$\Delta p = x - f(\theta)$$

$$J = \frac{\partial f}{\partial \theta}(\theta)$$

$$\Delta\theta = J^{+}\Delta p$$

$$\theta \leftarrow \theta + \Delta\theta$$



(Moore-Penrose) Pseudoinverse

$$\boxed{\mathbf{A}^{+} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}}$$

$$\boxed{\mathbf{A}^{+} = \mathbf{A}^{T}(\mathbf{A}\mathbf{A}^{T})^{-1}}$$

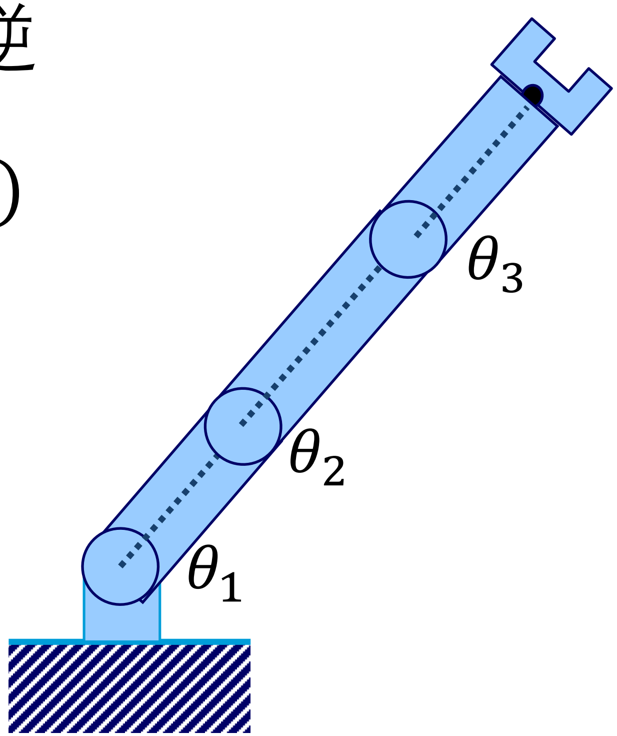
- 以上计算要求 A^TA 或 AA^T 可逆

(Moore-Penrose) Pseudoinverse

$$\mathbf{A}^{+} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}$$

$$\mathbf{A}^{+} = \mathbf{A}^{T}(\mathbf{A}\mathbf{A}^{T})^{-1}$$

- 以上计算要求 $A^T A$ 或 AA^T 可逆
- 然而可能 $\mathrm{rank}(A) < \min(m, n)$



(Moore-Penrose) Pseudoinverse

- 利用奇异值分解（SVD）计算伪逆

$$A = U \, \Sigma \, V^T$$

$$\boldsymbol{A} = \boldsymbol{U} \underbrace{\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\Sigma} \boldsymbol{V}^T$$

(Moore-Penrose) Pseudoinverse

- 利用奇异值分解（SVD）计算伪逆

$$A^{+} = \mathrm{V}\,\Sigma^{+}\,U^{T}$$

$$\boldsymbol{A}^{+} = V \underbrace{\begin{bmatrix} \frac{1}{\sigma_1} & 0 \\ 0 & \frac{1}{\sigma_2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\Sigma^{+}} U^{T}$$

$\Sigma^{+} = \Sigma$ 中**非零**主元取逆

(Moore-Penrose) Pseudoinverse

- 矩阵伪逆在奇异点附近有不稳定问题

$$A = U \begin{bmatrix} 1 & 0 \\ 0 & 0.0001 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} V^T$$

$$A^+ = V \begin{bmatrix} 1 & 0 \\ 0 & 10000 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} U^T$$

(Moore-Penrose) Pseudoinverse

- Damped Pseudoinverse

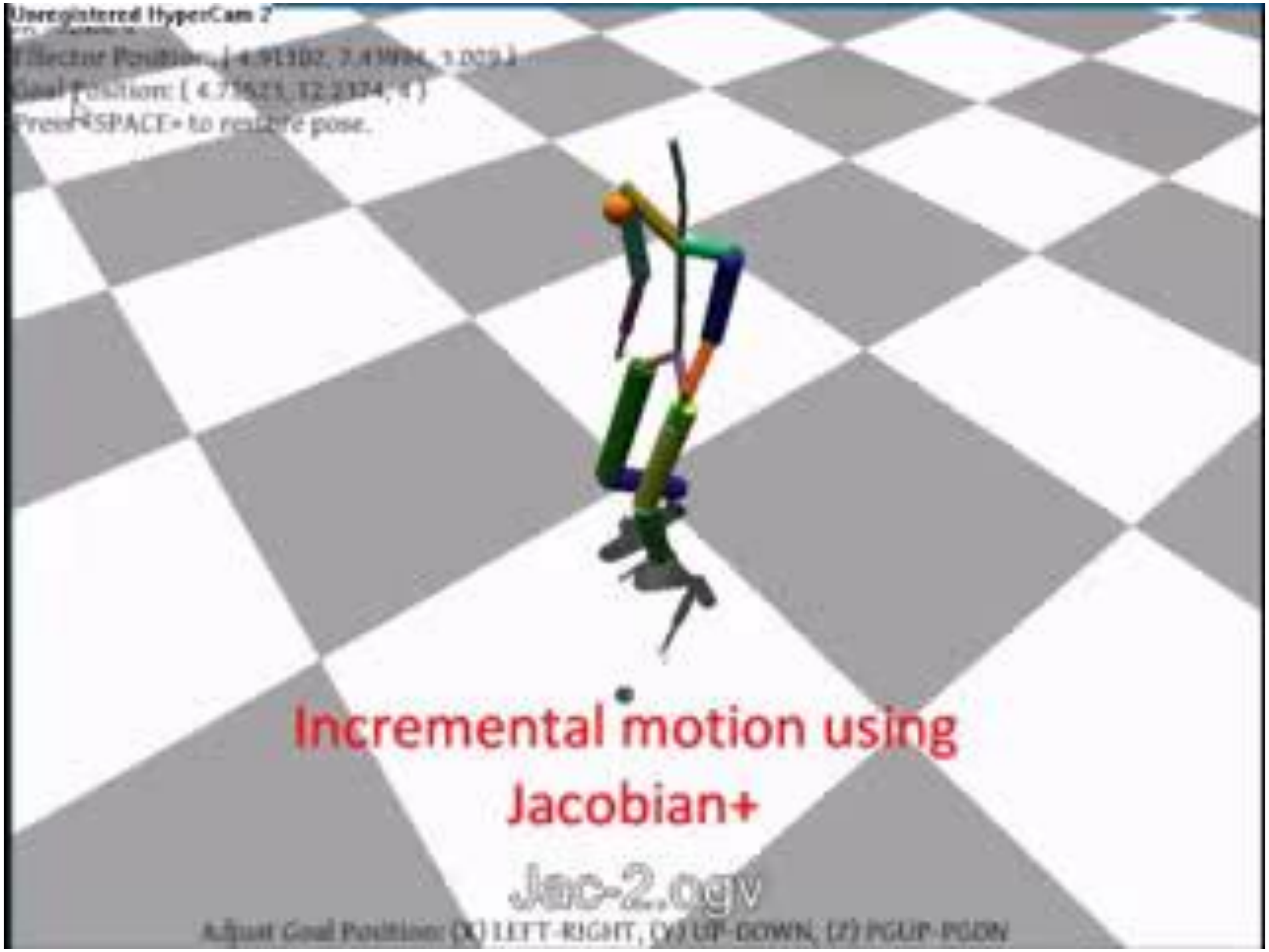
$$A^* = A^T(AA^T + \lambda^2 I)^{-1}$$

$$A^* = (A^T A + \lambda^2 I)^{-1} A^T$$

$$A^* = V\,\Sigma^*\,U^T$$

$$\Sigma^* = \begin{bmatrix} \dfrac{\sigma_1}{\sigma_1^2 + \lambda^2} & 0 \\ 0 & \dfrac{\sigma_2}{\sigma_2^2 + \lambda^2} \\ 0 & 0 \end{bmatrix}$$

Jacobian Pseudoinverse IK



(Moore-Penrose) Pseudoinverse

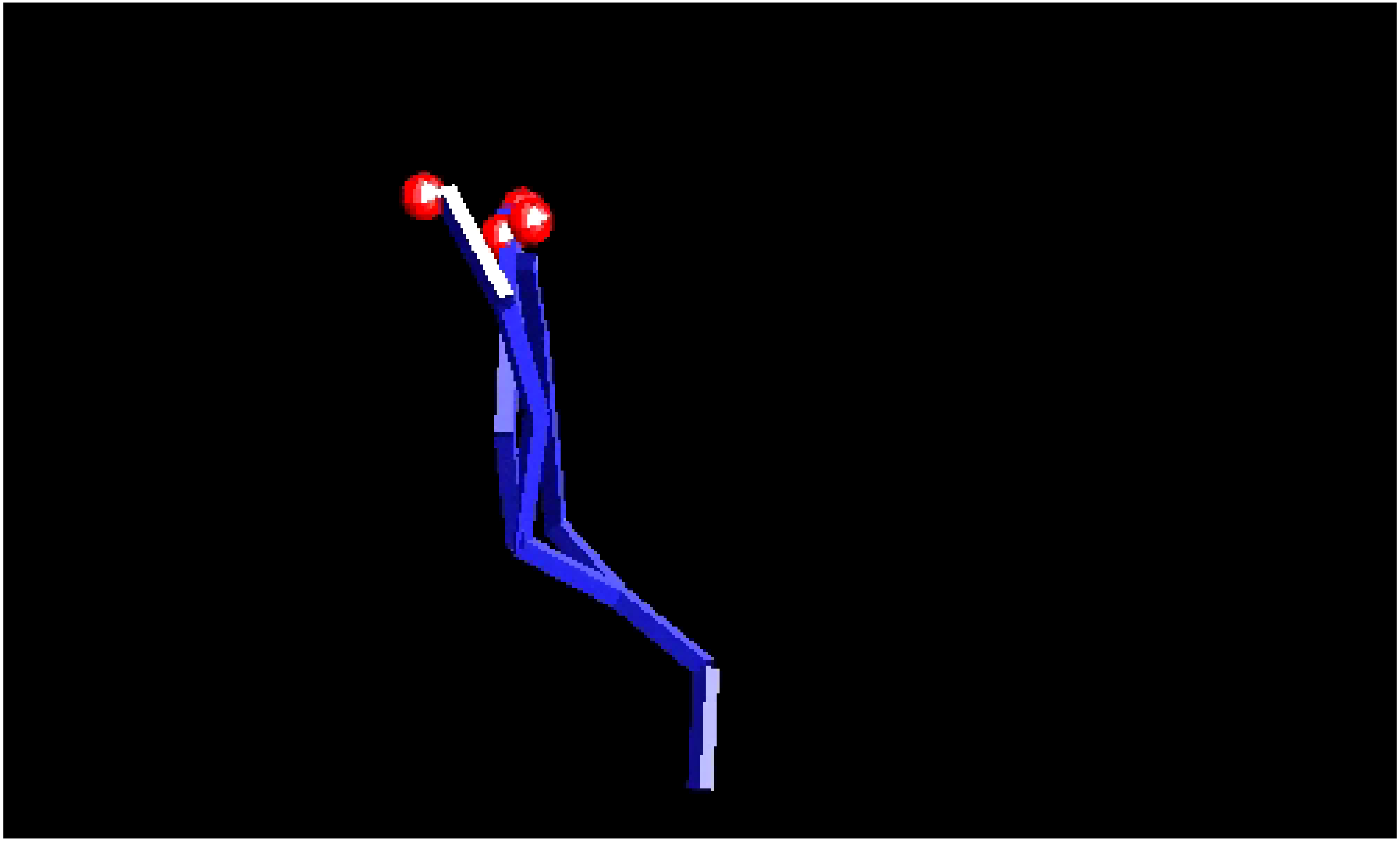
- Selectively Damped Pseudoinverse

$$A^* = V\,\Sigma^*\,U^T$$

$$\Sigma^* = \begin{bmatrix} \dfrac{\sigma_1}{\sigma_1^2 + \boldsymbol{\lambda_1^2}} & 0 \\ 0 & \dfrac{\sigma_2}{\sigma_2^2 + \boldsymbol{\lambda_2^2}} \\ 0 & 0 \end{bmatrix}$$

奇异值对应关节旋转的角度

Selectively Damped Least Squares

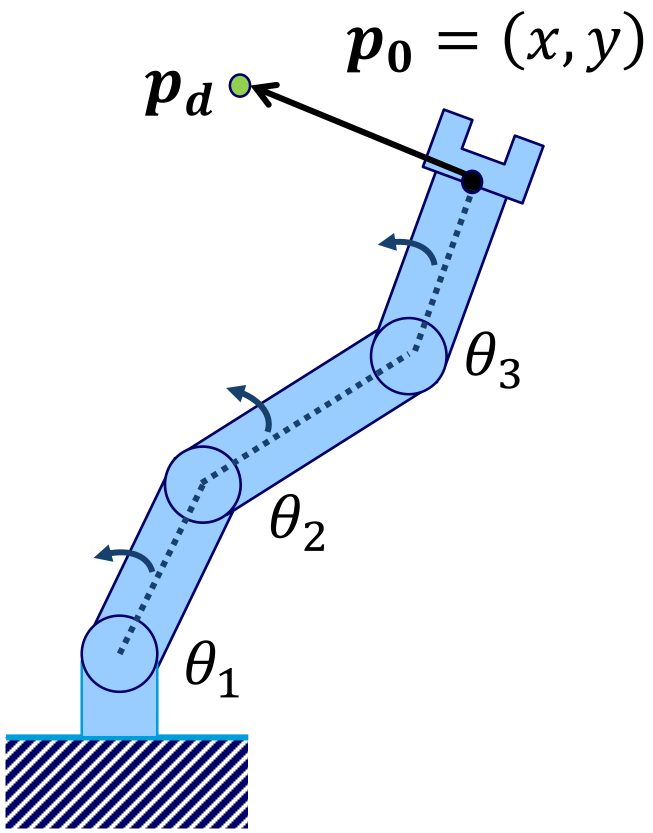


Samuel R. Buss and Jin-Su Kim. 2005. ***Selectively Damped Least Squares for Inverse Kinematics***. *Journal of Graphics Tools* 10, 3 (January 2005), 37–49.

IK 作为优化问题

$$\boldsymbol{p_0} = f(\boldsymbol{\theta}_0)$$

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \|f(\boldsymbol{\theta}) - \boldsymbol{p_d}\|^2$$



梯度

函数 $f(\boldsymbol{\theta}): \mathbb{R}^n \mapsto \mathbb{R}$ 的梯度

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial \theta_1} \\ \frac{\partial f}{\partial \theta_1} \\ \vdots \\ \frac{\partial f}{\partial \theta_1} \end{bmatrix} = \left(\frac{\partial f}{\partial \boldsymbol{\theta}}\right)^T = J^T$$

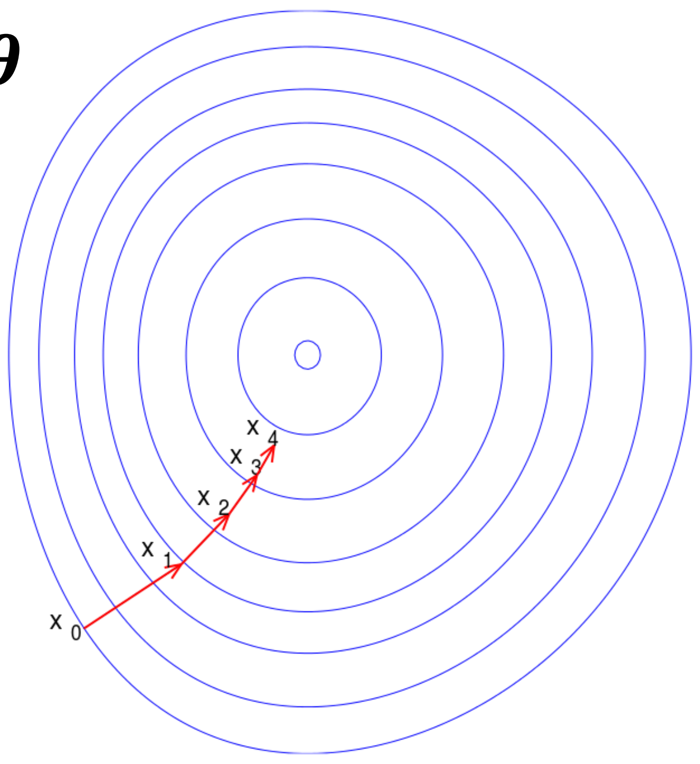
- 梯度代表函数**增加**的方向

梯度下降法

求函数 $f(\boldsymbol{\theta}): \mathbb{R}^n \mapsto \mathbb{R}$ 的最小值

- 梯度代表函数**增加**的方向
- 应该向着梯度相反的方向改变 $\boldsymbol{\theta}$

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \alpha \nabla f(\boldsymbol{\theta}_i)$$



IK 作为优化问题

$$g(\boldsymbol{\theta}) = \frac{1}{2}\|f(\boldsymbol{\theta}) - \boldsymbol{p}_d\|^2$$

$$\nabla g^T = \frac{\partial g}{\partial \boldsymbol{\theta}} = \frac{\partial g}{\partial f}\frac{\partial f}{\partial \boldsymbol{\theta}} = \underset{\substack{\uparrow \\ -\Delta p^T}}{(f(\boldsymbol{\theta}) - p_d)^T} \underset{\substack{\uparrow \\ J}}{\frac{\partial f}{\partial \theta}}$$

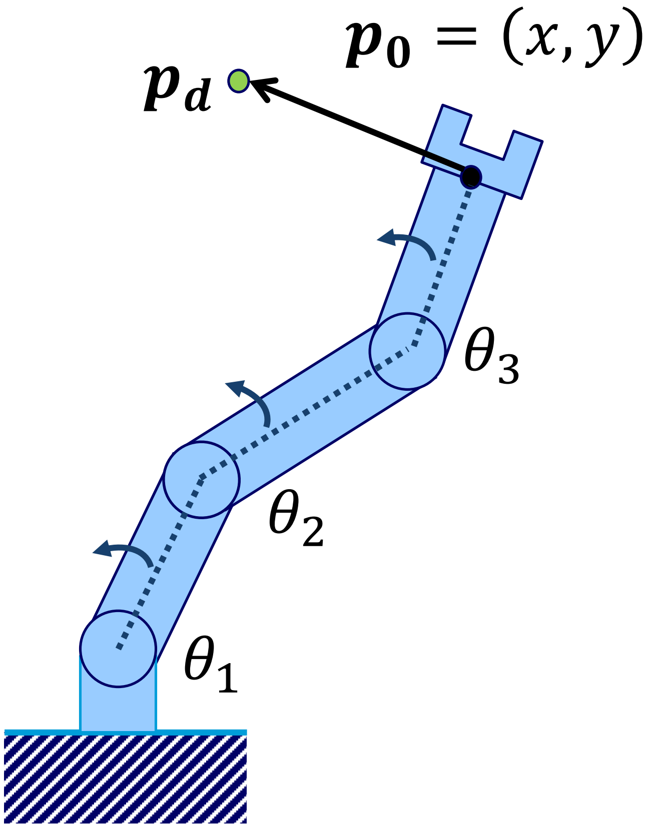
$$\nabla g = -J^T \Delta \boldsymbol{p}$$

梯度下降法

$$\boldsymbol{p}_{\mathbf{0}} = f(\boldsymbol{\theta}_0)$$

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \|f(\boldsymbol{\theta}) - \boldsymbol{p}_{\boldsymbol{d}}\|^2$$

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \alpha \nabla g \\ &= \boldsymbol{\theta}_t + \alpha J^T \Delta \boldsymbol{p} \end{aligned}$$



也被称作 **Jacobian Transpose** 方法

Jacobian方法

- 优点
  - 基于优化问题的数值解法
  - 容易加入约束
  - 结果较为稳定

- 缺点
  - 计算量普遍较大
  - 对超参数敏感

基于数据的方法

Data-driven IK

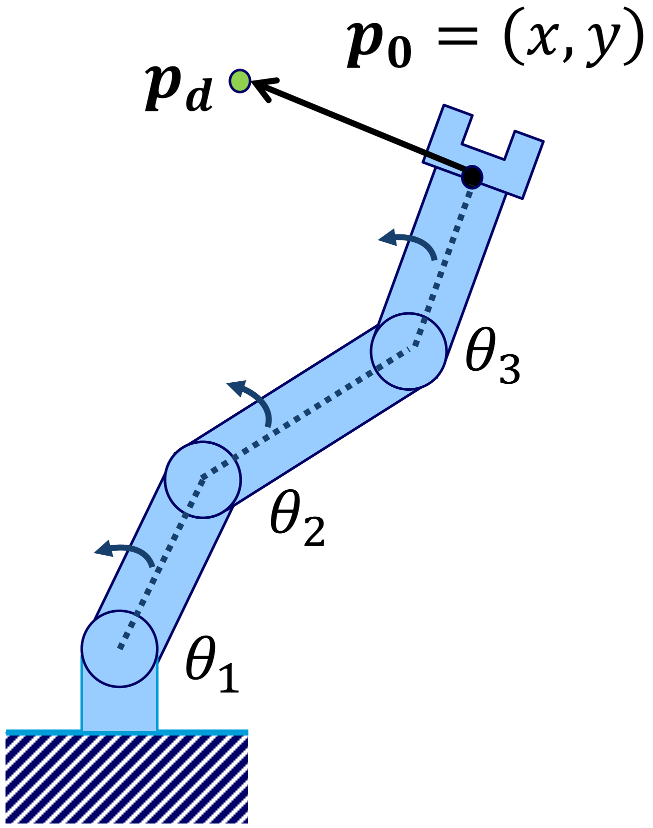
基于数据的方法

- IK问题通常是欠约束的
- 如何确定哪些解更好
  - 自然、可信
  - 运动具有特定风格
- 利用数据来约束解的范围

基本思路

- 优化问题

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \| f(\boldsymbol{\theta}) - \boldsymbol{p_d} \|^2$$



基本思路

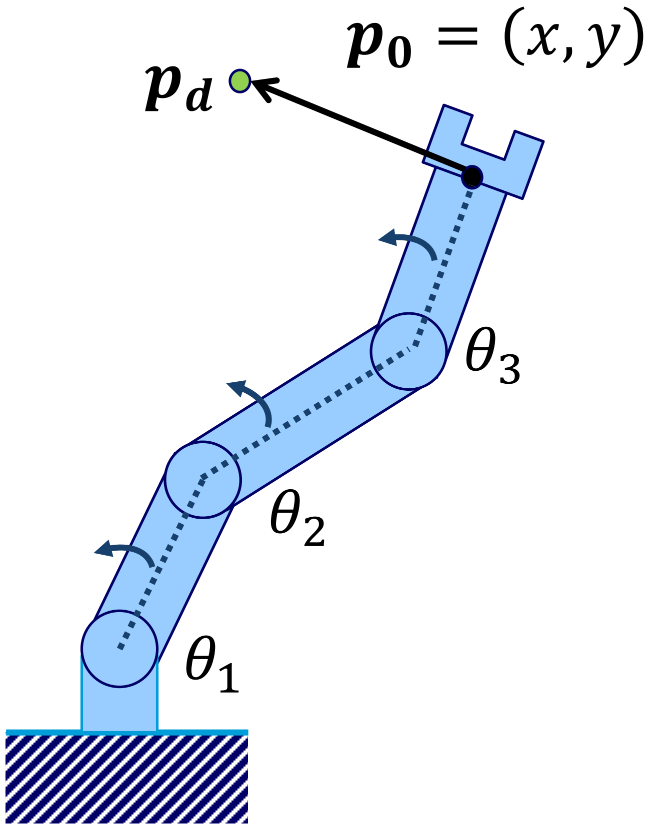
- 优化问题

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \| f(\boldsymbol{\theta}) - \boldsymbol{p}_d \|^2$$

- 基于数据的动作统计模型

$$g(\theta|D)$$

D: 动作数据集



基本思路

- 优化问题

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2}\|f(\boldsymbol{\theta})-\boldsymbol{p}_{\boldsymbol{d}}\|^2 - w_{\mathrm{D}}\, g(\theta|D)$$

D: 动作数据集

基本思路

- 优化问题

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2}\|f(\boldsymbol{\theta})-\boldsymbol{p_d}\|^2 - w_{\mathrm{D}}\, g(\theta|D)$$

D: 动作数据集

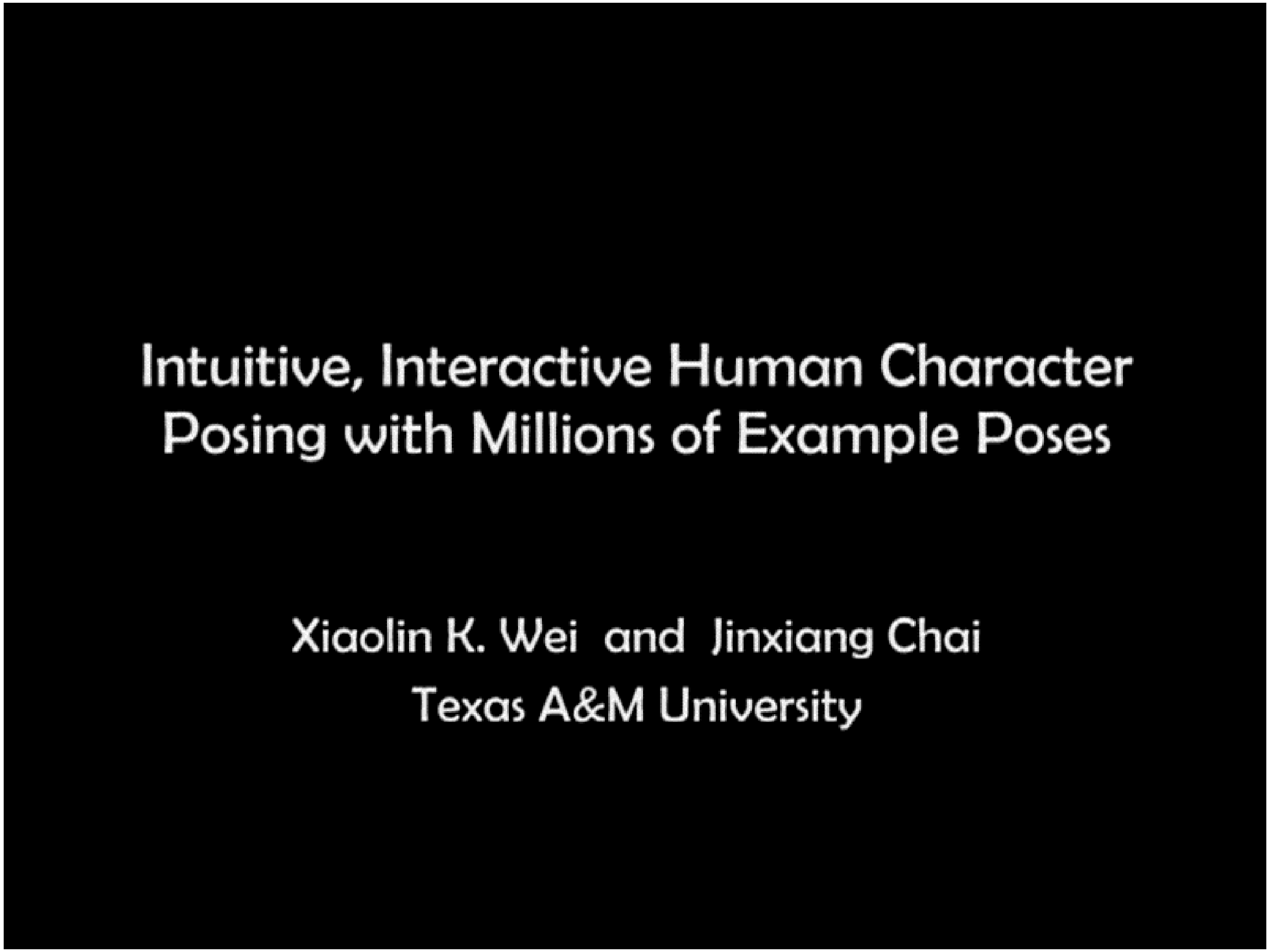
- 例如：

基本上为 Gaussian

$$g(\boldsymbol{\theta}|D) = e^{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})}$$

μ, Σ 为 D 中数据的均值和方差

基于数据的方法例子



Xiaolin Wei and Jinxiang Chai. 2011. ***Intuitive Interactive Human-Character Posing with Millions of Example Poses***. *IEEE Computer Graphics and Applications* 31, 4 (July 2011), 78–88.

基本思路

- 优化问题

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2}\|f(\boldsymbol{\theta}) - \boldsymbol{p_d}\|^2 - w_{\mathrm{D}}\, g(\theta|D)$$

D: 动作数据集

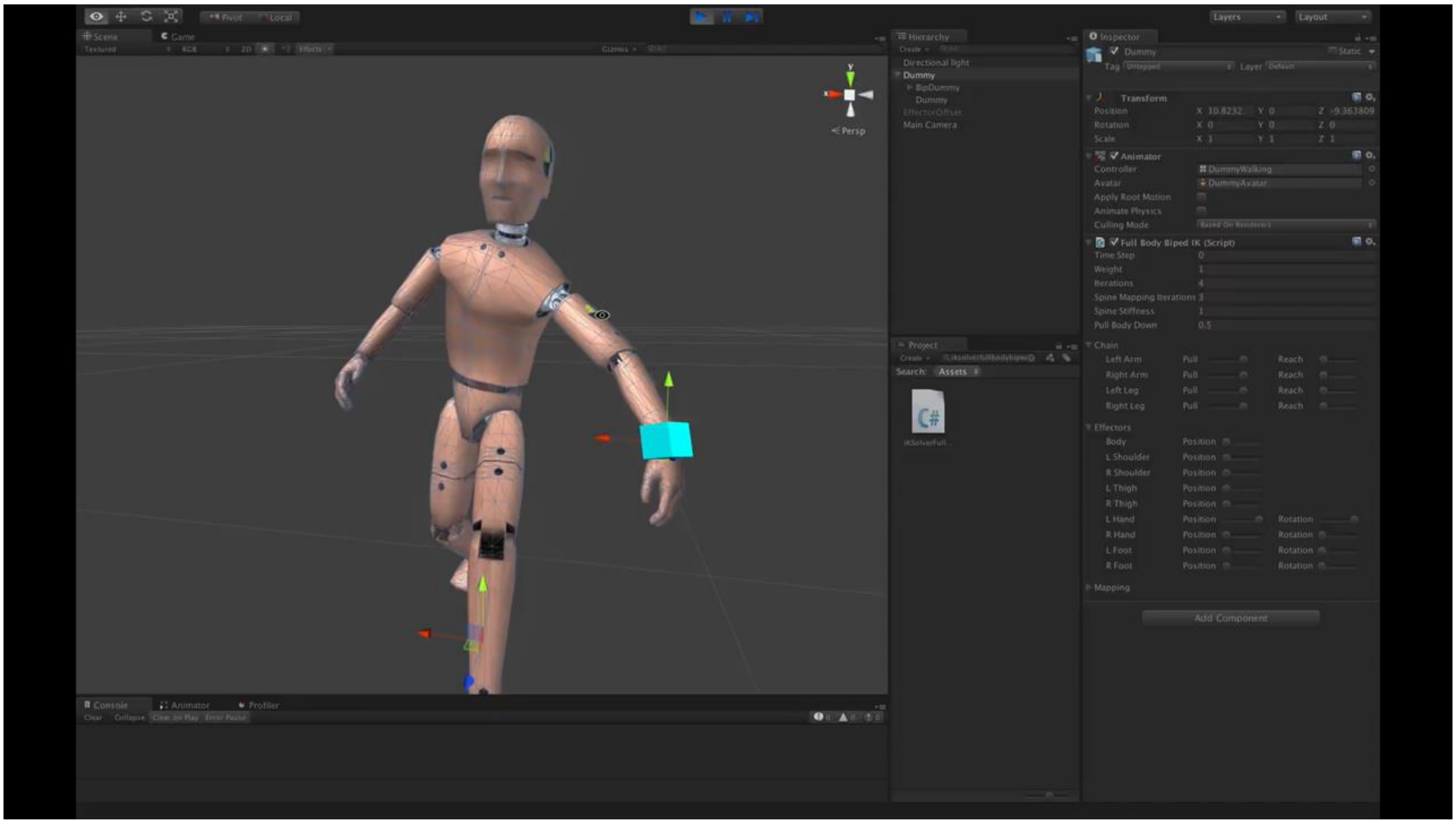
- 其他统计模型
  - Mixture of Gaussian
  - Gaussian Process
  - Neural Networks (e.g. GAN)

混合 IK 方法

- 针对不同部位使用不同的IK方法
  - 手臂、腿可以简化为two-link IK问题

- Maya IK handles
  - HumanIK

混合 IK 方法



Unity Final IK Asset

总结

- 两关节 (Two-link) IK问题的解析解
- 一般性问题的数值解
  - 启发式方法：
    - CCD， FABRIK
  - 基于Jacobian矩阵的方法
    - Jacobian inverse
    - Jacobian transpose
  - 基于数据的方法
- 没有涉及的部分
  - 有约束的IK
  - 有环结构的IK

A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. 2018. ***Inverse Kinematics Techniques in Computer Graphics: A Survey***. *Computer Graphics Forum* 37, 6 (September 2018), 35–58

有没有问题

Any Questions?