关节动画

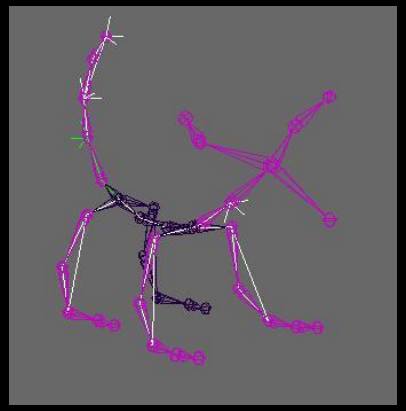
金小刚

Email: jin@cad.zju.edu.cn

浙江大学CAD&CG国家重点实验室

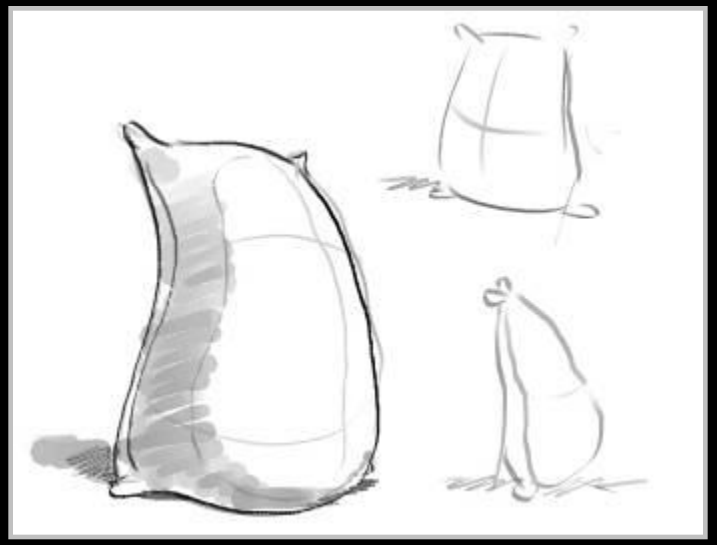
蒙民伟楼512室

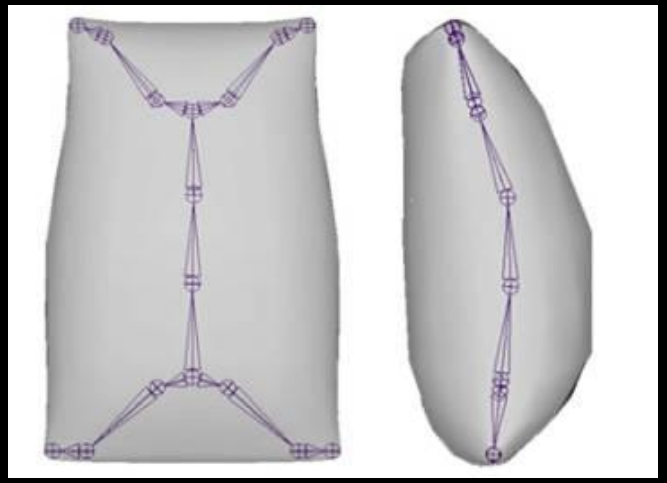
- 在计算机动画中，加入人、动物这样的角色会使画面活泼、具有生机活力。关节动画是实现这类角色动画不可缺少的部分。

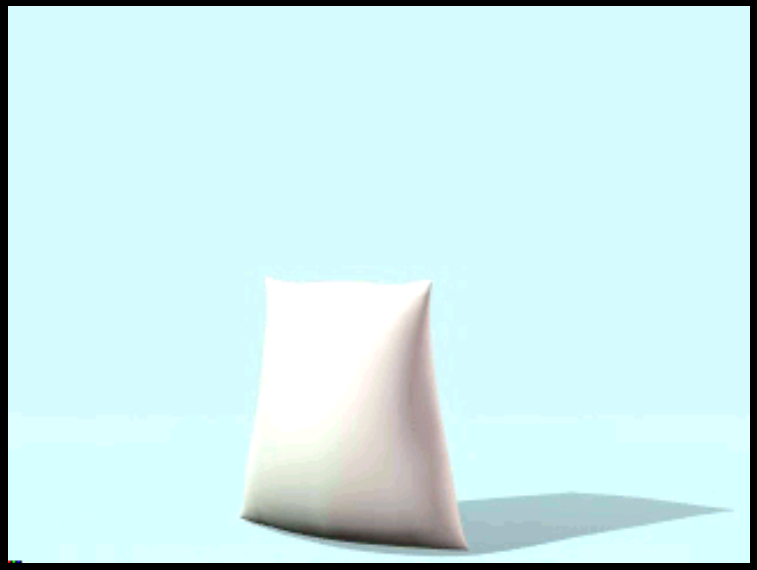


可用于游戏...

也可用于无生命的物体以创建拟人效果







DEMO

电影

《恐龙》剧照

《泰坦尼克号》剧照

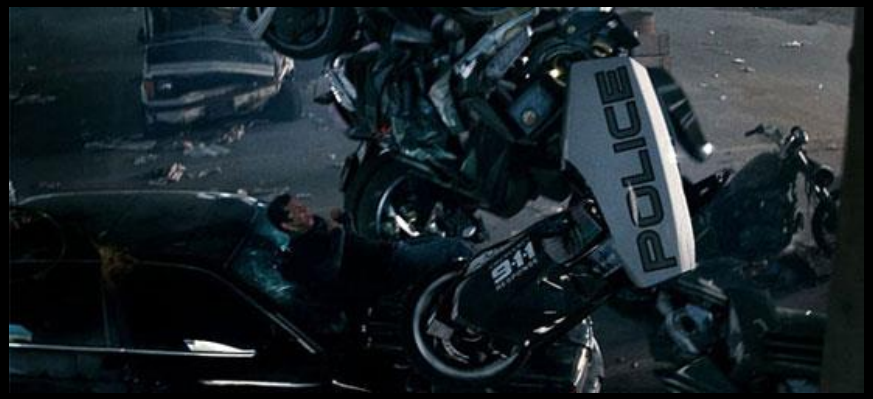
《海底总动员》剧照

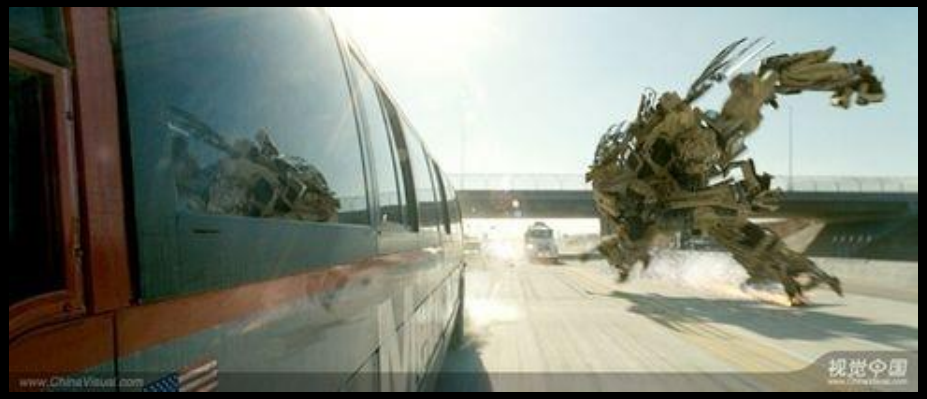
《金刚》剧照

《精灵鼠小弟2》剧照

《闪电狗》剧照

《变形金刚》剧照





《变形金刚》剧照

《史前一万年》剧照

《史前一万年》剧照

《纳尼亚传奇》剧照

《加勒比海盗》剧照

《加菲猫2之双猫记》剧照

《蜘蛛侠3》剧照

《星球大战前传3：西斯的反击》剧照

《哈利波特3》剧照

《冰川时代2》剧照

《尼斯湖怪：深水传说》剧照

《尼斯湖怪：深水传说》剧照

《极地快车》预告片


优酷
THE FOLLOWING PREVIEW HAS BEEN APPROVED FOR
ALL AUDIENCES
BY THE MOTION PICTURE ASSOCIATION OF AMERICA

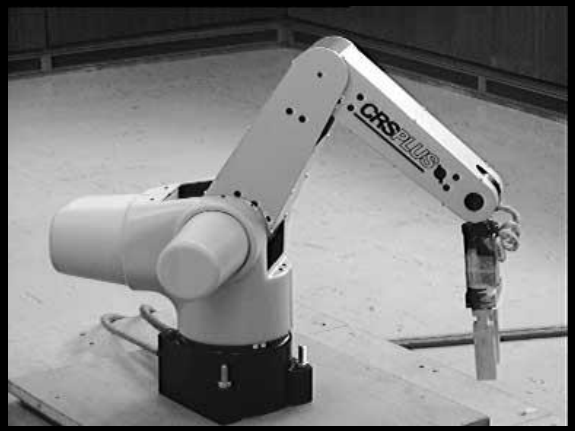
模型层次 vs. 运动层次

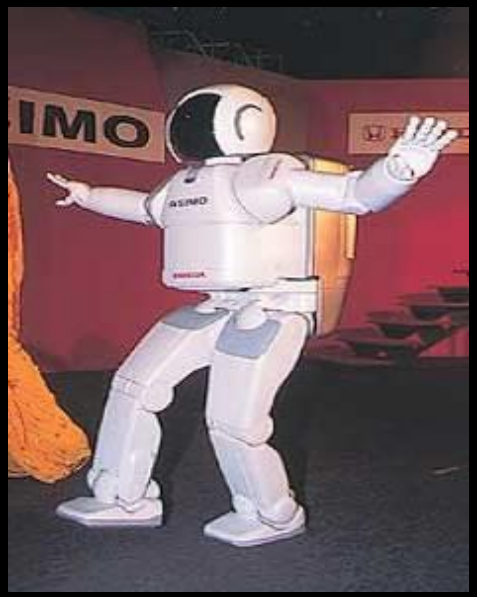
- 通常用相对运动
  - 一个物体相对于另外一个物体运动
- 物体层次 + 相对运动 构成 运动层次
  - 连杆
  - 运动通常受限制

运动学

- 如何通过设置位置随时间的参数来对连杆设置动画？

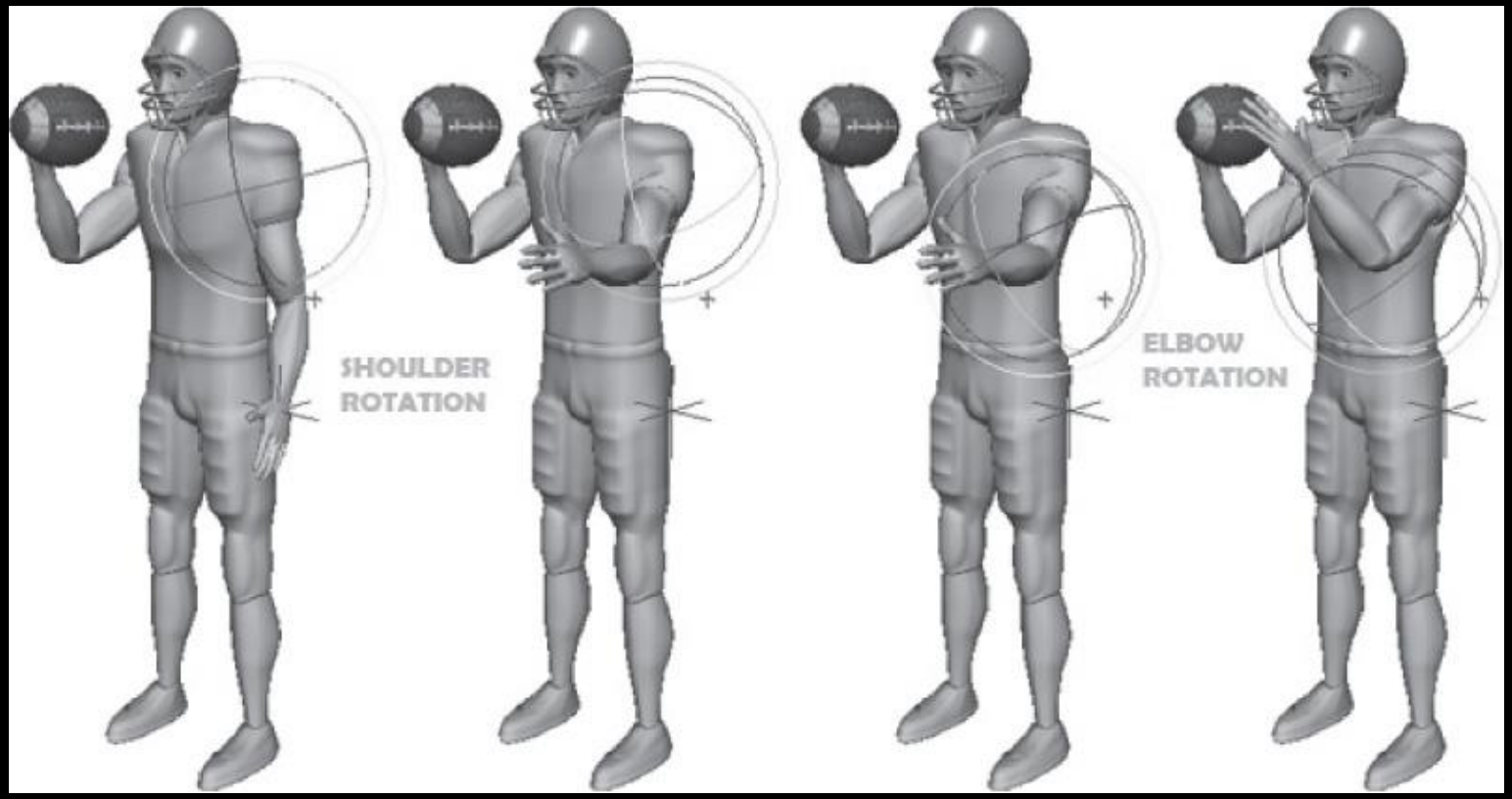
- 运动学不考虑引起运动的力（区别于动力学）





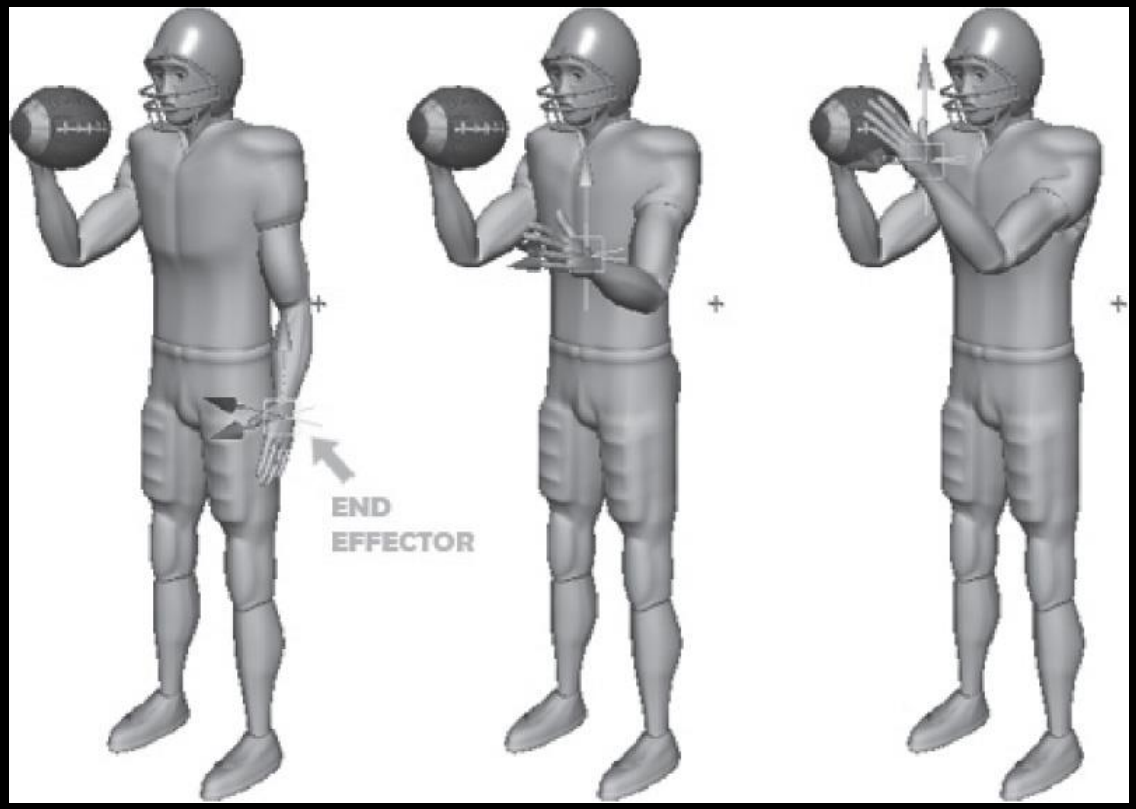
正向运动学(Forward Kinematics)

- 动画师通过直接指定关节处的关节运动参数来控制物体的运动



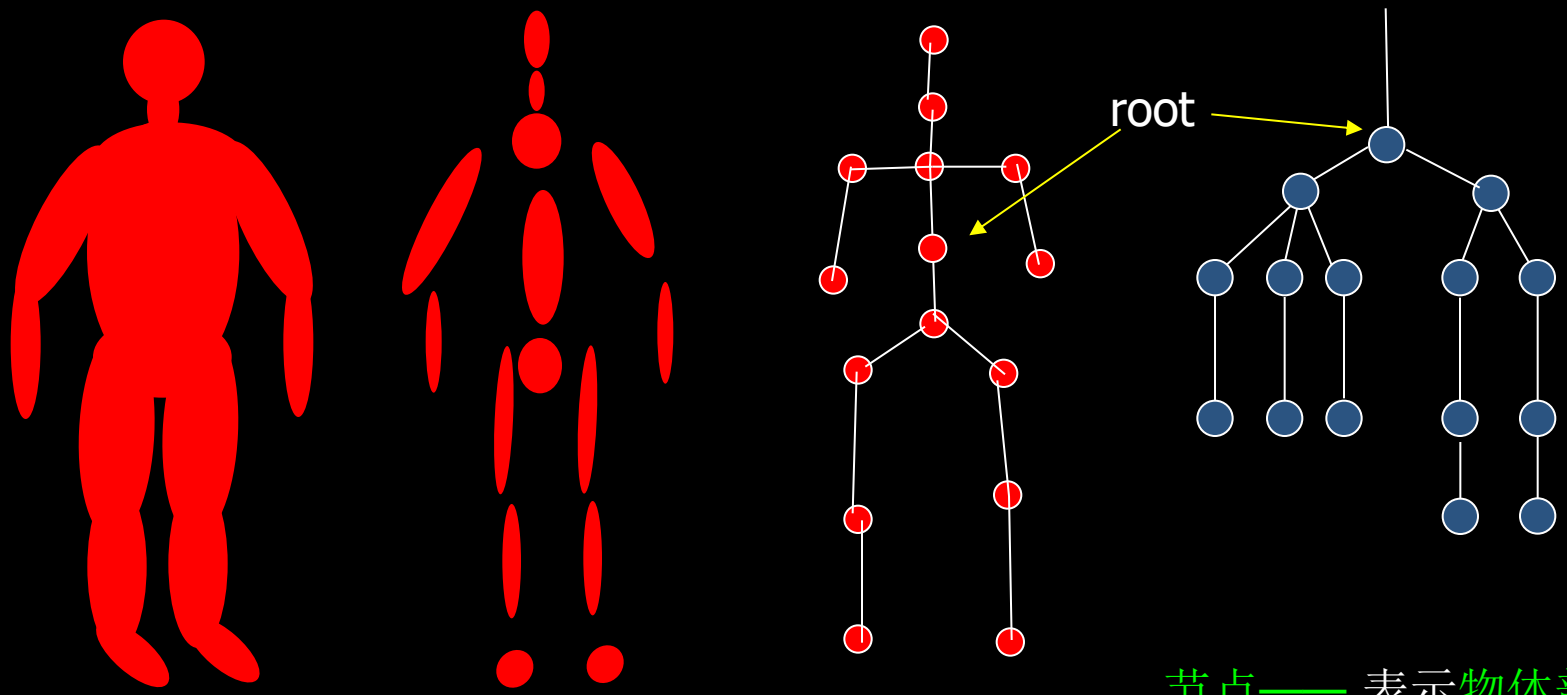
逆向运动学（Inverse Kinematics, IK）

- 动画师指定目标位置，系统求解满足要求的关节角。



关节模型

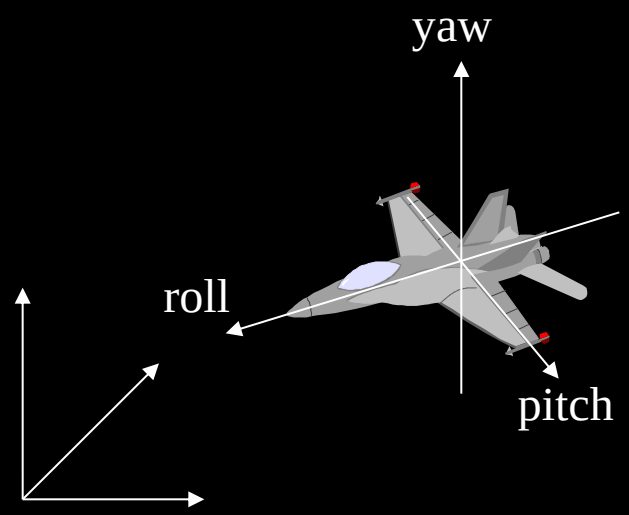
- 把关节角色表示为一系列通过关节(joints)相连接的连杆(links)



节点—— 表示物体部件

自由度(Degrees of Freedom, DOF)

- 完全指定一个物体运动所需的最小坐标数目

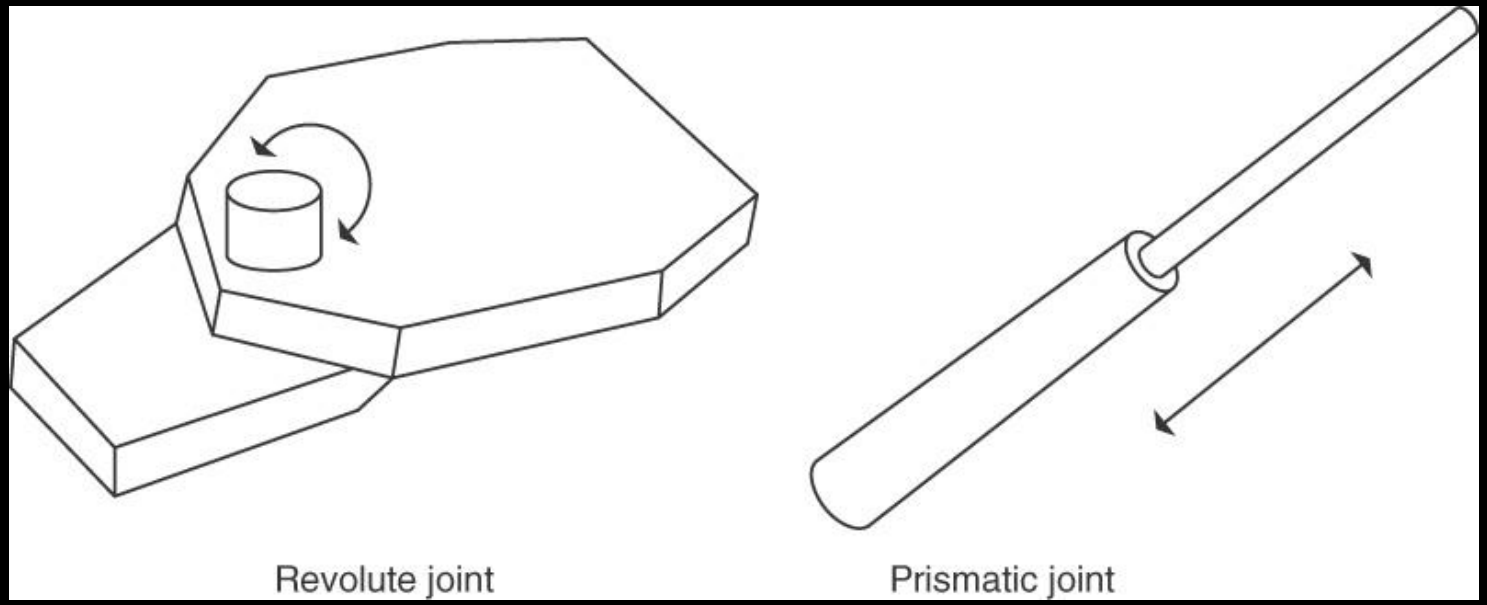


6 DOF: x, y, z, raw, pitch, yaw

单个自由度关节

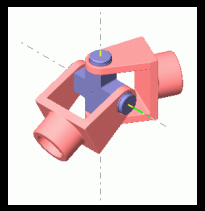
- 允许在一个方向运动

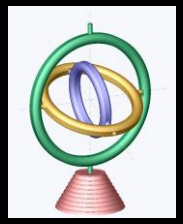
  例如: 转动关节, 移动关节(Prismatic joint)



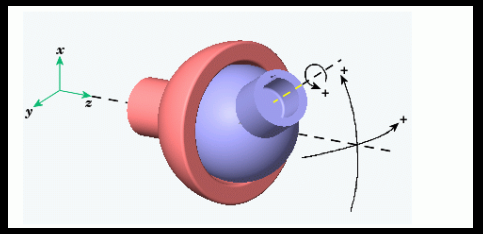
复杂关节

- 复杂关节
  - 2自由度关节
  - 3自由度关节
    - 万向节(Gimbal)
    - 球状关节(如肩、膝关节) (Ball and socket joint)

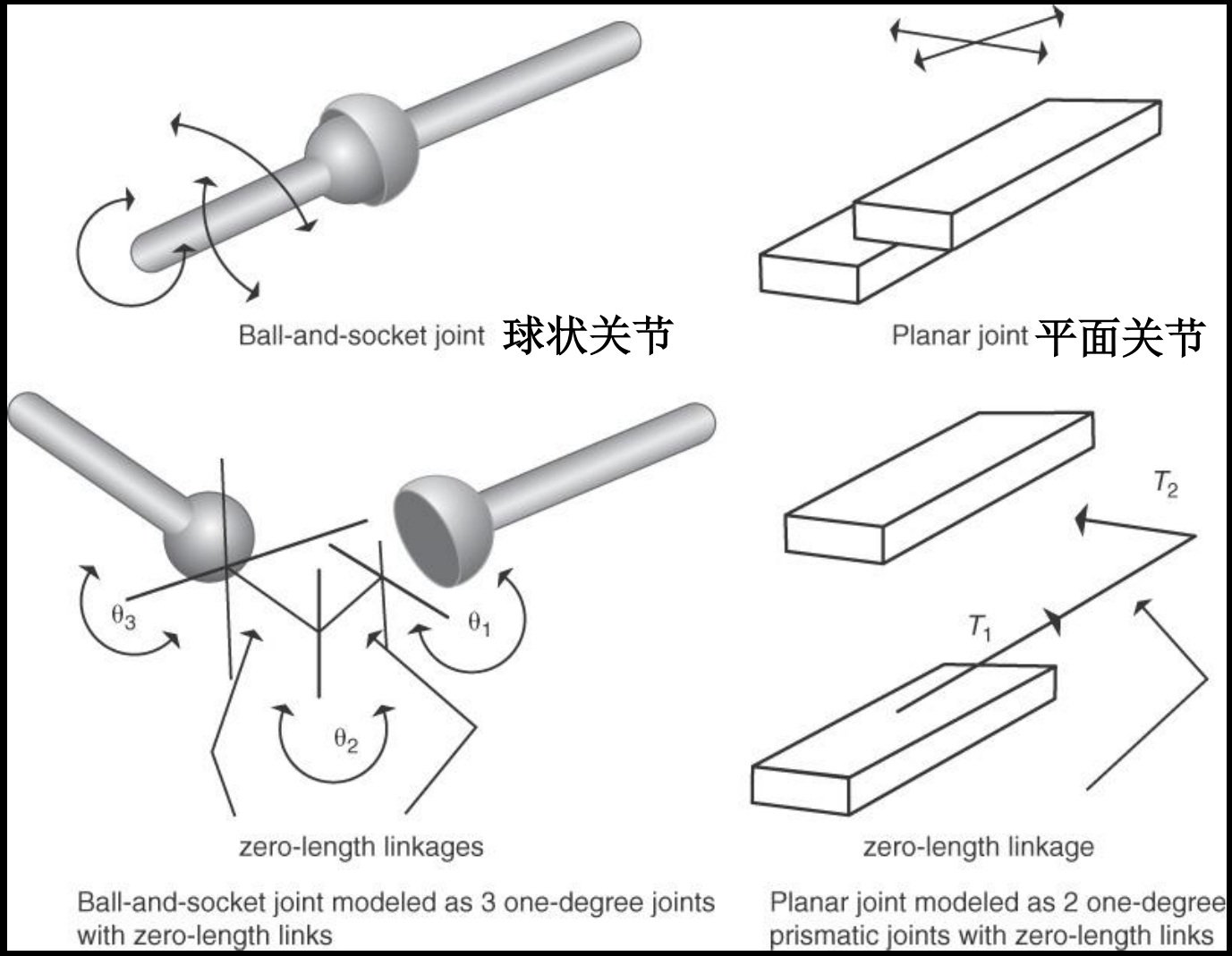




- *n*个自由度的复杂关节可看成由*n*个自由度为1的关节通过 *n*-1 个长度为0的连杆相连。



复杂关节


Ball-and-socket joint 球状关节
Planar joint 平面关节
θ3
θ1
θ2
zero-length linkages
Ball-and-socket joint modeled as 3 one-degree joints with zero-length links
T2
T1
zero-length linkage
Planar joint modeled as 2 one-degree prismatic joints with zero-length links

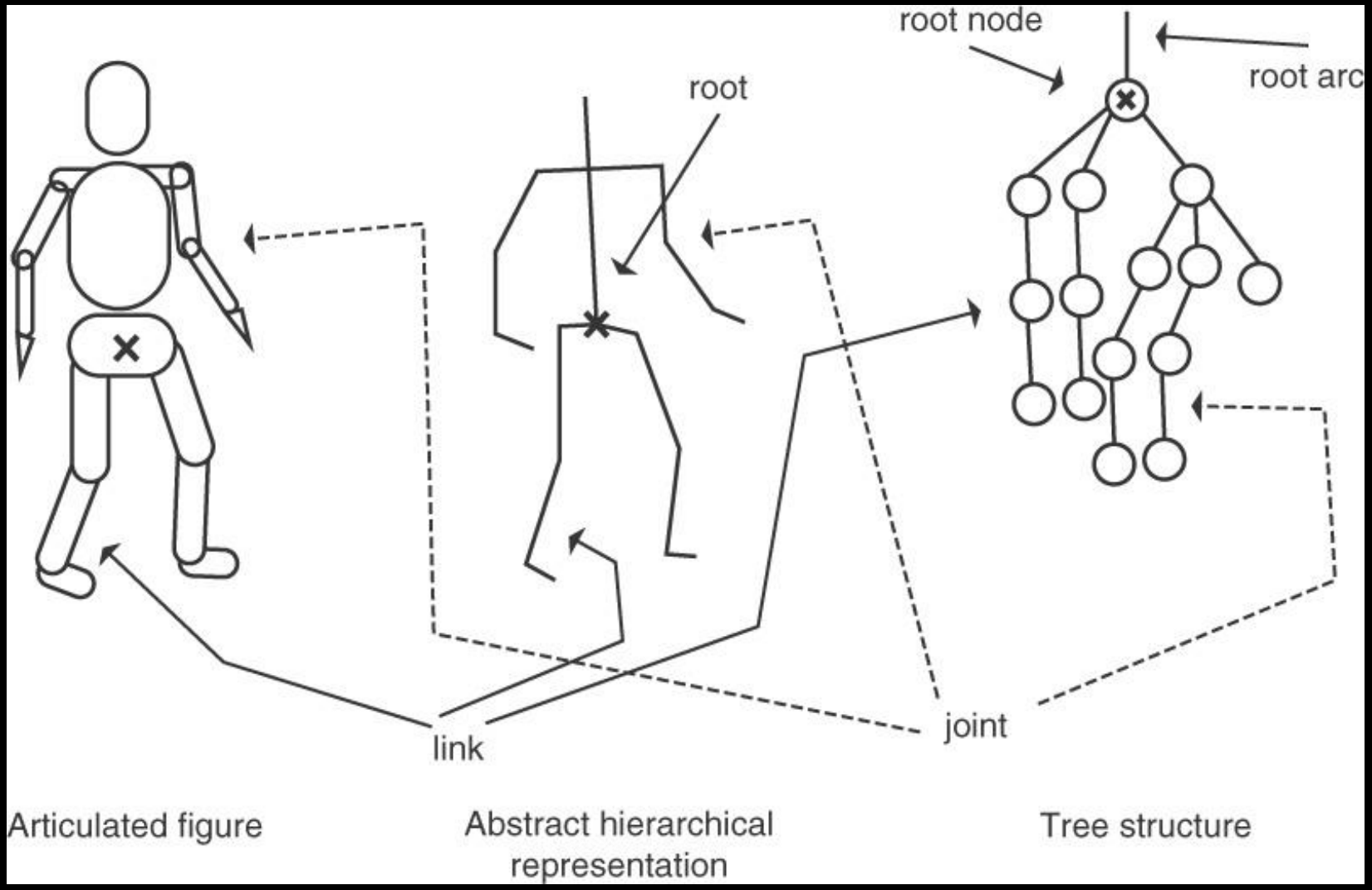
人体模型的自由度

- 根节点(Root): 3 translational DOF + 3 rotational DOF
- 人体模型通常用旋转关节(Rotational joints)
- 每个关节至多有3个自由度
  - 肩关节(Shoulder): 3 DOF
  - 腕关节(Wrist): 2 DOF
  - 膝关节(Knee): 1 DOF

层次模型的数据结构

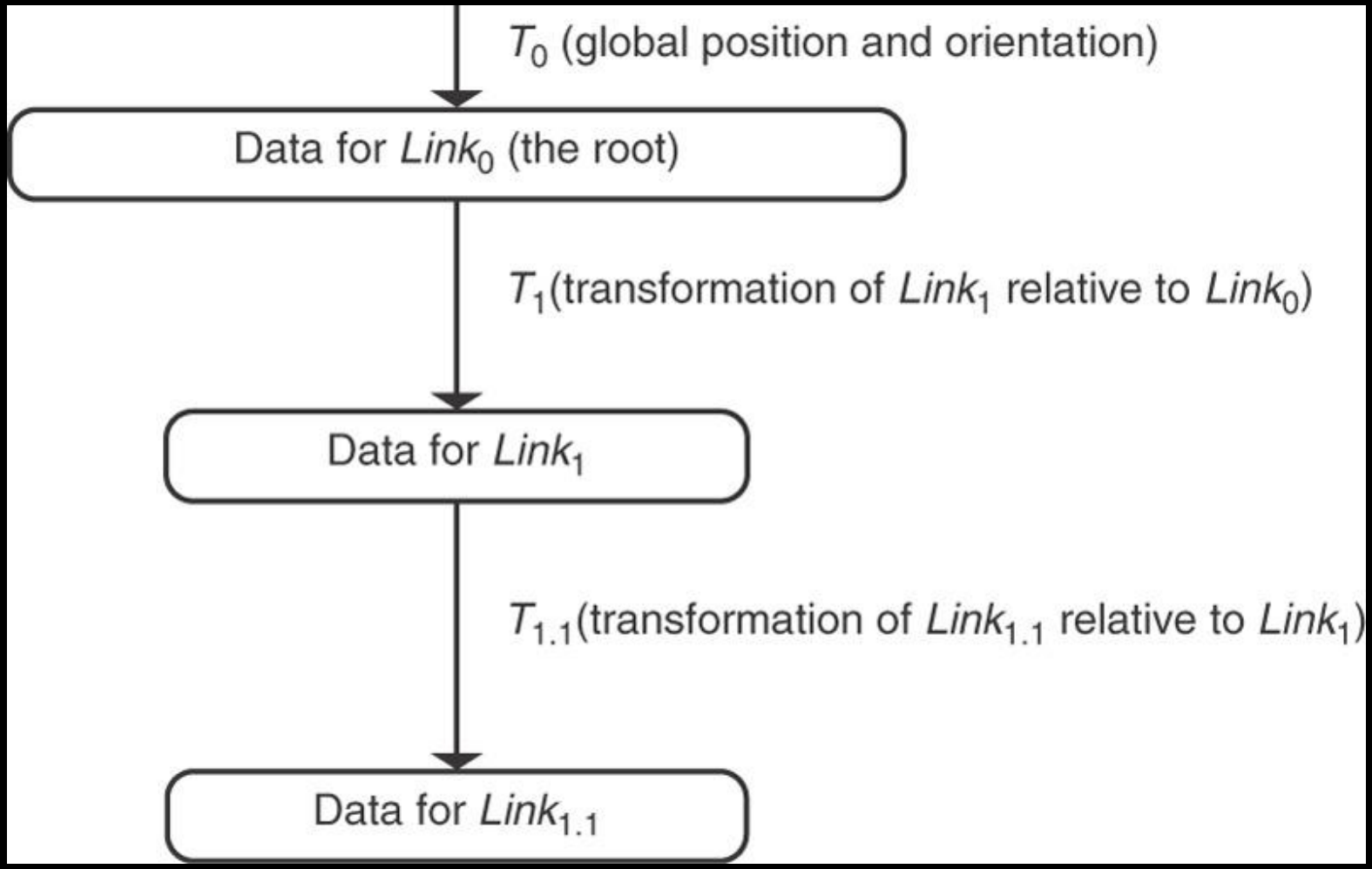
- 层次连杆可用一树状结构来表示
  - 根节点——对应于物体的根部，其位置在世界坐标系中给出
  - 其它节点——相对于根节点来表示
  - 叶节点
- 层次结构与树之间的映射
  - 节点—— 表示物体部件
  - 连接弧——表示层次结构中应用于物体部件之间的关节或变换

层次模型的数据结构


root node
root arc
root
joint
link
Articulated figure
Abstract hierarchical representation
Tree structure

层次模型的树状结构

T_0： $Link_0$到世界坐标系位置和方向的变换



层次模型的变换（平移）

V_0: Link_0的一个顶点

它的世界坐标为: $V_0' = T_0 V_0$

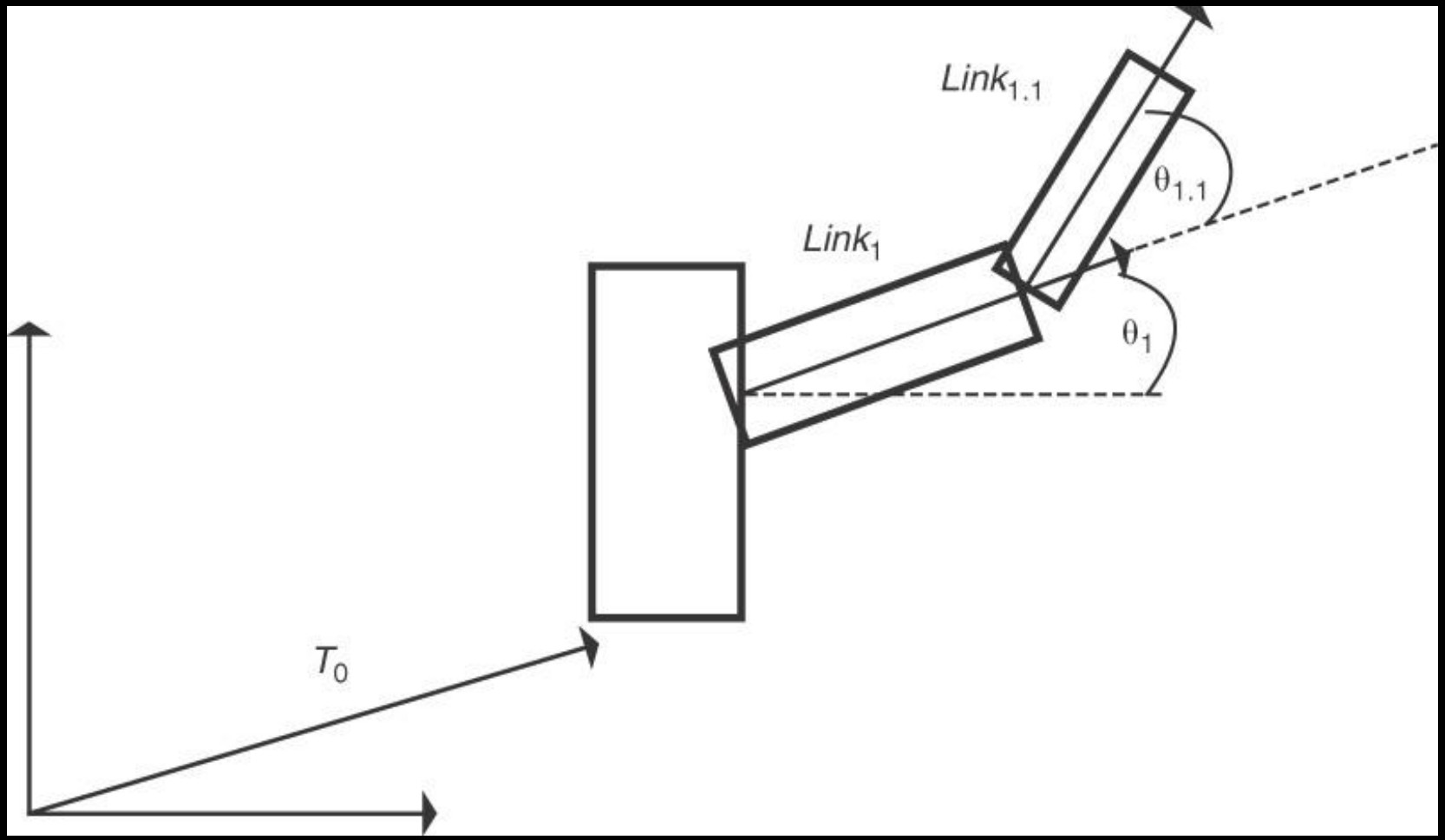
V_1: Link_1的一个顶点

它的世界坐标为: $V_1' = T_0 T_1 V_1$

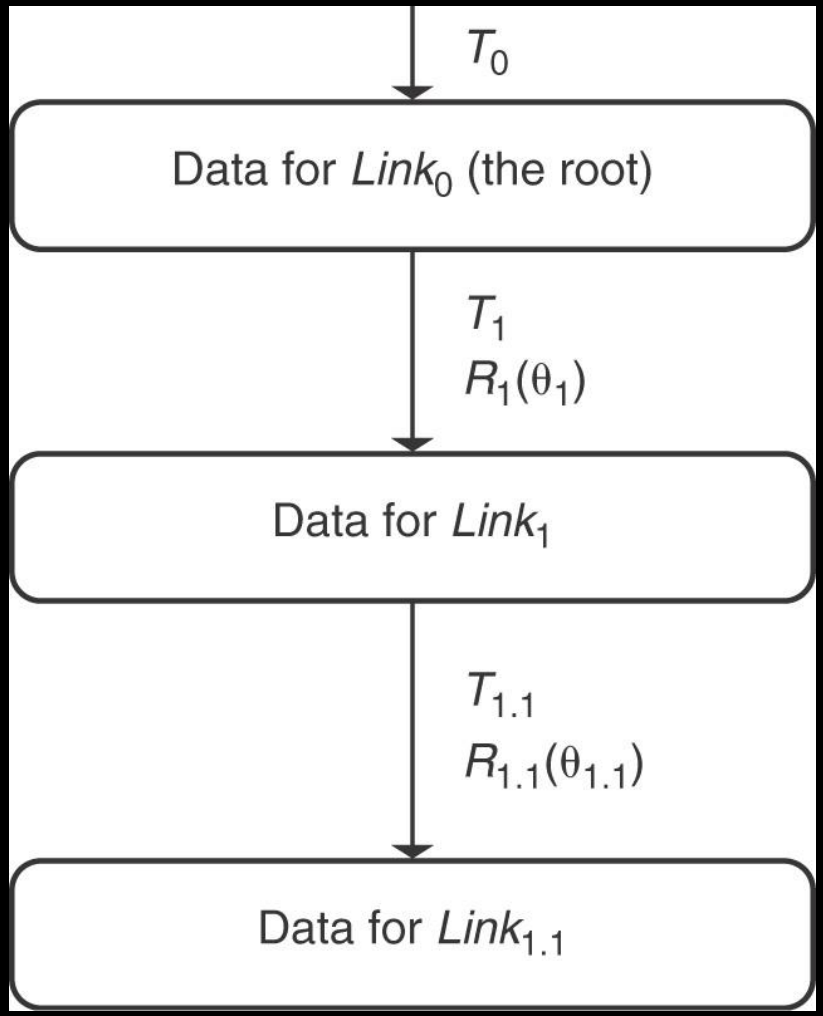
$V_{1,1}$: $\text{Link}_{1,1}$的一个顶点

它的世界坐标为: $V_{1,1}' = T_0 T_1 T_{1,1} V_{1,1}$

层次模型关节处的可变旋转



层次模型的树状结构


T_0
Data for $Link_0$ (the root)
T_1
$R_1(\theta_1)$
Data for $Link_1$
$T_{1.1}$
$R_{1.1}(\theta_{1.1})$
Data for $Link_{1.1}$

层次模型的变换（平移和旋转）

V_0: Link_0的一个顶点

它的世界坐标为: $V_0' = T_0 V_0$

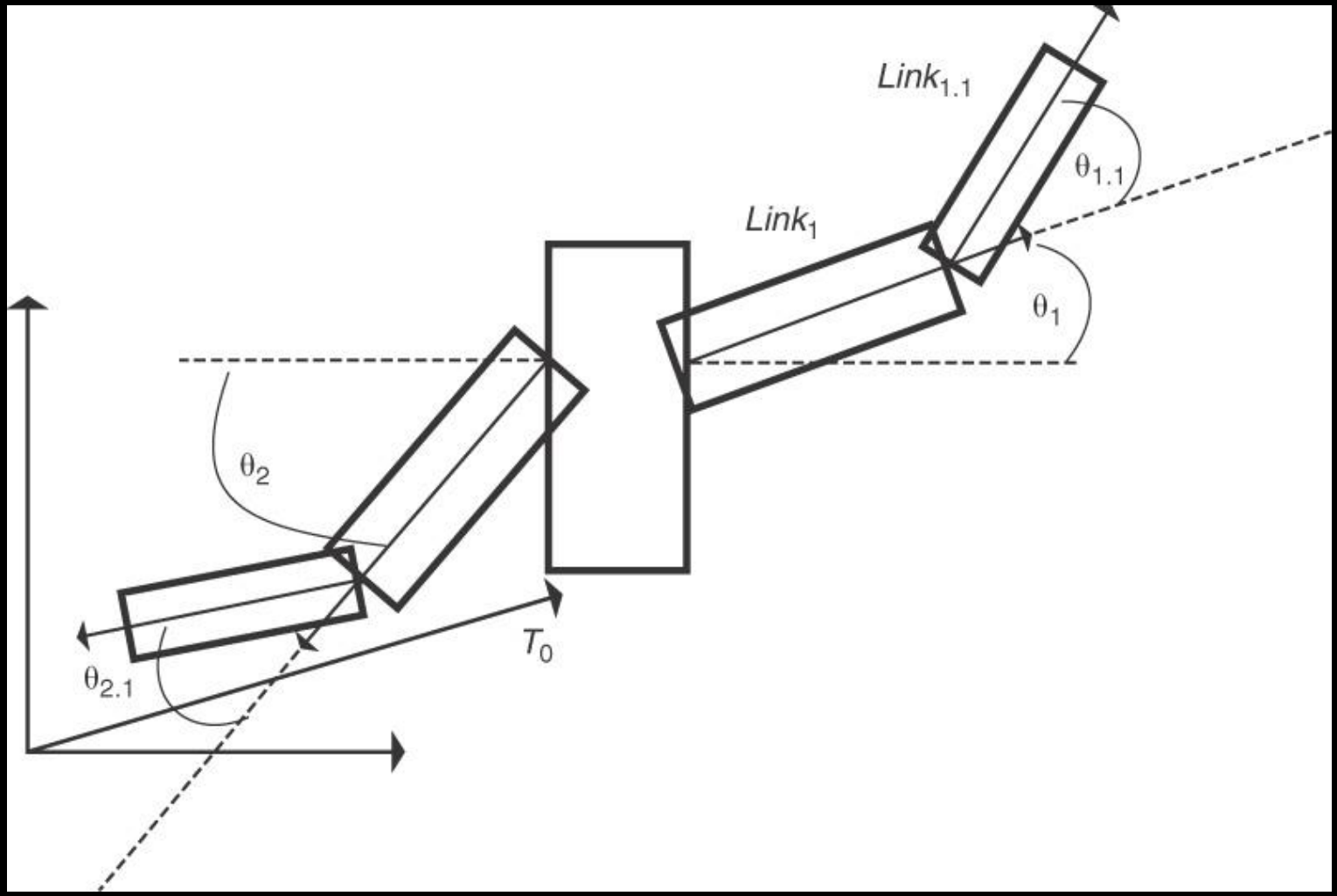
V_1: Link_1的一个顶点

它的世界坐标为: $V_1' = T_0 T_1 R_1(\theta_1) V_1$

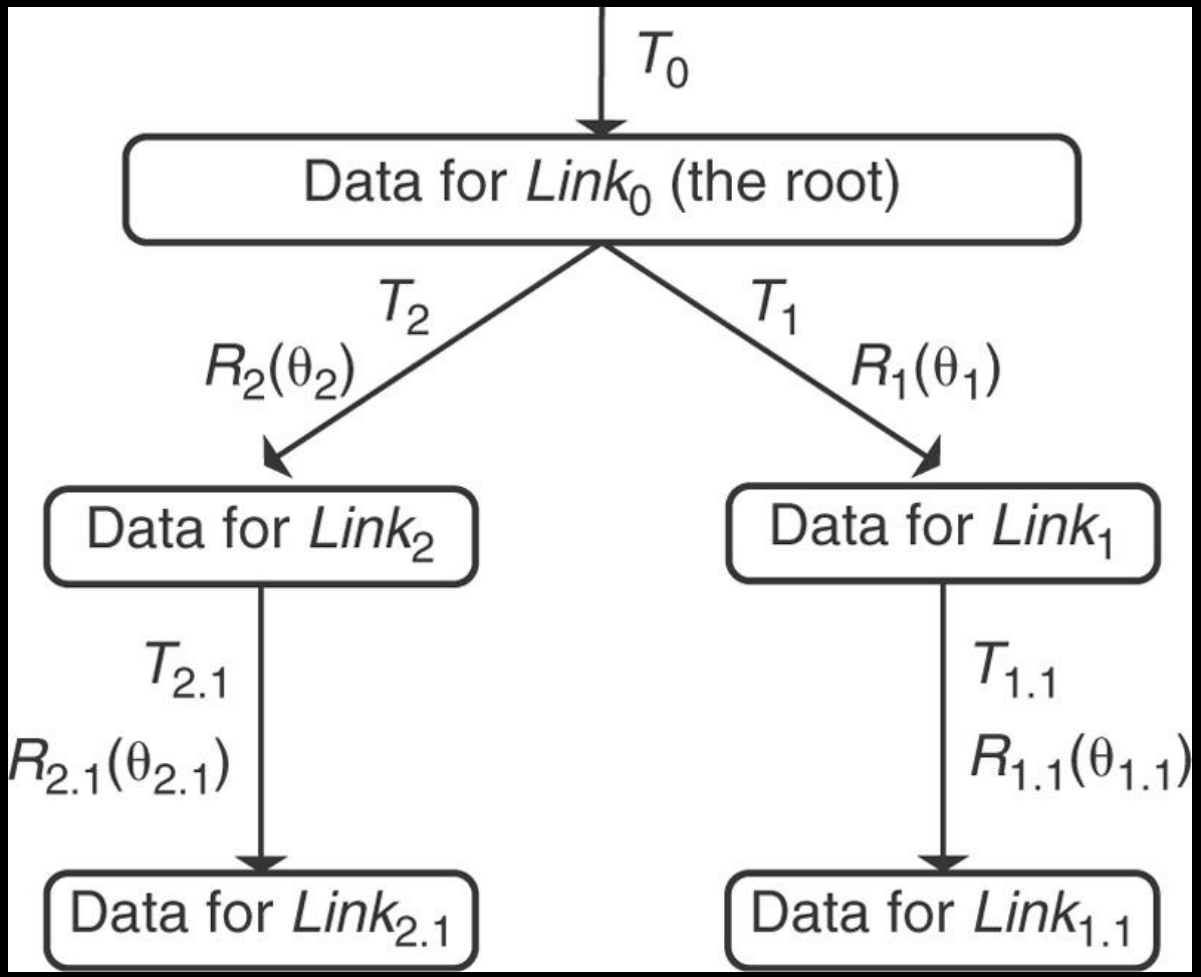
$V_{1,1}$: $\text{Link}_{1,1}$的一个顶点

它的世界坐标为: $V_{1,1}' = T_0 T_1 R_1(\theta_1) T_{1,1} R_{1,1}(\theta_{1,1}) V_{1,1}$

双肢体(Two Appendages)层次模型



双肢体(Two Appendages)层次模型

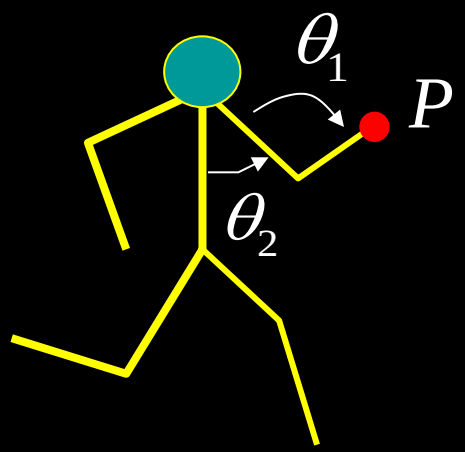


关节空间 vs. 笛卡尔空间

- 关节空间
  - 由关节角形成的空间
  - 可对所有关节进行细微控制
- 笛卡尔空间
  - 三维空间
  - 指定与环境的相互作用

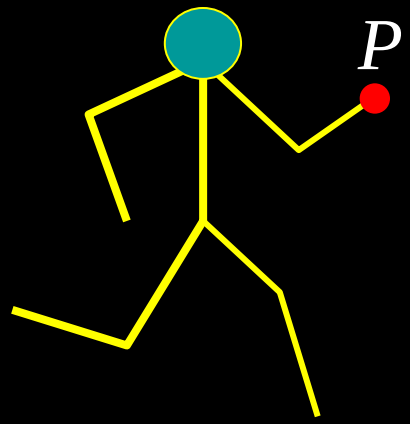
正向和逆向运动学

- 正向运动学(Forward kinematics)
  - 从关节空间映射到笛卡尔空间
- 逆向运动学(Inverse kinematics)
  - 从笛卡尔空间映射到关节空间



Forward Kinematics

$$P = f(\theta_1, \theta_2)$$

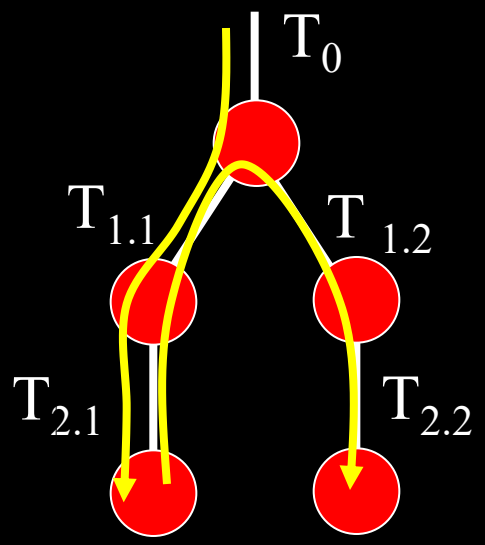


Inverse Kinematics

正向运动学

- 计算整棵树
  - 从根节点到叶节点进行深度优先遍历
  - 重复以下步骤，直到所有节点和连接弧都被访问过
    - 对树进行回溯，直到遇到一个未被访问过的向下连接弧
    - 对向下连接弧进行遍历
  - 在实现阶段…
    - 需要堆栈存贮变换

正向运动学——树遍历



$M = I$

$M = T_0$

$M = T_0 * T_{1.1}$

$M = T_0 * T_{1.1} * T_{2.1}$

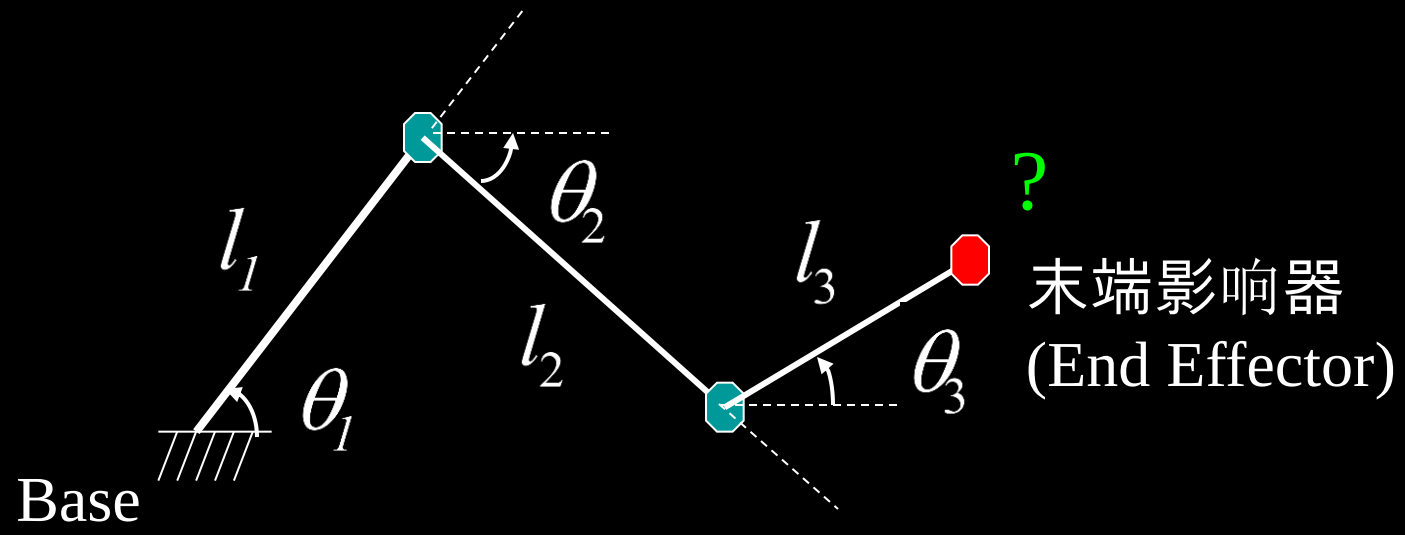
$M = T_0 * T_{1.1}$

$M = T_0$

$M = T_0 * T_{1.2}$

$M = T_0 * T_{1.2} * T_{2.2}$

正向运动学例子



$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3)$$

$$y = l_1 \sin(\theta_1) - l_2 \sin(\theta_2) + l_3 \sin(\theta_3)$$

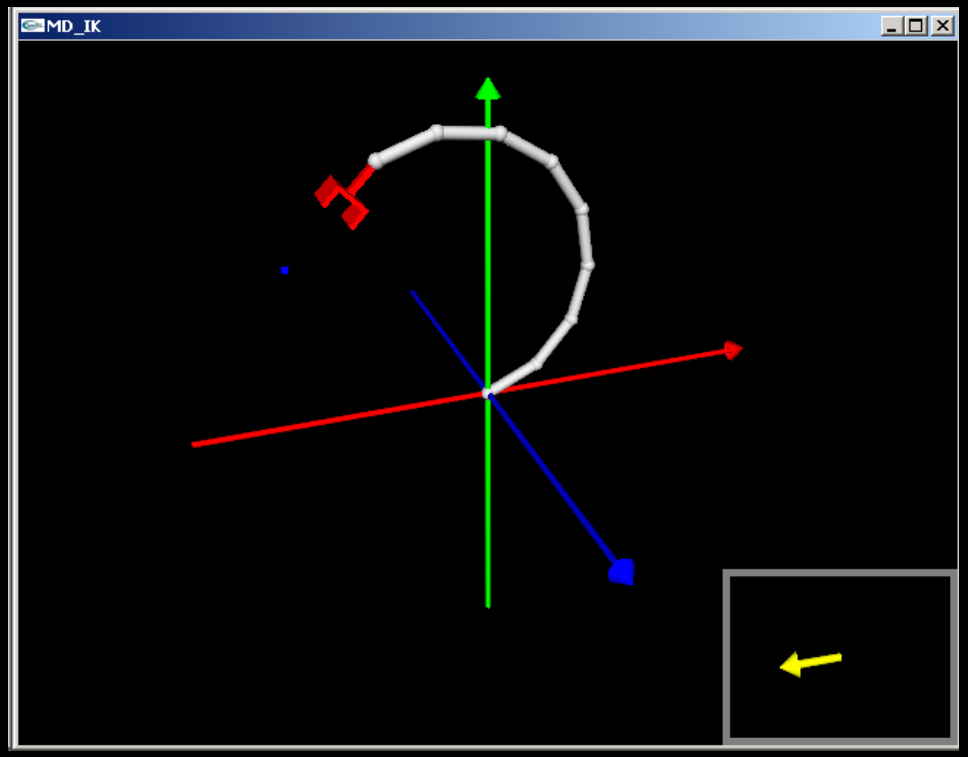
DEMO

MOCAP Animation

逆向运动学(Inverse Kinematics)

- 给定初始姿态向量和目标姿态向量，计算关节向量的值，使得物体满足所需的姿势
  - 可以有0个、1个或多个解
    - 过约束系统(Over-constrained system)
    - 欠约束系统(Under-constrained system)
  - 奇异问题

IK Demo



- **Inverse and Forward kinematics demo application**

http://www.fit.vutbr.cz/~dobsik/projects/kinem_INV/kinem_INV.html

逆向运动学

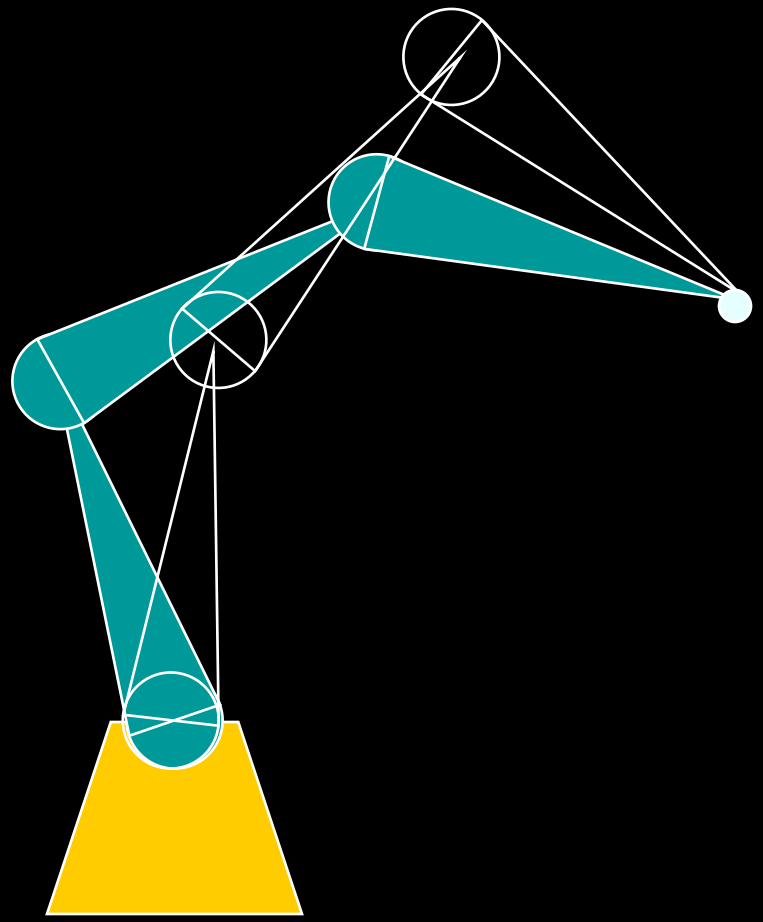
- 例子
  - 2个方程 (约束条件)
  - 3个未知数

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3)$$
$$y = l_1 \sin(\theta_1) - l_2 \sin(\theta_2) + l_3 \sin(\theta_3)$$

  - 有无穷多个解!
- 这种情况常常遇到!
  - 在保持你的手指碰到鼻子时，可以移动你的肘

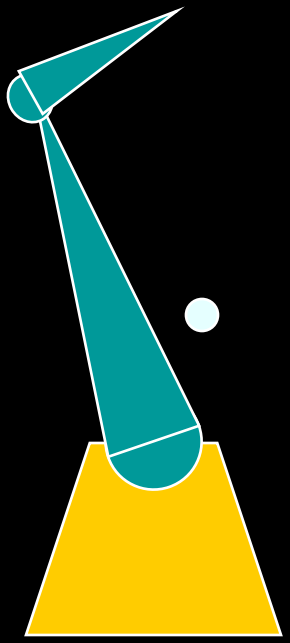
IK中的其它问题

- 有无穷多解！



IK中的其它问题

- 无解



逆向运动学

- 一旦得到关节向量值后，我们可对角色的初始姿势和最终姿势的关节向量值进行插值，从而得到角色的动画。
  - 缺点：如果初始和最终姿势向量有很大差异时，动画效果可能并不好。解决办法：
    - 插值姿态向量
    - 对每个得到的插值姿态向量进行IK计算，得到中间某一时刻的关节向量值

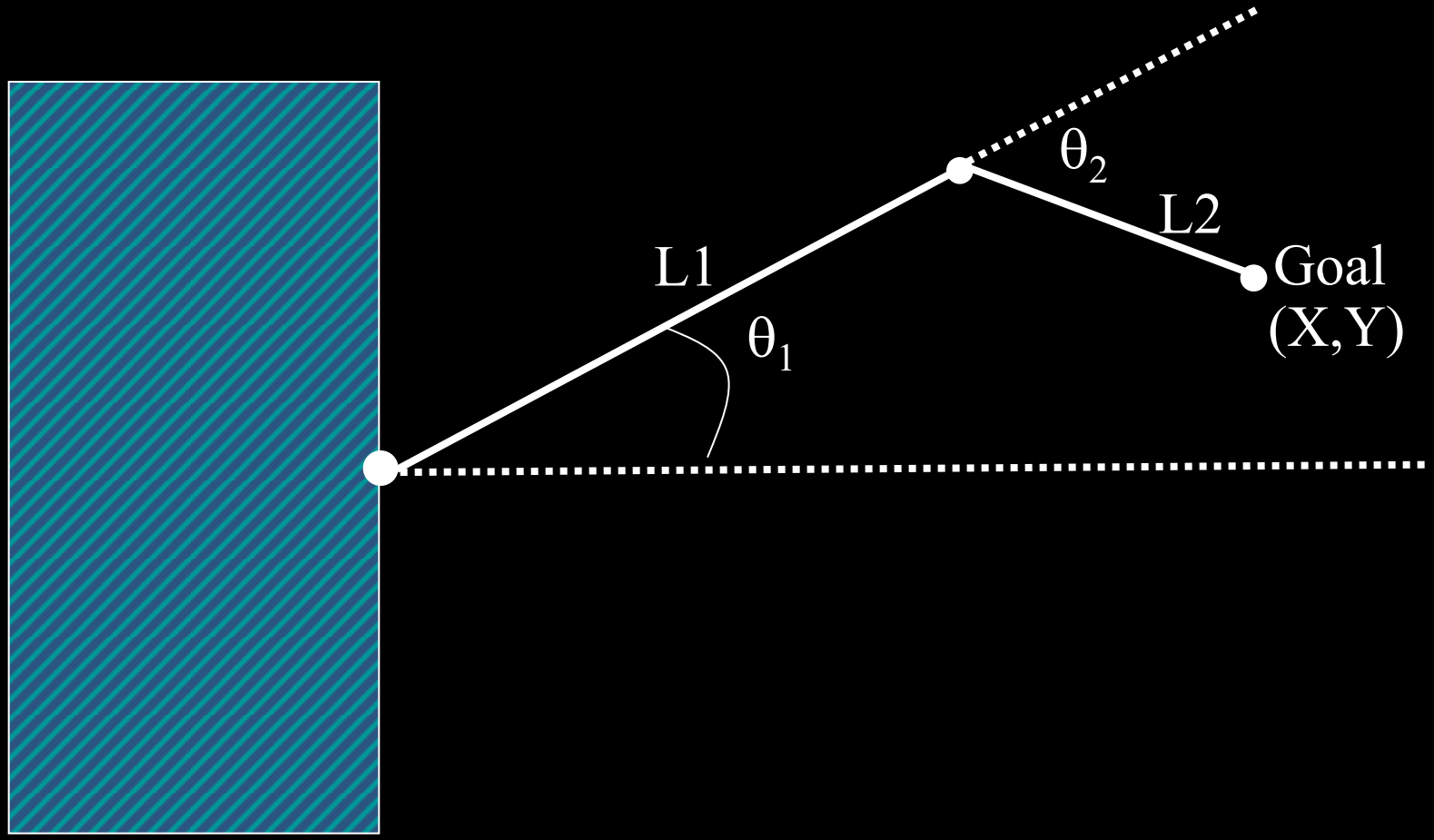
逆向运动学

解析法 vs. 数值计算

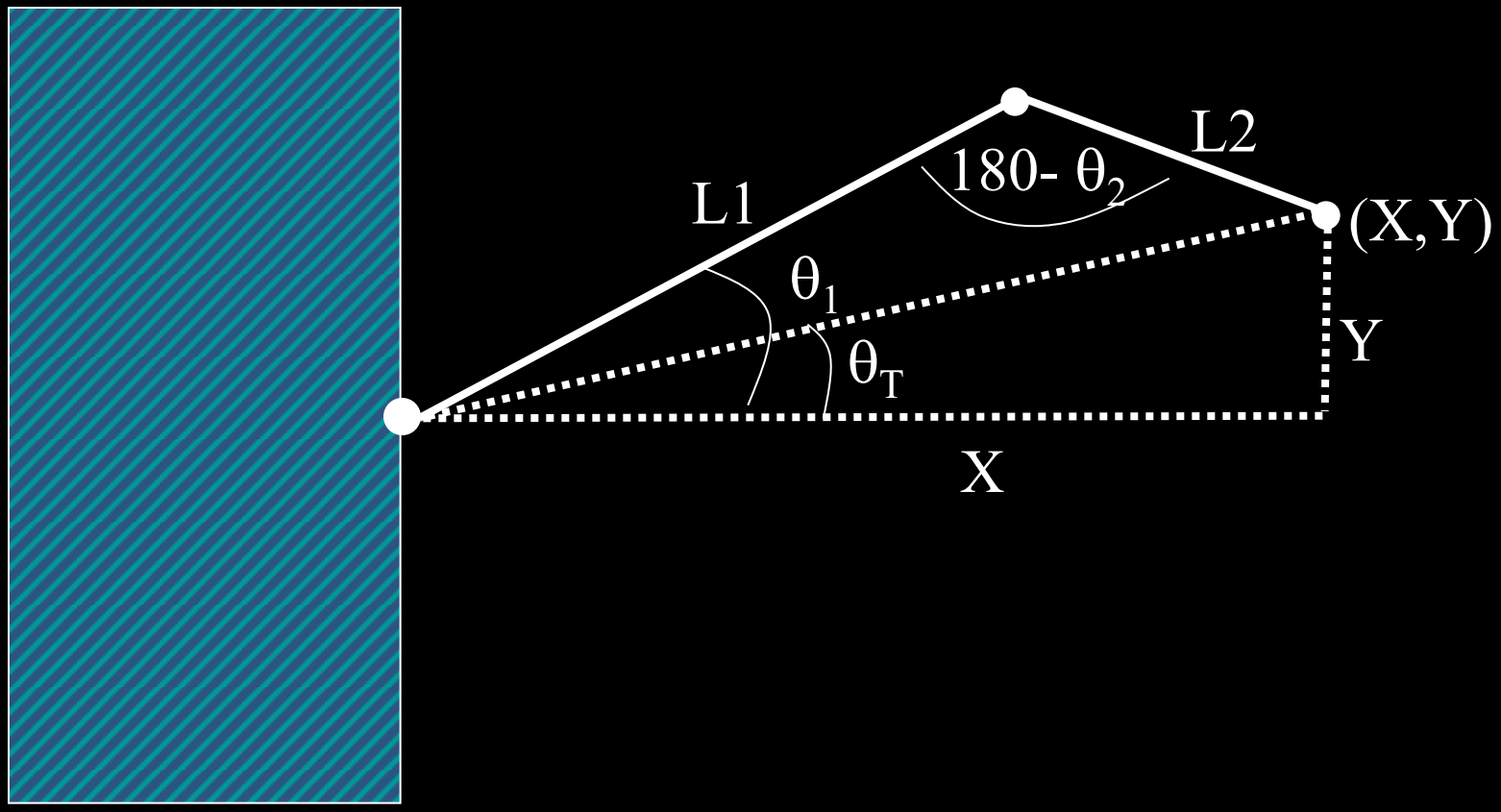
- 只有当关节链非常简单时，才存在解析解

- 一般可用数值增量法求解
  - 逆向雅可比方法(Inverse Jacobian method)
  - 其它方法

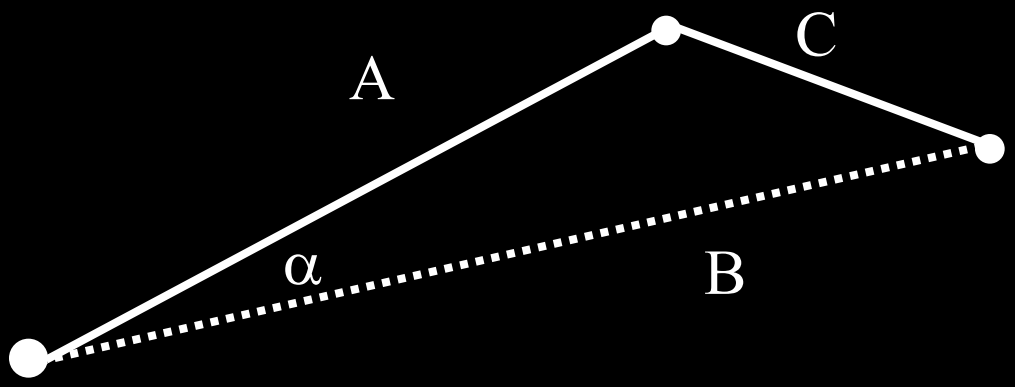
逆向运动学——解析求解法



逆向运动学——解析求解法

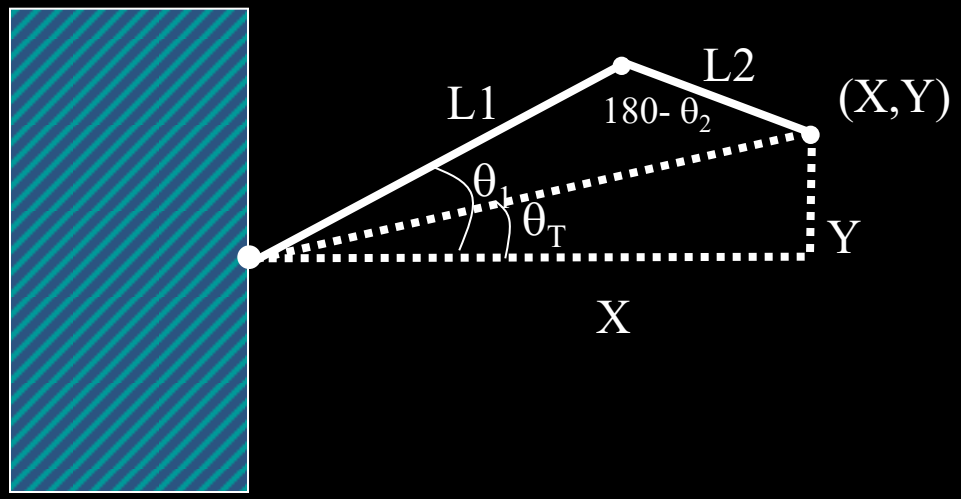


逆向运动学——解析求解法



$$\cos(\alpha) = \frac{A^2 + B^2 - C^2}{2AB}$$

逆向运动学——解析求解法



$$\cos(\theta_T) = \frac{X}{\sqrt{X^2 + Y^2}}$$

$$\theta_T = \cos^{-1}\left(\frac{X}{\sqrt{X^2 + Y^2}}\right)$$

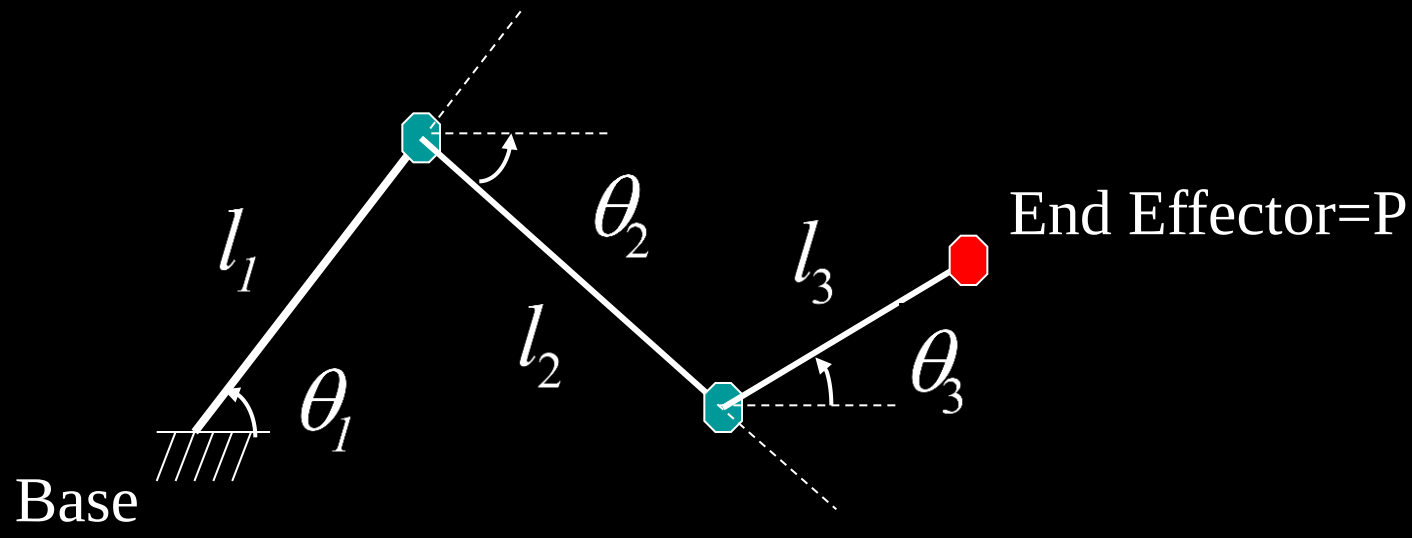
$$\cos(180 - \theta_2) = \frac{L_1^2 + L_2^2 - \left(X^2 + Y^2\right)}{2L_1L_2}$$

$$\cos(\theta_1 - \theta_T) = \frac{L_1^2 + X^2 + Y^2 - L_2^2}{2L_1\sqrt{X^2 + Y^2}}$$

$$\theta_2 = 180 - \cos^{-1}\left(\frac{L_1^2 + L_2^2 - \left(X^2 + Y^2\right)}{2L_1L_2}\right)$$

$$\theta_1 = \cos^{-1}\left(\frac{L_1^2 + X^2 + Y^2 - L_2^2}{2L_1\sqrt{X^2 + Y^2}}\right) + \theta_T$$

逆向运动学——更复杂的关节



$$\theta_1, \theta_2, \theta_3 = \mathrm{f}^{-1}(P)$$

为什么求解IK是困难的?

- 冗余(Redundancy)
- 要求自然的运动控制
  - 关节限制
  - 最小的抖动(minimum jerk)
  - 运动方式
- 奇异问题(Singularities)
  - 病态方程(ill-conditioned)
  - 奇异方程Singular

数值求解IK

- 逆向雅克比方法(Inverse-Jacobian method)
- 循环坐标下降(Cyclic Coordinate Descent (CCD))法
- 基于优化的方法(Optimization-based method)
- 基于样例的方法(Example-based method)

逆向雅克比方法

$P(t) = f(\theta(t))$

$P \in R^n$ (通常$n = 6$)

$\theta \in R^m$ (m = 自由度)

- 雅克比矩阵为$n \times m$的矩阵，它把θ的微分($d\theta$)与P的微分相关联 (dP)

$$\frac{dP}{dt} = J(\theta)\frac{d\theta}{dt} \qquad \text{其中} J_{ij} = \frac{\partial f_i}{\partial \theta_j}$$

  - 所以雅克比矩阵把关节空间的速度映射到笛卡尔空间的速度 $V = J(\theta)\dot{\theta}$

逆向雅克比方法

- 逆向雅克比问题为：

$$\theta = f^{-1}(P)$$

  - f 是一个高度非线性函数
- 通过把雅克比矩阵求逆，把该问题在当前位置局部线性化 $\dot{\theta} = J^{-1}V$

  - 通过一系列增量步骤，迭代到所需要的位置

逆向雅克比方法

- 给定初始姿势和所需要的姿势，迭代变化关节角，使得末端影响器朝目标位置和方向前进
  - 对于中间帧，插值得到所需的姿态向量
    - 对于每一步k, 通过上述公式得到关节的角速度 $\dot{\theta}$，然后执行

$$\theta_{k+1} = \theta_k + \Delta t \dot{\theta}$$

逆向雅克比方法——迭代法

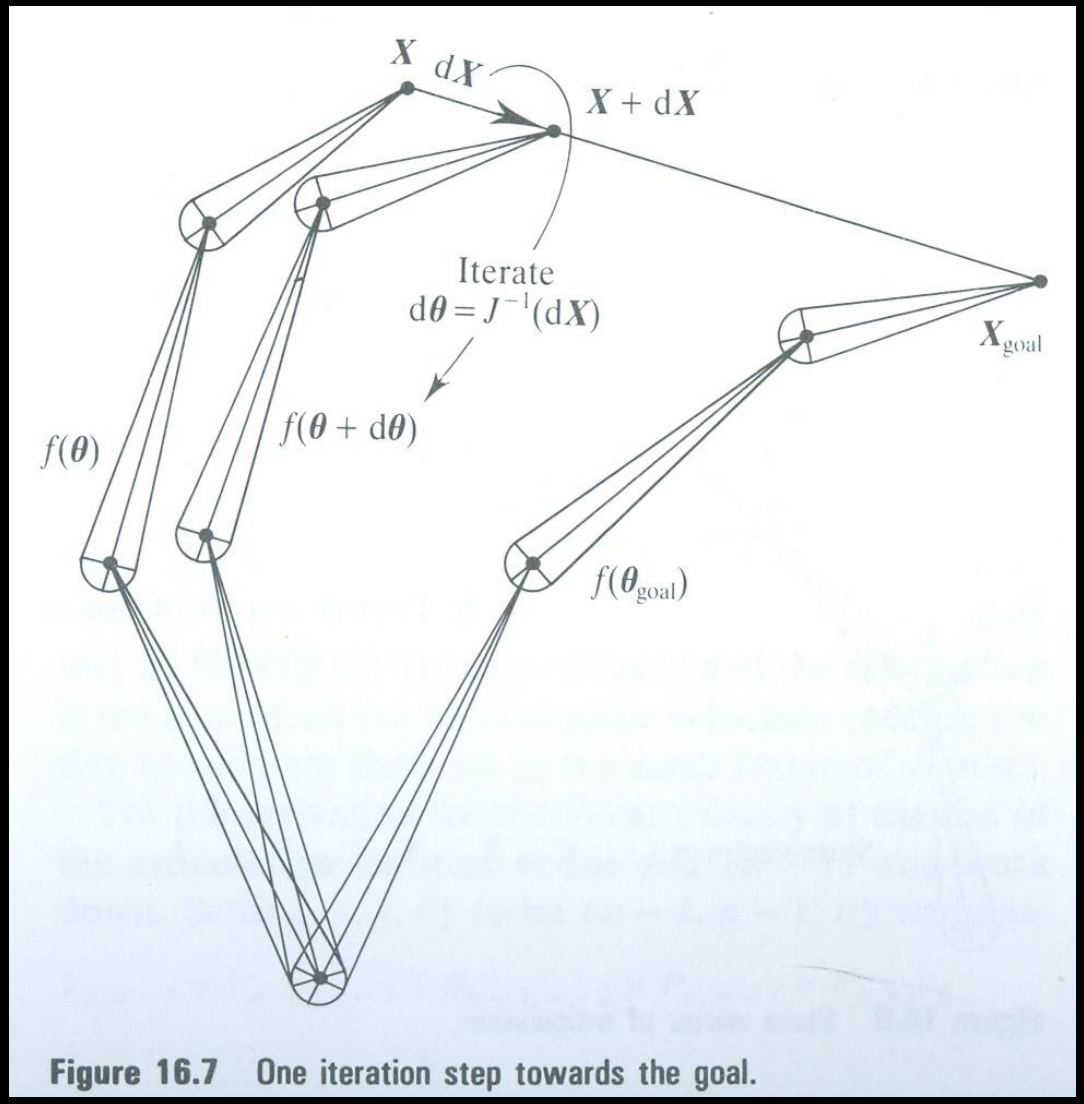


Figure 16.7 One iteration step towards the goal.

逆向雅克比方法——迭代法

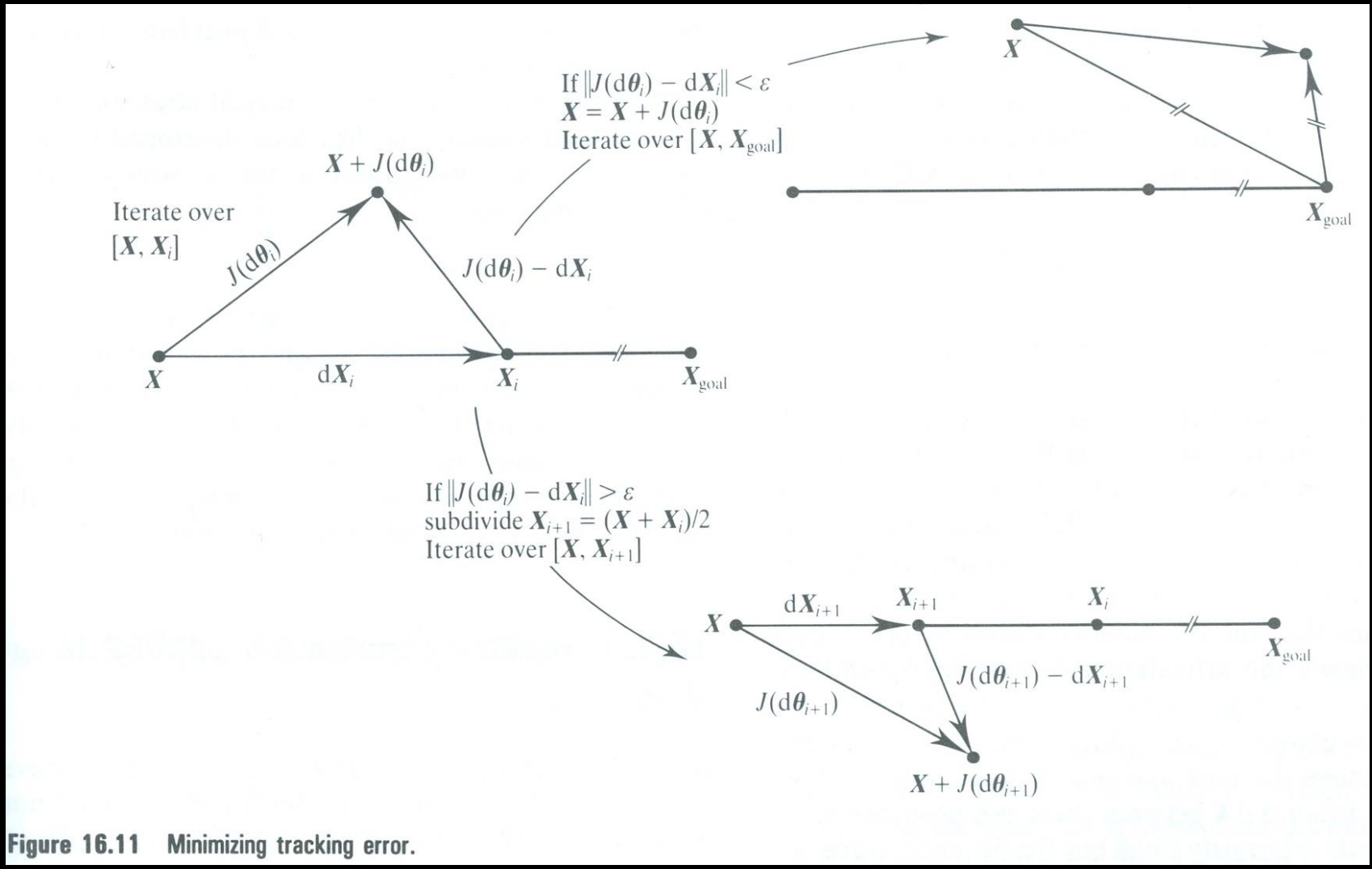


Figure 16.11 Minimizing tracking error.

逆向雅克比方法——DEMO

逆向雅克比方法

- 雅克比矩阵把关节空间的速度映射到笛卡尔空间的速度。

$$X(t) = f(\theta(t)), \qquad V = J\dot{\theta}$$

- 逆向雅克比矩阵把笛卡尔空间的速度映射到关节空间的速度。

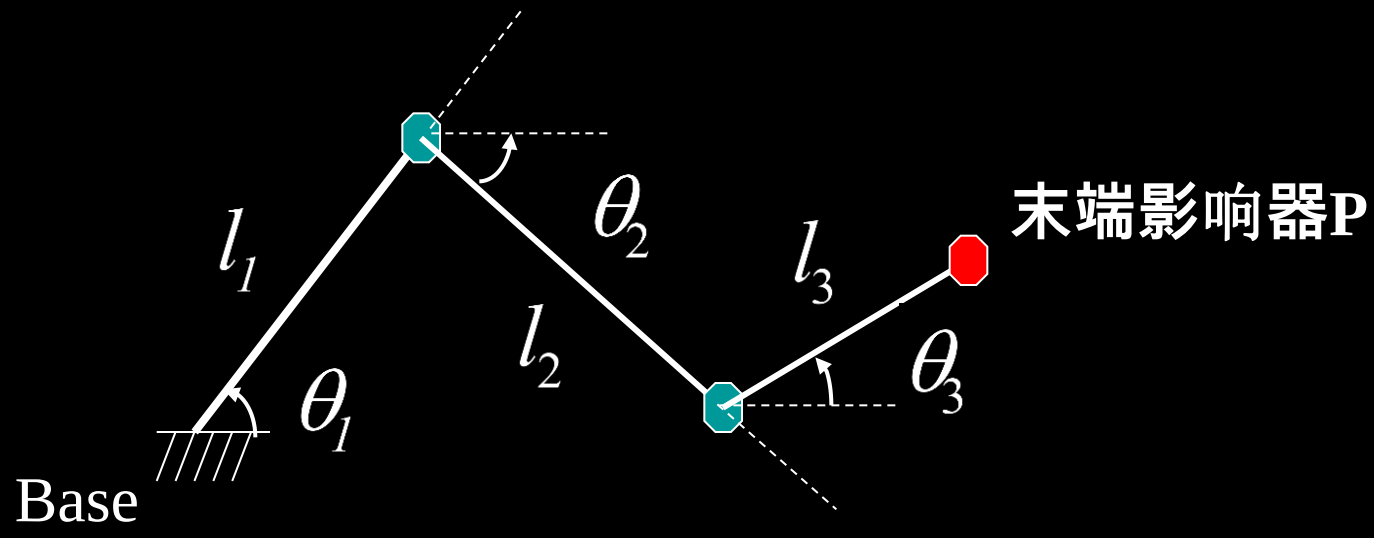
$$\dot{\theta} = J^{-1}V, \quad d\theta = J^{-1}dX$$

剩下的问题：如何计算雅克比矩阵？

$$d\theta = J^{-1}dX$$

- 解析计算
  - 对于简单关节可行
- 几何方法
  - 适合于复杂关节
  - 比如说，我们现在只关心末端影响器的位置e
    - 这时，雅克比矩阵为$3\times N$矩阵，其中N为自由度的数目
    - 对于每个自由度，我们分析e如何随自由度变化

解析计算雅克比矩阵——一个例子



解析计算雅克比矩阵——一个例子

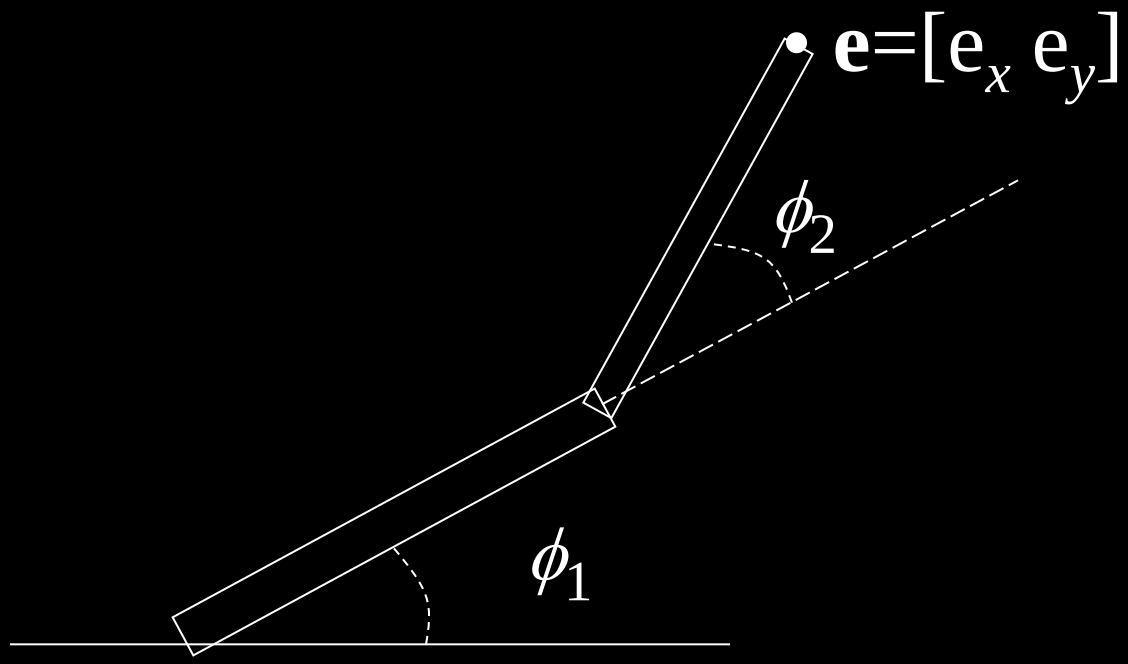
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{\theta}) \\ f_2(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} l_1\cos\theta_1 + l_2\cos\theta_2 + l_3\cos\theta_3 \\ l_1\sin\theta_1 - l_2\sin\theta_2 + l_3\sin\theta_3 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{J}\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial f_1(\boldsymbol{\theta})}{\partial\theta_1} & \dfrac{\partial f_1(\boldsymbol{\theta})}{\partial\theta_2} & \dfrac{\partial f_1(\boldsymbol{\theta})}{\partial\theta_3} \\ \dfrac{\partial f_2(\boldsymbol{\theta})}{\partial\theta_1} & \dfrac{\partial f_2(\boldsymbol{\theta})}{\partial\theta_2} & \dfrac{\partial f_2(\boldsymbol{\theta})}{\partial\theta_3} \end{bmatrix} = \begin{bmatrix} -l_1\sin\theta_1 & -l_2\sin\theta_2 & -l_3\sin\theta_3 \\ l_1\cos\theta_1 & -l_2\cos\theta_2 & l_3\cos\theta_3 \end{bmatrix}$$

雅克比矩阵计算的几何法——一段二维手臂

- 假设我们有一段简单的二维机器人手臂，包含两个自由度为1的旋转关节：



雅克比矩阵计算的几何法——一段二维手臂

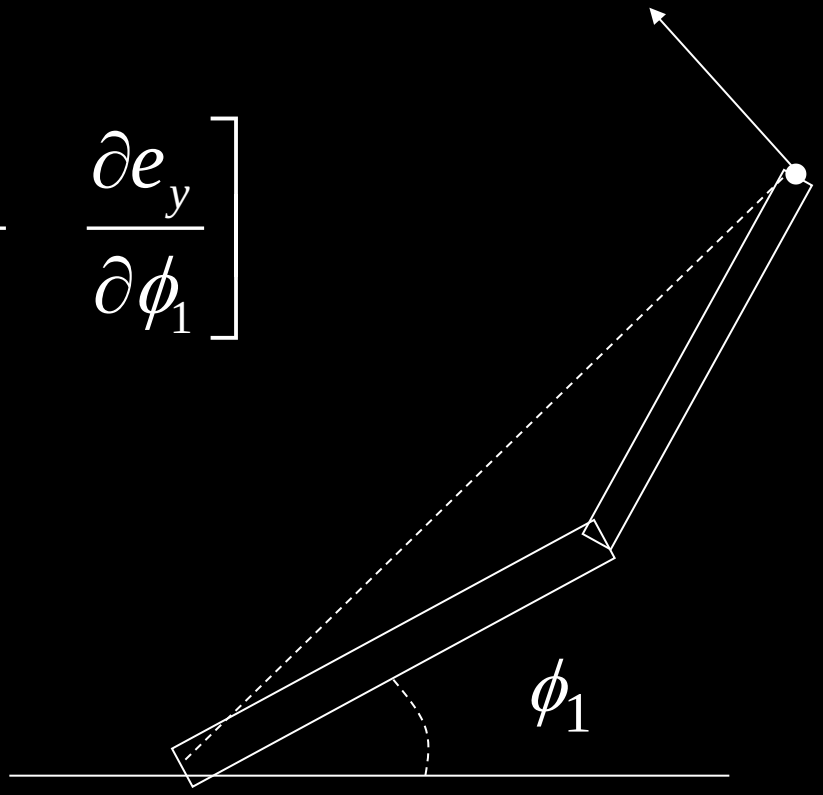
- 雅克比矩阵$J(\Phi)$显示了e的每个分量如何随每个关节角变化

$$J(\boldsymbol{\Phi}) = \begin{bmatrix} \frac{\partial e_x}{\partial \phi_1} & \frac{\partial e_x}{\partial \phi_2} \\ \frac{\partial e_y}{\partial \phi_1} & \frac{\partial e_y}{\partial \phi_2} \end{bmatrix}$$

雅克比矩阵计算的几何法——一段二维手臂

- 考虑如果把φ_1少量增加一点，e 会发生什么变化?

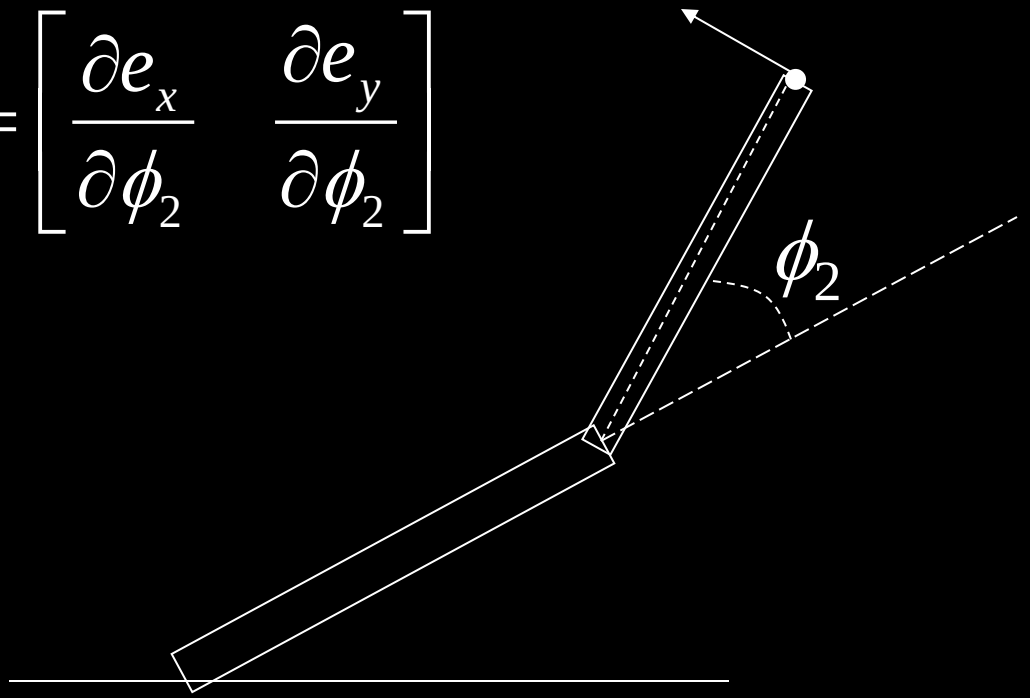
$$\frac{\partial \mathbf{e}}{\partial \phi_1} = \begin{bmatrix} \frac{\partial e_x}{\partial \phi_1} & \frac{\partial e_y}{\partial \phi_1} \end{bmatrix}$$



雅克比矩阵计算的几何法——一段二维手臂

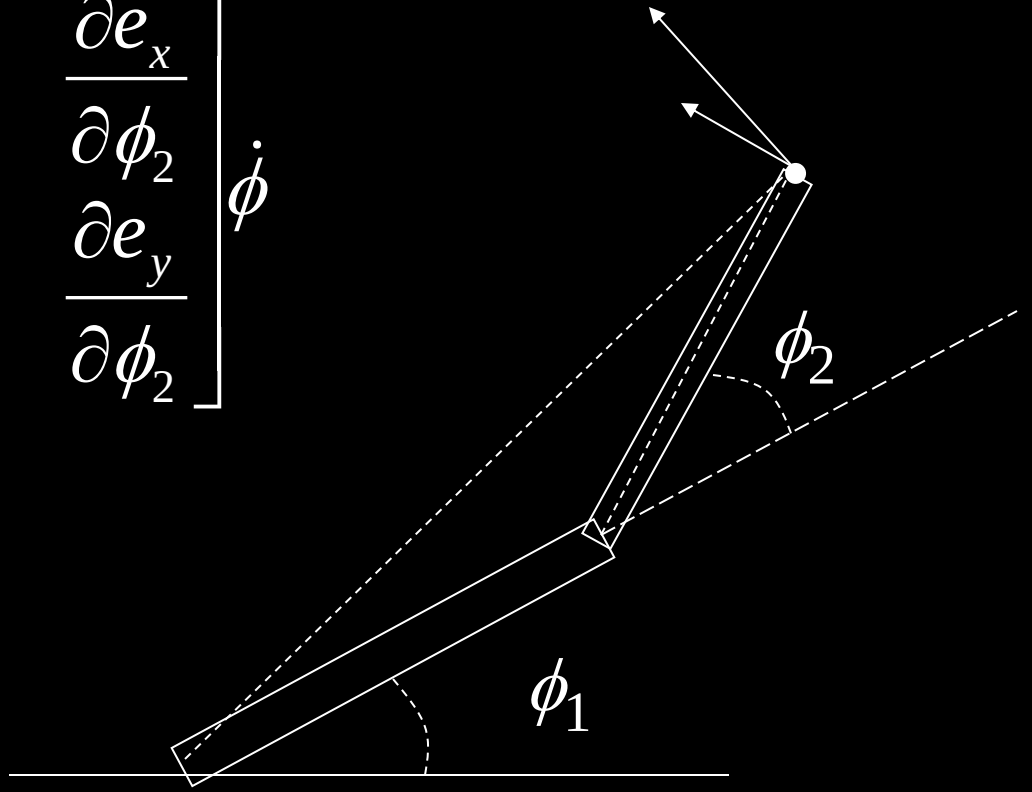
- 同理，如果把φ_2少量增加一点，e 会发生什么变化?

$$\frac{\partial \mathbf{e}}{\partial \phi_2} = \begin{bmatrix} \frac{\partial e_x}{\partial \phi_2} & \frac{\partial e_y}{\partial \phi_2} \end{bmatrix}$$

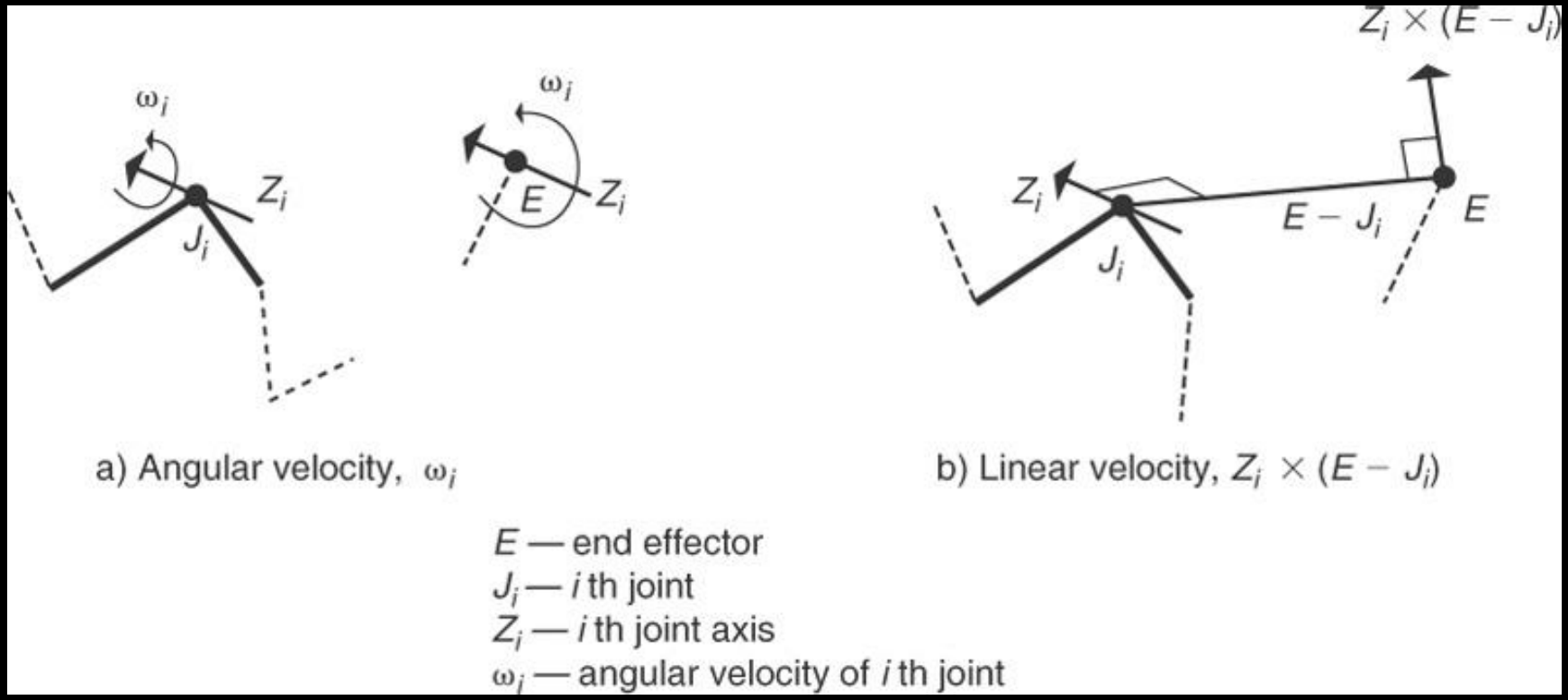


雅克比矩阵计算的几何法——一段二维手臂

$$V = J(\mathbf{\Phi})\dot{\phi} = \begin{bmatrix} \frac{\partial e_x}{\partial \phi_1} & \frac{\partial e_x}{\partial \phi_2} \\ \frac{\partial e_y}{\partial \phi_1} & \frac{\partial e_y}{\partial \phi_2} \end{bmatrix} \dot{\phi}$$



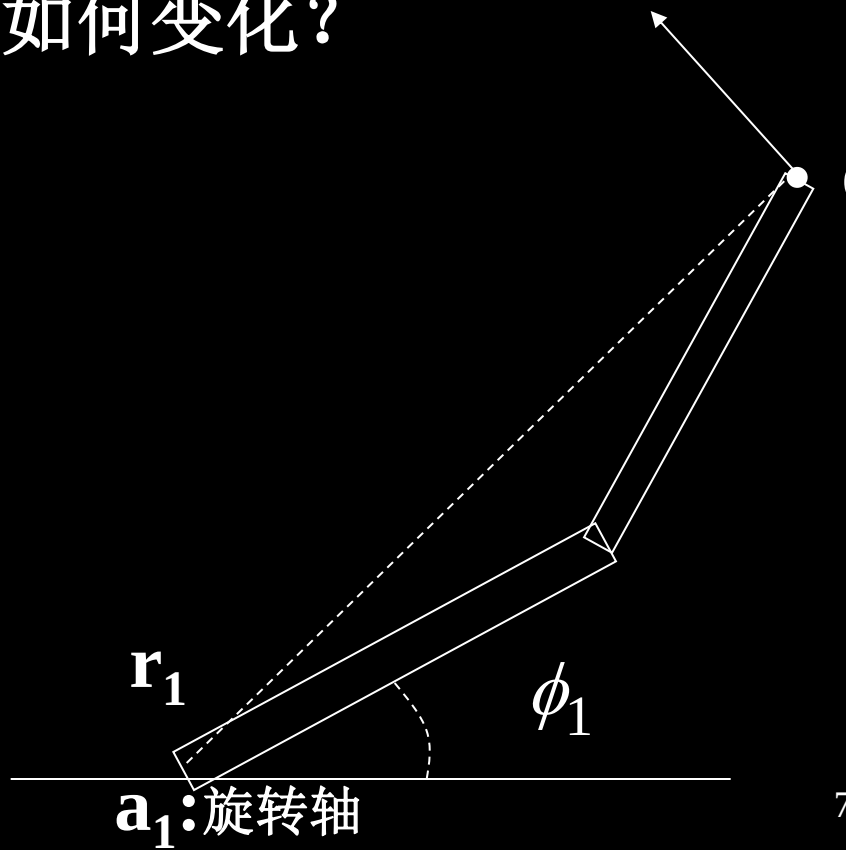
雅克比矩阵计算的几何法——一段二维手臂



由关节轴旋转引起的角速度和线速度

雅克比矩阵计算的几何法——一段二维手臂

- 让我们首先考虑1个旋转自由度关节的问题
- 我们想知道如果绕该关节的轴旋转，整体位置e如何变化？

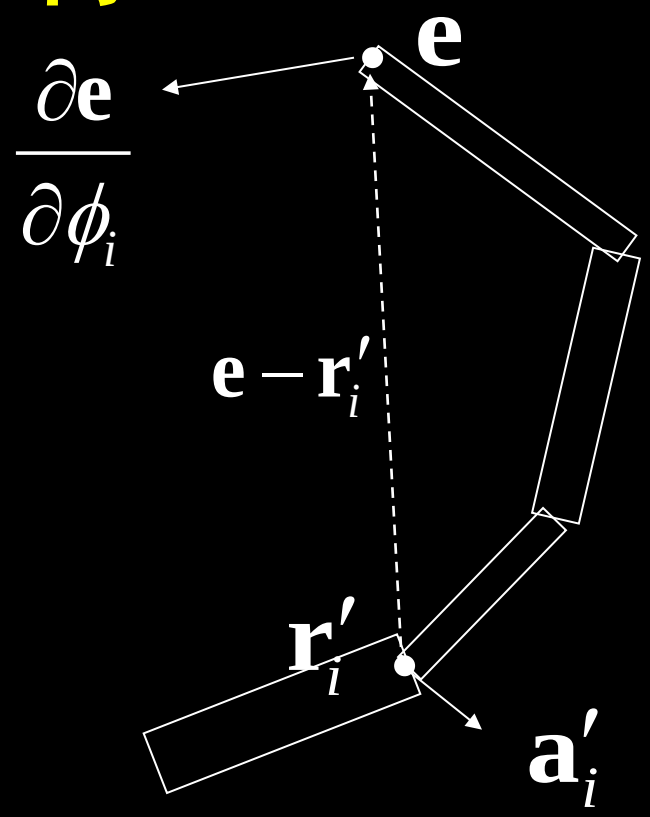


$$\frac{\partial \mathbf{e}}{\partial \phi_1} = \begin{bmatrix} \frac{\partial e_x}{\partial \phi_1} & \frac{\partial e_y}{\partial \phi_1} \end{bmatrix}$$

$$= \mathbf{a}_1 \times (\mathbf{e} - \mathbf{r}_1)$$

雅克比矩阵计算的几何法——一段二维手臂

$$\frac{\partial \mathbf{e}}{\partial \phi_i} = \mathbf{a}'_i \times \left(\mathbf{e} - \mathbf{r}'_i\right)$$



a'$_i$: 第*i*个关节在世界坐标系中的单位旋转轴

r'$_i$: 第*i*个关节在世界坐标系中旋转中心

e: 末端影响器在世界坐标系中位置

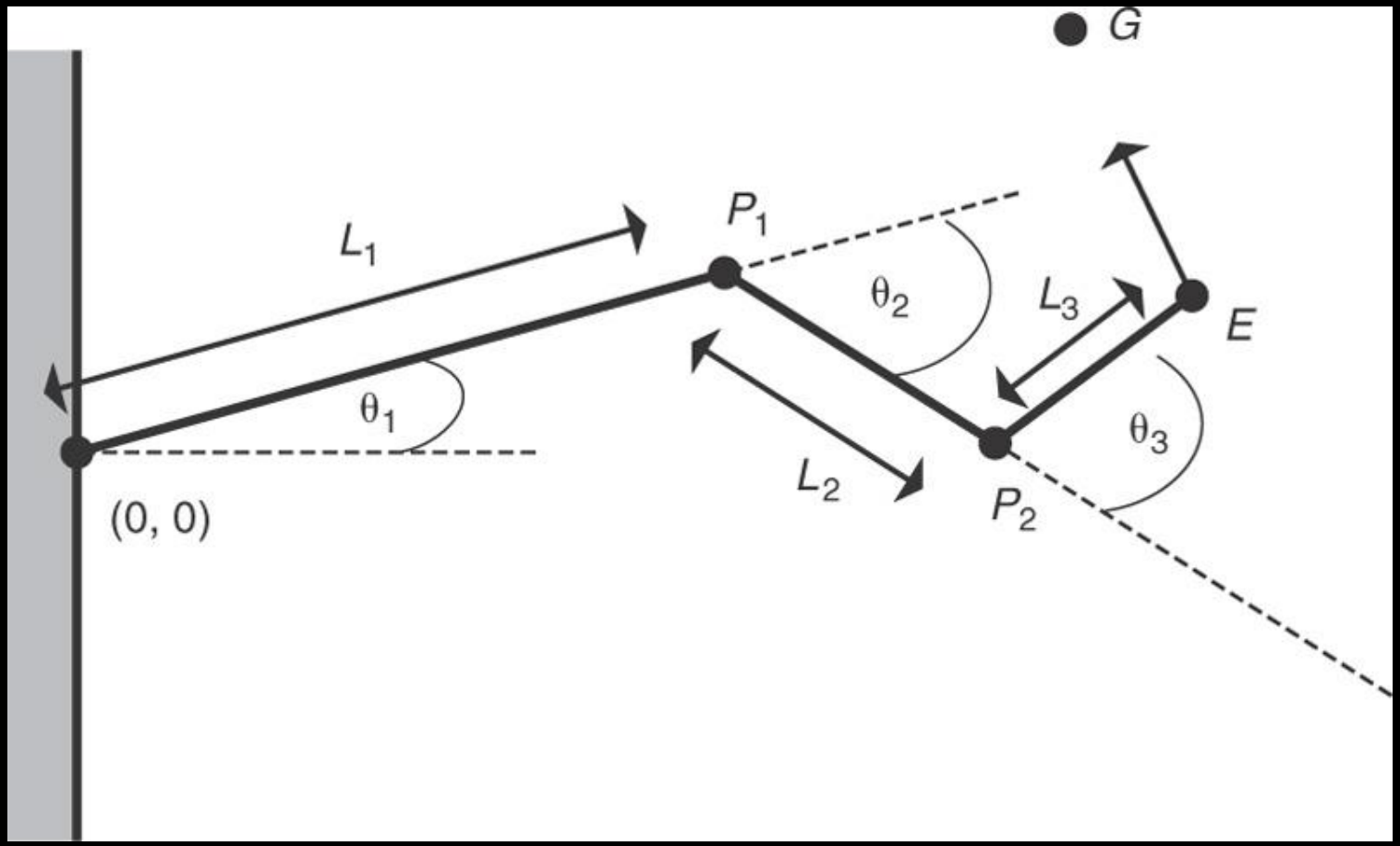
雅克比矩阵计算的几何法——3自由度旋转关节

- 一旦我们有了关节在世界坐标系的每个轴，我们就得到了雅克比矩阵的一列
- 我们把3自由度旋转关节当成3个1自由度的关节，以便可以用相同的公式来计算导数：

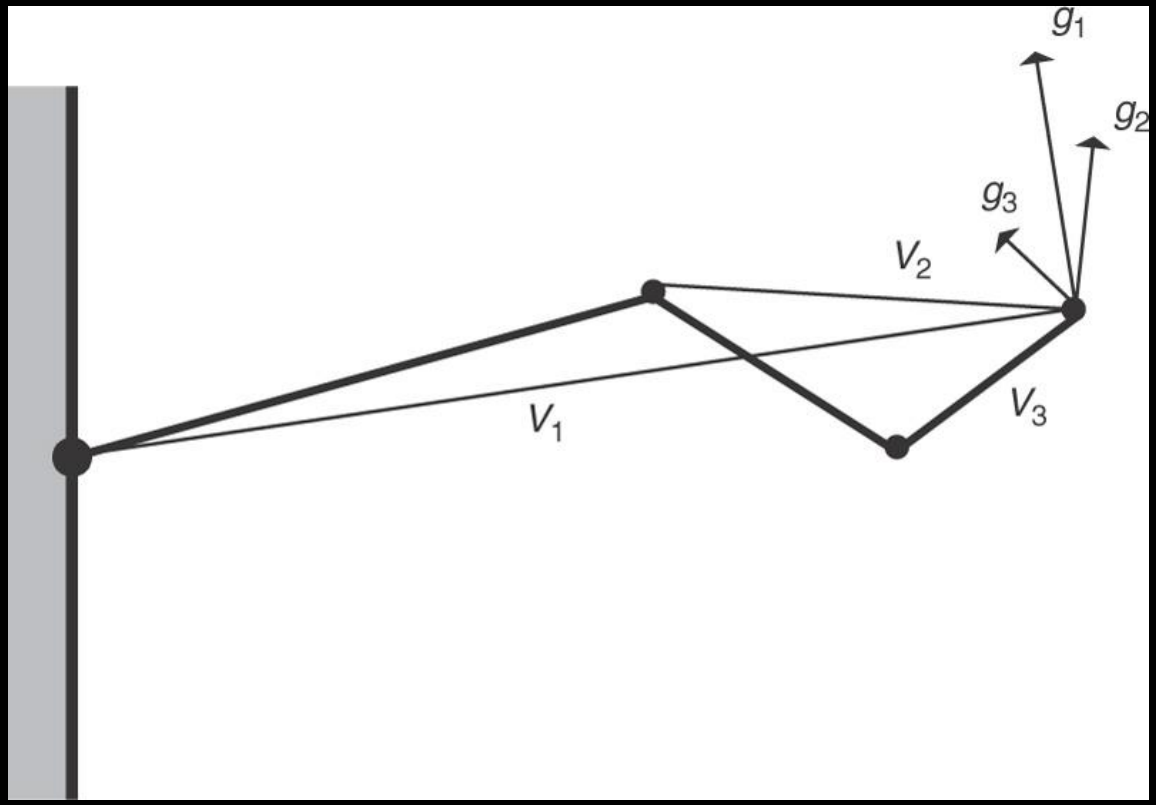
$$\frac{\partial \mathbf{e}}{\partial \phi_i} = \mathbf{a}'_i \times \left(\mathbf{e} - \mathbf{r}'_i\right)$$

  - 对三个轴的每个轴重复应用上述公式

雅克比矩阵计算的几何法——另一段二维手臂



雅克比矩阵计算的几何法——另一段二维手臂



- 问题变成：计算由各个关节引起的速度的线性组合，从而得到末端影响器的速度。

雅克比矩阵计算的几何法——另一段二维手臂

$$V=\begin{bmatrix}(G-E)_x\\(G-E)_y\\(G-E)_z\end{bmatrix}=J(\boldsymbol{\Phi})\dot{\phi}=\begin{bmatrix}\frac{\partial e_x}{\partial\phi_1} & \frac{\partial e_x}{\partial\phi_2} & \frac{\partial e_x}{\partial\phi_3}\\ \frac{\partial e_y}{\partial\phi_1} & \frac{\partial e_y}{\partial\phi_2} & \frac{\partial e_y}{\partial\phi_3}\end{bmatrix}\dot{\phi}$$

$$=\begin{bmatrix}((0,0,1)\times E)_x & ((0,0,1)\times(E-P_1)_x & ((0,0,1)\times(E-P_2)_x\\ ((0,0,1)\times E)_y & ((0,0,1)\times(E-P_1)_y & ((0,0,1)\times(E-P_2)_y\\ ((0,0,1)\times E)_z & ((0,0,1)\times(E-P_1)_z & ((0,0,1)\times(E-P_2)_z\end{bmatrix}\dot{\phi}$$