

机械臂理论介绍

- 串联机械臂理论
 - 机械臂类型
 - 机械臂坐标系表达与转换
 - 机械臂运动学
 - 机械臂动力学
 - 机械臂运动规划
 - 机械臂运动控制
 - 机械臂视觉应用

串联机械臂理论简介

- 参考视频
 - [机械臂运动学教程: 机械臂+旋转矩阵+变换矩阵+DH+逆解+轨迹规划](#)
- 参考书籍
 - 《机器人学导论》，J. J. Craig
 - 《现代机器人学：力学，规划，控制》，K. M. Lynch
 - LUH 机器人学课件

机械臂类型

常见的机械臂类型：

- 6自由度串联机械臂
 - 工作空间大
 - 灵活度高
 - 速度一般
 - 精度一般
 - 运动控制复杂



- 4自由度SCARA机械臂
 - 工作空间大

- 灵活度一般
- 速度快
- 精度高
- 运动控制相对简单



- 并联型机械臂
 - 工作空间相对安装空间较小
 - 灵活度低
 - 速度非常快
 - 精度非常高
 - 运动控制相对复杂



- 冗余机械臂
 - 关节数量大于末端的自由度
 - 能提供更多的灵活度，同样的目标位置，可以有不同的关节配置，以达到避障和避免奇异性的问题。
 - 运动控制相比普通6轴机械臂更复杂
 - 例子：KUKA-iiwa 7轴机械臂



- 视频资料
 - [机械臂的进化](#)
 - [7轴零空间运动](#)
 - [并联机械手](#)

坐标系表达与转换

- 可视化资料
 - [四元数可视化](#)
 - [在线旋转表达转换](#)
 - [VisRo: Making Robotics Easy to Learn through Visualization](#)

一些向量计算规则

- 向量内积，可以描述向量间的投影关系

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} = \sum_{i=1}^3 a_i b_i = |\mathbf{a}| |\mathbf{b}| \cos \langle \mathbf{a}, \mathbf{b} \rangle$$

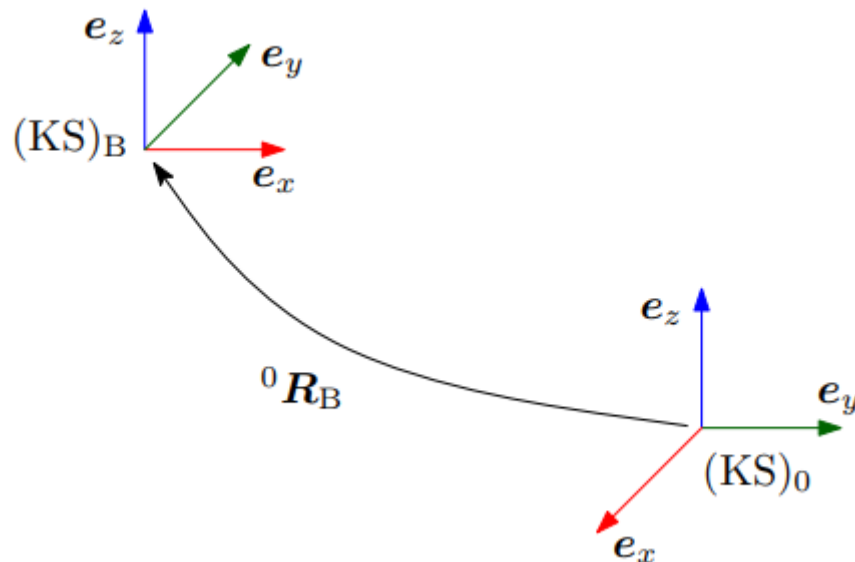
- 向量外积，外积的结果是一个垂直于叉乘的两个向量的向量，大小为 $|\mathbf{a}| |\mathbf{b}| \sin \langle \mathbf{a}, \mathbf{b} \rangle$ ，是两个向量张成的四边形的有向面积

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \mathbf{b} \stackrel{\text{def}}{=} \mathbf{a}^\wedge \mathbf{b}$$

- 反对称矩阵：

$$\mathbf{a}^\wedge = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

空间旋转的矩阵表达



- 旋转矩阵 0R_B 表示了坐标系 $(KS)_B$ 在坐标系 $(KS)_0$ 中的表达

$${}^0R_B = \begin{pmatrix} {}_{(0)}e_x^{(B)} & {}_{(0)}e_y^{(B)} & {}_{(0)}e_z^{(B)} \end{pmatrix}$$

$${}^0R_B = \begin{pmatrix} {}_{(0)}e_y^{(0)} & -{}_{(0)}e_x^{(0)} & {}_{(0)}e_z^{(0)} \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- 三维空间下旋转矩阵大小为 3x3
- 旋转矩阵是正交矩阵，其逆为自身的转置
- 旋转矩阵的行列式为1
- 旋转矩阵每一列的平方和为1

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} | RR^T = I, \det(R) = 1\}$$

$$R^{-1} = R^T$$

$$a' = R^{-1}a = R^T a$$

- 旋转矩阵属于特殊正交群 $SO(3)$ (Special Orthogonal Group)，（只有一个良好的运算的集合，称之为群）

$$R_1 + R_2 \notin SO(3)$$

$$R_1 R_2 \in SO(3)$$

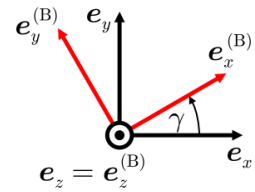
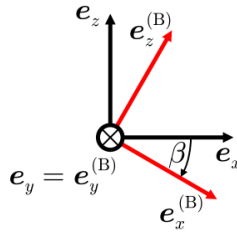
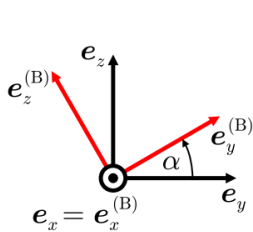
- 旋转矩阵不满足交换律

$$R_1 R_2 \neq R_2 R_1$$

空间旋转的复式变换表达（欧拉角）

- 基础角度变换
 - 任意两个坐标系之间的旋转关系，可以通过**三次**按照一定顺序的**基础旋转变换**来表达。
 - 基础旋转变换：指通过XYZ其中一个轴进行旋转得到的旋转变换
 - 旋转角度方向默认通过右手定则，即右手大拇指指向轴箭头方向，四指环绕方向指向旋转角正方向

$$\begin{aligned} \mathbf{R}_x(\alpha) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \\ \mathbf{R}_y(\beta) &= \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \\ \mathbf{R}_z(\gamma) &= \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned}$$



- 欧拉角的分类

- 按照旋转轴分类

- **Proper Euler angles** (z-x-z, x-y-x, y-z-y, z-y-z, x-z-x, y-x-y)

- 也被称为 classisc euler angles
 - 特点：第一次与第三次旋转角相同

- **Tait-Bryan angles** (x-y-z, y-z-x, z-x-y, x-z-y, z-y-x, y-x-z)

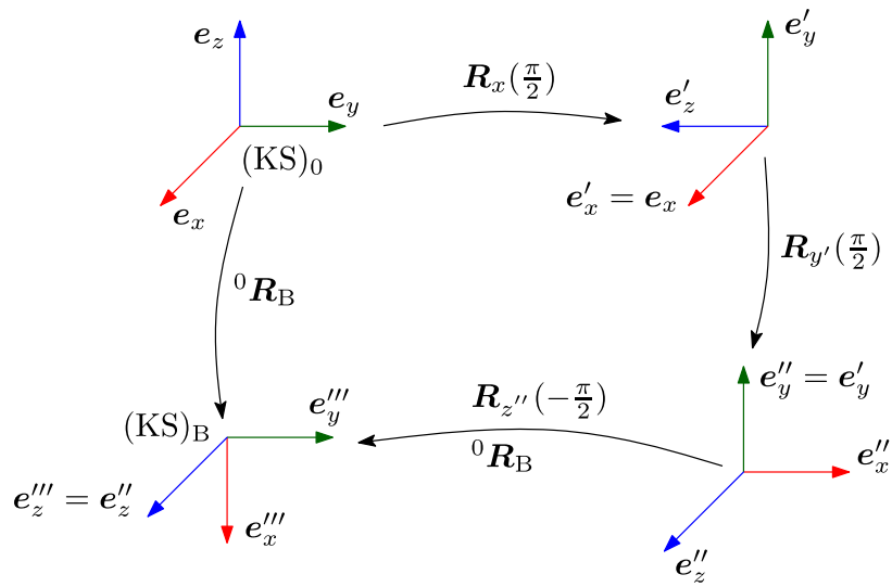
- 也被称为 Cardan angles, nautical angles, (heading, elevation, and bank), (RPY: yaw, pitch, and roll).
 - 特点：三次旋转角不同

- 按照参考坐标系分类

- 欧拉角-内旋法 intrinsic rotations (又称动态欧拉角)

- 基于每次旋转之后的坐标系
 - 旋转矩阵应用右乘顺序
 - 示例：(Cardan-Angle)
 1. 绕原坐标系的 x 轴旋转 α
 2. 绕新坐标系的 y 轴旋转 β
 3. 绕新坐标系的 z 轴旋转 γ

$$\mathbf{R}_{\text{KARD}}(\alpha, \beta, \gamma) = \underbrace{\overbrace{\mathbf{R}_x(\alpha)}^{1.} \overbrace{\mathbf{R}_y(\beta)}^{2.} \overbrace{\mathbf{R}_z(\gamma)}^{3.}}_{\text{right multiplication}}$$



- 欧拉角-外旋法 extrinsic rotations (又称静态欧拉角)

- 基于世界坐标系
- 旋转矩阵应用左乘顺序
- 示例: (RPY-Angle)
 1. 绕世界坐标系的 x 轴旋转 α
 2. 绕世界坐标系的 y 轴旋转 β
 3. 绕世界坐标系的 z 轴旋转 γ

$$\mathbf{R}_{\text{RPY}}(\alpha, \beta, \gamma) = \underbrace{\overbrace{\mathbf{R}_x(\gamma)}^{3.} \overbrace{\mathbf{R}_y(\beta)}^{2.} \overbrace{\mathbf{R}_z(\alpha)}^{1.}}_{\text{left multiplication}}$$

- 欧拉角 Gimbal-Lock 万向锁问题

- 一旦选择 $\pm 90^\circ$ 作为第二次旋转角度, 就会导致第一次旋转和第三次旋转等价(第三次的旋转相当于叠加在第一次旋转上), 丢失了一个表示维度。这种性质不适合插值和迭代等应用。
- [万向锁问题动画演示](#)
- [万向锁问题解释](#)

- 欧拉角的正逆解

- 正解: 通过三次基础旋转获得复式旋转矩阵

$$\mathbf{R}_{\text{xyz}} = \mathbf{R}_x(\alpha) \mathbf{R}_y(\beta) \mathbf{R}_z(\gamma)$$

- 逆解: 通过复式旋转矩阵逆推得到三次基础旋转
 - 分析法
 - 数值法
 - 多解

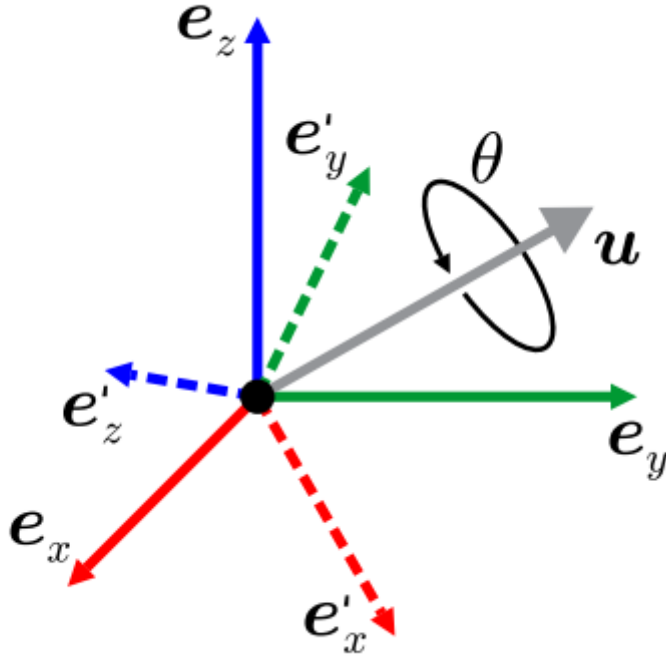
- 欧拉角表达法的优缺点

- 优点
 - 标记简单, 只需三个变量
 - 表达方式直观

- 缺点
 - 旋转顺序影响结果
 - 欧拉角表达方式有多种变体
 - 万向锁问题

空间旋转的轴角表达

- 任意旋转都可以用一个旋转轴和一个旋转角来刻画
- 旋转向量的方向与单位长度的旋转轴 \mathbf{u} 一致，长度等于旋转角 θ ，则可以表示为 $\theta\mathbf{u}$



- 罗德里格斯公式
 - 从旋转向量 $\theta\mathbf{u}$ 到旋转矩阵 \mathbf{R}

$$\mathbf{R} = \cos(\theta)\mathbf{I} + (1 - \cos(\theta))\mathbf{u}\mathbf{u}^T + \sin(\theta)\mathbf{u}^\wedge$$

$$\mathbf{u} = (u_x, u_y, u_z)^T$$

$$c_\theta = \cos(\theta)$$

$$s_\theta = \sin(\theta).$$

$$\mathbf{R} = \begin{pmatrix} u_x^2(1 - c_\theta) + c_\theta & u_x u_y(1 - c_\theta) - u_z s_\theta & u_x u_z(1 - c_\theta) + u_y s_\theta \\ u_x u_y(1 - c_\theta) + u_z s_\theta & u_y^2(1 - c_\theta) + c_\theta & u_y u_z(1 - c_\theta) - u_x s_\theta \\ u_x u_z(1 - c_\theta) - u_y s_\theta & u_y u_z(1 - c_\theta) + u_x s_\theta & u_z^2(1 - c_\theta) + c_\theta \end{pmatrix}$$

- 从旋转矩阵 \mathbf{R} 到旋转向量 $\theta\mathbf{n}$,
 - 方法1 (SLAMBOOK2)：罗德里格斯公式 (Rodrigues)，通过对 \mathbf{R} 求迹 (tr, trace, 即求矩阵对角线元素之和)：

$$\theta = \arccos\left(\frac{(\text{tr}(\mathbf{R})) - 1}{2}\right)$$

$$\theta = \arccos\left(\frac{(1 + 2 \cos(\theta)) - 1}{2}\right)$$

$$\mathbf{R}\mathbf{n} = \mathbf{n}, (\mathbf{R} - \mathbf{I})\mathbf{n} = \mathbf{0}$$

- 方法2 (LUH ROBOTIK1 Script) :

$$\mathbf{R}^* = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

$$\cos(\theta) = \frac{1}{2}(r_{11} + r_{22} + r_{33} - 1)$$

$$\sin(\theta) = \pm \frac{1}{2} \sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2}$$

$$\theta = \arctan 2(\sin(\theta), \cos(\theta))$$

$$\mathbf{u} = (u_x, u_y, u_z)^T = \frac{1}{2 \sin(\theta)} (r_{32} - r_{23}, r_{13} - r_{31}, r_{21} - r_{12})^T$$

- 轴角表达法的优缺点
 - 优点
 - 较少的参数数量
 - 适用于插值应用
 - 相对直观的表达
 - 缺点
 - 不方便表示连续旋转
 - 旋转向量的奇异性发生在转角 θ 超过 2π 时产生周期性

旋转的四元数表达

- 四元数没有奇异性，可以解决三维向量描述旋转时的奇异性问题
- 四元数基本定义

$$\mathbf{q} = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}$$

$$\mathbf{q} = [s, \mathbf{v}^T], s = q_0 \in \mathbb{R}, \mathbf{v} = [q_1, q_2, q_3]^T \in \mathbb{R}^3$$

- 四元数运算
 - 加减

$$\mathbf{q}_a \pm \mathbf{q}_b = [s_a \pm s_b, \mathbf{v}_a \pm \mathbf{v}_b]^T$$

- 乘法

$$\mathbf{q}_a \mathbf{q}_b = [s_a s_b - \mathbf{v}_a^T \mathbf{v}_b, s_a \mathbf{v}_b + s_b \mathbf{v}_a + \mathbf{v}_a \times \mathbf{v}_b]^T$$

- 模长

$$\|\mathbf{q}_a\| = \sqrt{s_a^2 + x_a^2 + y_a^2 + z_a^2}$$

$$\|\mathbf{q}_a \mathbf{q}_b\| = \|\mathbf{q}_a\| \|\mathbf{q}_b\|$$

- 共轭

$$\mathbf{q}^* = [s_a, -\mathbf{v}_a]^T$$

$$\mathbf{q}^* \mathbf{q} = \mathbf{q} \mathbf{q}^* = [s_a^2 + \mathbf{v}_a^T \mathbf{v}_a, \mathbf{0}]^T$$

- 逆

$$\begin{aligned} \boldsymbol{q}^{-1} &= \boldsymbol{q}^* / \|\boldsymbol{q}\|^2 \\ \boldsymbol{q}\boldsymbol{q}^{-1} &= \boldsymbol{q}^{-1}\boldsymbol{q} = 1 \\ (\boldsymbol{q}_a\boldsymbol{q}_b)^{-1} &= \boldsymbol{q}_b^{-1}\boldsymbol{q}_a^{-1} \end{aligned}$$

- 数乘

$$\mathbf{k}\boldsymbol{q} = [\mathbf{k}s, \mathbf{k}\boldsymbol{v}]^T$$

- 四元数旋转

- 空间三维点 $\boldsymbol{p} = [x, y, z] \in \mathbb{R}^3$ 经过旋转 \boldsymbol{q} 变为 \boldsymbol{p}'

$$\begin{aligned} \boldsymbol{p} &= [0, x, y, z]^T = [0, \boldsymbol{v}]^T \\ \boldsymbol{p}' &= \boldsymbol{q}\boldsymbol{p}\boldsymbol{q}^{-1} \end{aligned}$$

- 四元数转换

- 四元数到旋转矩阵

$$\boldsymbol{R} = \boldsymbol{v}\boldsymbol{v}^T + s^2\boldsymbol{I} + 2s\boldsymbol{v}^\wedge + (\boldsymbol{v}^\wedge)^2$$

$$\boldsymbol{R} = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}$$

- 旋转矩阵到四元数

$$\boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

$$\begin{aligned} q_0 &= \frac{\sqrt{\text{tr}(\boldsymbol{R}) + 1}}{2} \\ q_1 &= \frac{r_{23} - r_{32}}{4q_0} \\ q_2 &= \frac{r_{31} - r_{13}}{4q_0} \\ q_3 &= \frac{r_{12} - r_{21}}{4q_0} \end{aligned}$$

- 四元数到轴角表达

$$\begin{aligned} \theta &= 2 \arccos(s) \\ [n_x, n_y, n_z]^T &= [q_1, q_2, q_3]^T / \sin\left(\frac{\theta}{2}\right) \end{aligned}$$

- 四元数表达法的优缺点

- 优点

- 参数数量少
- 方便表达连续旋转
- 适用于插值计算
- 计算相比旋转矩阵更快
- 无奇异性

- 缺点

- 表达方式不够直观

齐次变换矩阵（空间旋转+位移）

- 三维空间下齐次变换矩阵大小为 4×4 ，属于特殊欧氏群 $SE(3)$
- 在三维空间中的欧氏齐次变换矩阵，由一个旋转矩阵 \mathbf{R} 和一个位移向量 \mathbf{t} 组成

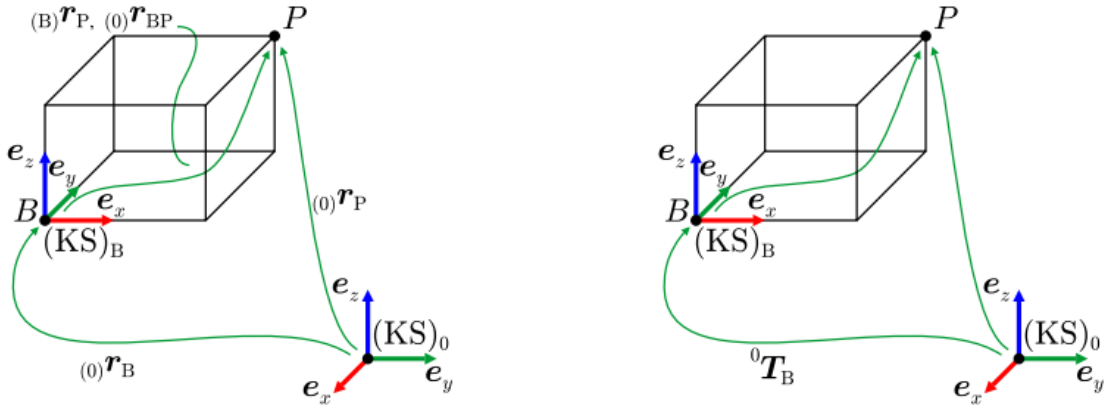
$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}$$

- 求逆，逆矩阵表示相反的变换

$$({}^0\mathbf{T}_B)^{-1} = {}^B\mathbf{T}_0 = \begin{bmatrix} ({}^0\mathbf{R}_B)^T & -({}^0\mathbf{R}_B)^T({}^0\mathbf{t}_B) \\ \mathbf{0} & 1 \end{bmatrix}$$

- 行列式为1

$$\det(\mathbf{T}) = 1.$$



- 通过齐次变换矩阵 ${}^0\mathbf{T}_B$ ，求解坐标系B中的点P ${}_{(B)}\mathbf{x}_P$ 在坐标系0中的坐标 ${}_{(0)}\mathbf{x}_P$ ，位置向量补齐时需要加元素 1：

$${}_{(0)}\mathbf{x}_P = ({}_{(0)}\mathbf{t}_P^T, 1)^T = ({}_{(0)}x_P, {}_{(0)}y_P, {}_{(0)}z_P, 1)^T$$

$${}_{(B)}\mathbf{x}_P = ({}_{(B)}\mathbf{t}_P^T, 1)^T = ({}_{(B)}x_P, {}_{(B)}y_P, {}_{(B)}z_P, 1)^T$$

$${}_{(0)}\mathbf{x}_P = {}^0\mathbf{T}_{B(B)}\mathbf{x}_P = \begin{pmatrix} {}^0\mathbf{R}_B & ({}^0\mathbf{t}_P) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}_{(B)}\mathbf{x}_P \\ 1 \end{pmatrix} = \begin{pmatrix} {}_{(0)}\mathbf{t}_B + {}^0\mathbf{R}_{B(B)}\mathbf{x}_P \\ 1 \end{pmatrix}$$

- 通过齐次变换矩阵 ${}^0\mathbf{T}_B$ ，求解坐标系 0 中的向量 ${}_{(0)}\mathbf{a}$ 在坐标系 B 中的向量 ${}_{(B)}\mathbf{a}$ ，向量补齐时加元素 0：

$${}_{(B)}\mathbf{a}' = {}^B\mathbf{T}_0\mathbf{a}' = \begin{pmatrix} {}^B\mathbf{R}_{0(0)}\mathbf{a}' \\ 0 \end{pmatrix}$$

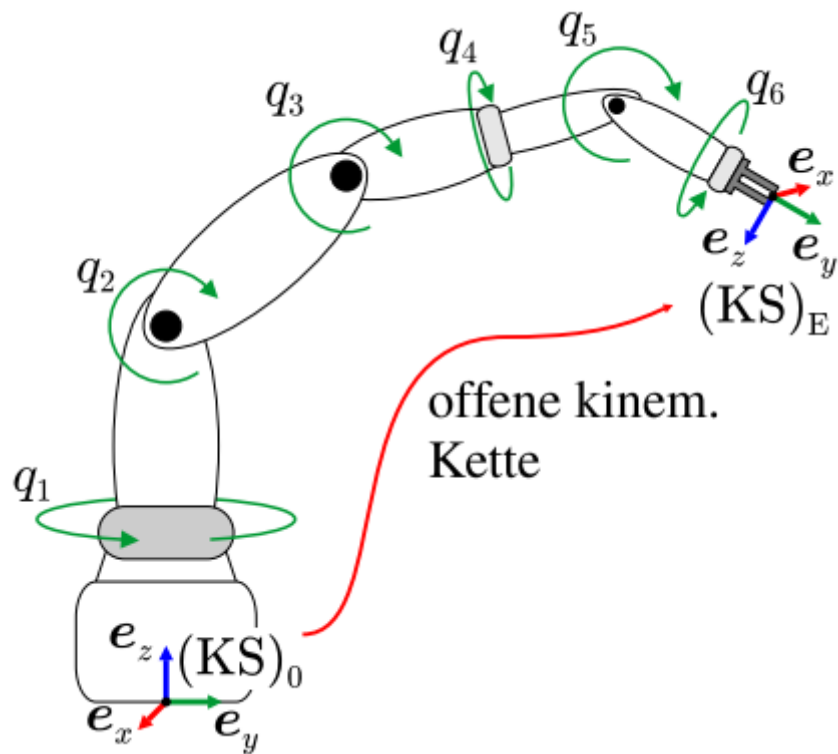
- 齐次变换矩阵的链式法则

$${}_{(0)}\mathbf{x} = {}^0\mathbf{T}_1{}^1\mathbf{T}_2 \cdots {}^{n-2}\mathbf{T}_{n-1}{}^{n-1}\mathbf{T}_{n(n)}\mathbf{x} = {}^0\mathbf{T}_{n(n)}\mathbf{x}$$

$${}^i\mathbf{T}_j = ({}^0\mathbf{T}_i)^{-1}{}^0\mathbf{T}_j = {}^i\mathbf{T}_0{}^0\mathbf{T}_j.$$

机械臂运动学

运动学定义



- 关节配置

- 以六自由度的机械臂为例，则关节配置为一个6维向量，表示每个关节的角度

$$\mathbf{q} = (q_1, q_2, \dots, q_6)^T$$

- 末端位姿

- 表示机械臂末端 Endeffector 相对于机械臂基底坐标系的空间位姿， x , y , z 表示三维空间坐标， ϕ, ψ, θ 表示末端坐标系到基底坐标系的三个欧拉角变化角度。

$$\mathbf{x}_E = ((0)x_E, (0)y_E, (0)z_E, \phi_E, \psi_E, \theta_E)^T$$

- 正运动学

- 已知各关节角度，求末端位姿

$$\mathbf{x}_E = f(\mathbf{q})$$

- 逆运动学

- 已知末端位姿，求各关节角度

$$\mathbf{q} = f(\mathbf{x}_E)$$

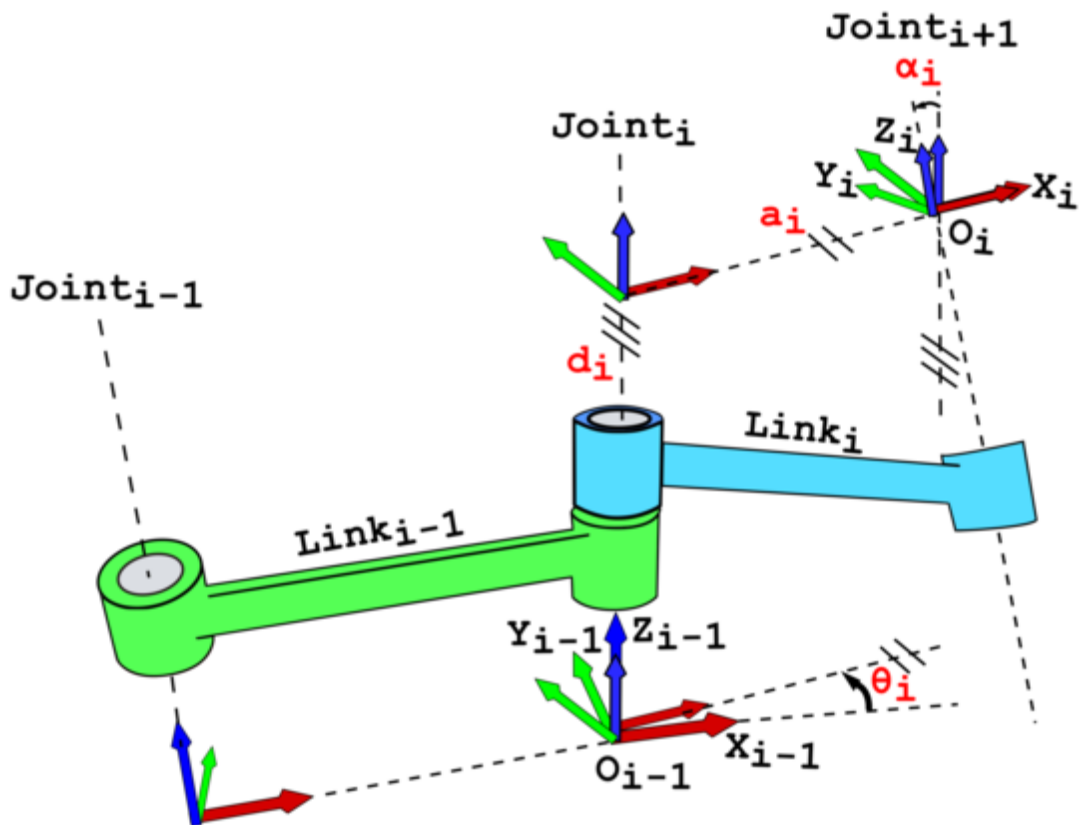
正运动学

- 已知各关节角度，求末端位姿，即找到每个关节之间的坐标转换，通过齐次变换链式法则得到从机械臂基座到末端关节的变换矩阵。

$$\mathbf{x}_E = f(\mathbf{q})$$

$${}^0\mathbf{T}_E(\mathbf{q}) = {}^0\mathbf{T}_1(q_1){}^1\mathbf{T}_2(q_2) \cdots {}^{n-1}\mathbf{T}_E(q_n)$$

- DH参数 (Denavit-Hartenberg parameters)：通过确立各个关节的坐标系，以及连续两个坐标系之间的齐次变换矩阵（通常为四个参数）参数，完成整个机械臂的建模。
 - 参考
 - [DH表示法\(Standard Version\)](#)
 - [DH表示法\(Craig Version\)](#)
 - [什么是改进DH参数](#)
 - [Denavit-Hartenberg parameters](#)
 - [在线DH参数可视化](#)
 - [为什么改进的DH模型优于标准DH模型，却大多都是使用标准D-H模型？](#)
- SDH 标准DH参数



1. 确定坐标系

1. 确定每个关节以及他们的顺序 $q = \{q_1, q_2, \dots, q_n\}$
2. 坐标系下标从0开始，若有 n 个关节，则有 $n+1$ 个坐标系
3. 关节 q_i 的旋转轴或位移轴为该关节坐标系的 z_{i-1} 轴，如果是旋转轴， z_{i-1} 轴方向通过右手定则确定正方向。
4. 每个 x_{i-1} 为 z_{i-1} 与 z_i 的公垂线
5. 通过 x_i 与 z_i 确定 y_i 以及该坐标系的坐标原点 O_i

2. 确定参数

- 两个相邻坐标系之间的参数的定义
 1. d_i : 沿 z_{i-1} 轴的位移，针对平移轴， q_i 会包括在该参数中
 2. θ_i : 绕 z_{i-1} 轴的旋转，针对旋转轴， q_i 会包括在该参数中
 3. a_i : 沿 x_i 轴的位移
 4. α_i : 绕 x_i 轴的旋转

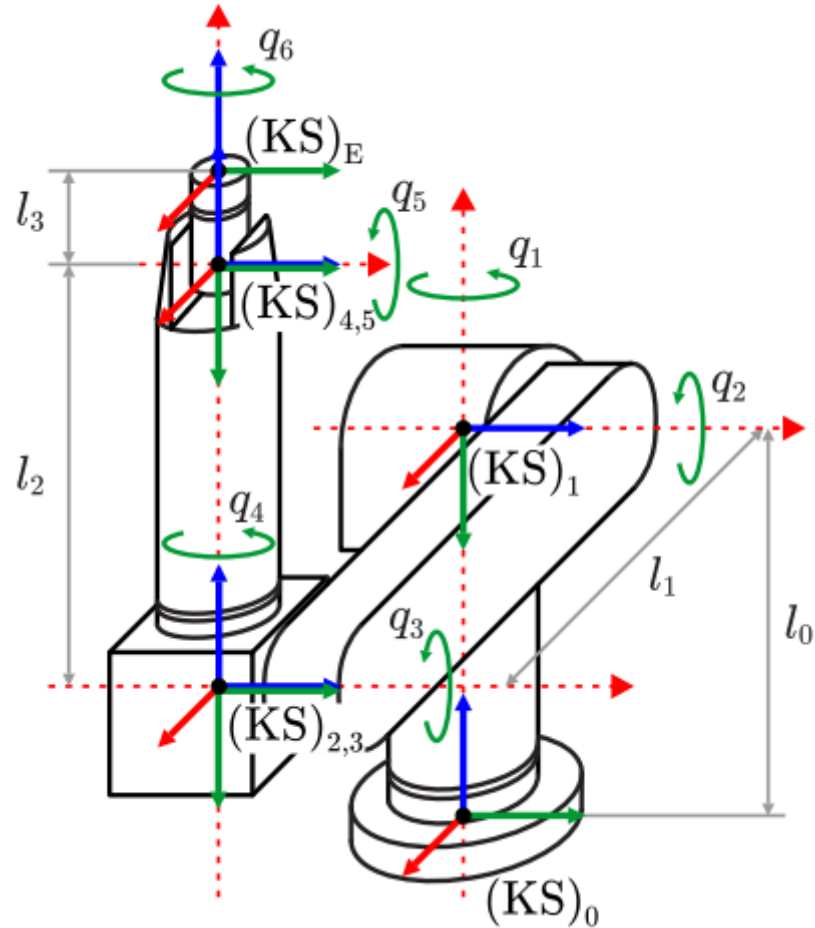
- 如果有 n 个关节，则有 $nx4$ 个参数

3. 确定变换矩阵

$${}^{i-1}T_i = \text{Trans}_{z_{i-1}}(d_i) \cdot \text{Rot}_{z_{i-1}}(\theta_i) \cdot \text{Trans}_{x_i}(a_i) \cdot \text{Rot}_{x_i}(\alpha_i)$$

$${}^0T_E(q) = {}^0T_1(q_1) {}^0T_2(q_2) \cdots {}^0T_n(q_n)$$

4. 举例



i	θ_i	d_i	a_i	α_i
1	q_1	l_0	0	$-\pi/2$
2	q_2	0	l_1	0
3	q_3	0	0	$\pi/2$
4	q_4	l_2	0	$-\pi/2$
5	q_5	0	0	$\pi/2$
6	q_6	l_3	0	0

• MDH 改进DH参数

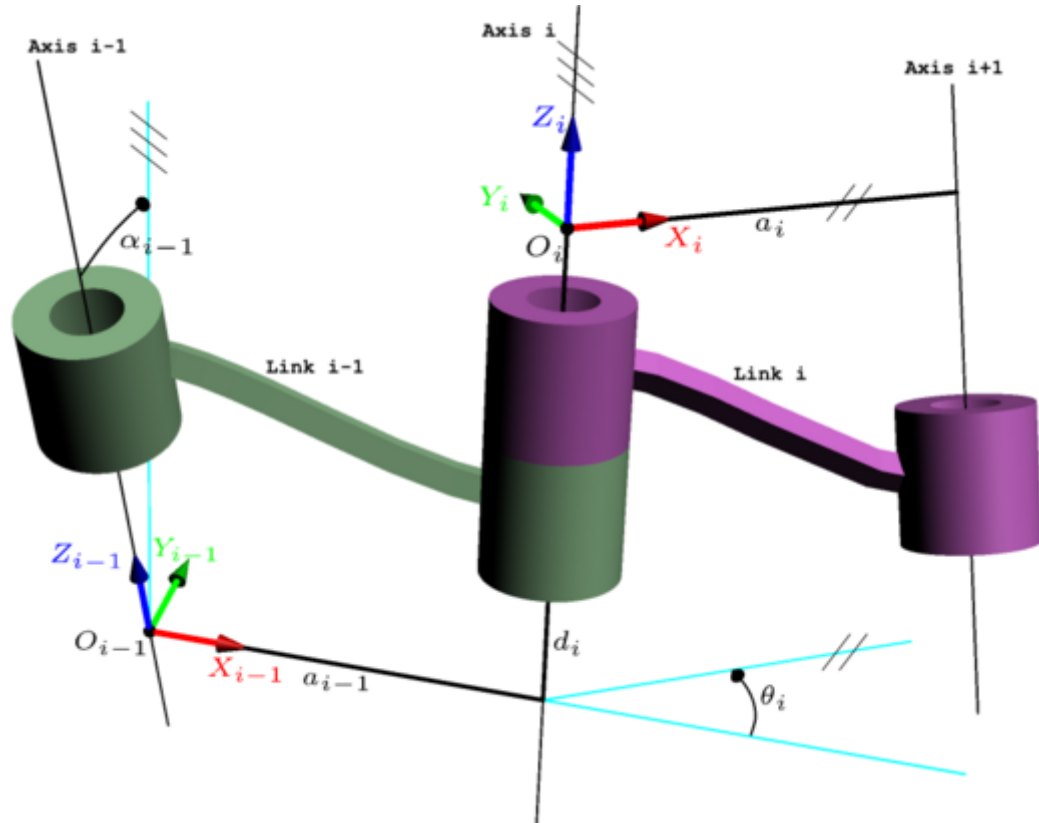
◦ 与 SDH 的区别

1. 关节 q_i 的旋转轴或位移轴为该关节坐标系的 z_i 轴，而非 z_{i-1}
2. 参数顺序为
 1. a_i : 沿 x_{i-1} 轴的位移
 2. α_i : 绕 x_{i-1} 轴的旋转

3. d_i : 沿 z_i 轴的位移, 针对平移轴, q_i 会包括在该参数中
4. θ_i : 绕 z_i 轴的旋转, 针对旋转轴, q_i 会包括在该参数中
3. 两个关节坐标系之间的转换顺序

$${}^{i-1}\mathbf{T}_i = \text{Rot}_{x_{i-1}}(\alpha_i) \cdot \text{Trans}_{x_{i-1}}(a_i) \cdot \text{Rot}_{z_i}(\theta_i) \cdot \text{Trans}_{z_i}(d_i)$$

4. 标准的 DH 坐标系在处理树形结构 (一个连杆末端连接两个关节) 的时候会产生歧义; 而改进的 DH 则没有这个问题。



逆运动学

- 参考资料
 - [机器人的运动学解——逆向运动学解](#)
 - [机械臂的逆运动学求解有统一的方法吗](#)
 - [MATLAB机器人工具箱中机器人逆解是如何求出来的](#)

- 已知末端位姿, 求各关节角度

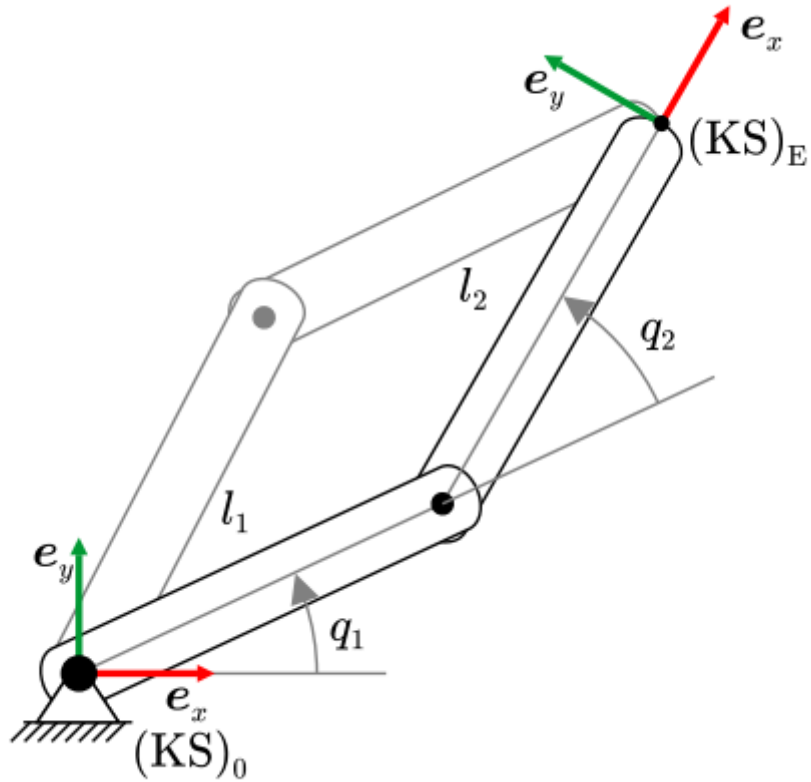
$$\mathbf{q} = f(\mathbf{x}_E)$$

- 逆运动学的解法
 - 分析法
 - 特点
 - 通过正运动学公式或通过几何关系进行分析推理
 - 针对非冗余机械臂有限个解 (存在多解)
 - 效率高于数值法
 - 随着自由度上升, 求得解析解的难度也越来越大
 - 通用性差, 适用性弱

- 一般的6自由度机械臂需要满足**Pieper准则**，即机器人满足以下两个充分条件中的一个，就会得到封闭解，这两个条件是：
 - 三个相邻关节轴相交于一点
 - 三个相邻关节轴相互平行
 - 具体方法
 - 通过对正运动学公式利用三角函数等进行推理
 - 通过几何关系进行推理
 - Rosen Diankov 于2010年前后提出的 IKfast 算法，适用于 $\text{dof} \leq 7$ 的机械臂
- 路径规划法
 - 在关节空间中进行路径规划
 - 基于图搜索
 - Dijkstra
 - A*
 - 基于采样
 - PRM
 - RRT
 - 例：OMPL 机械臂运动规划库
- 数值法
 - 特点
 - 适用性广，更适合冗余机械臂
 - 需要更多时间进行迭代计算
 - 受初始位姿影响
 - 收敛性无法保证，可能陷入局部最优
 - 具体方法
 - 优化法
 - 传统优化，通过构建优化方程 $f(q) - x_E = 0$ 获取目标 q
 - Steepest Descent (SD) method
 - Newton-Raphson (NR) method
 - Variable metric (VM) method
 - Levenberg-Marquardt (LM) method
 - 智能优化
 - 遗传优化
 - 粒子群优化
 - 神经网络优化
 - ...
 - 迭代法
 - 基于雅克比矩阵的微分运动学

分析法求解逆运动学的例子

- 例1：简单的 2DOF 平面连杆机械臂，通过**正运动学公式**分析求解逆运动学



$${}^0T_E = \left(\begin{array}{ccc|c} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \stackrel{!}{=} \left(\begin{array}{ccc|c} c_{\theta_E} & -s_{\theta_E} & 0 & ({}^{(0)}x_E \\ s_{\theta_E} & c_{\theta_E} & 0 & ({}^{(0)}y_E \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$({}^{(0)}x_E = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2),$$

$$({}^{(0)}y_E = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2).$$

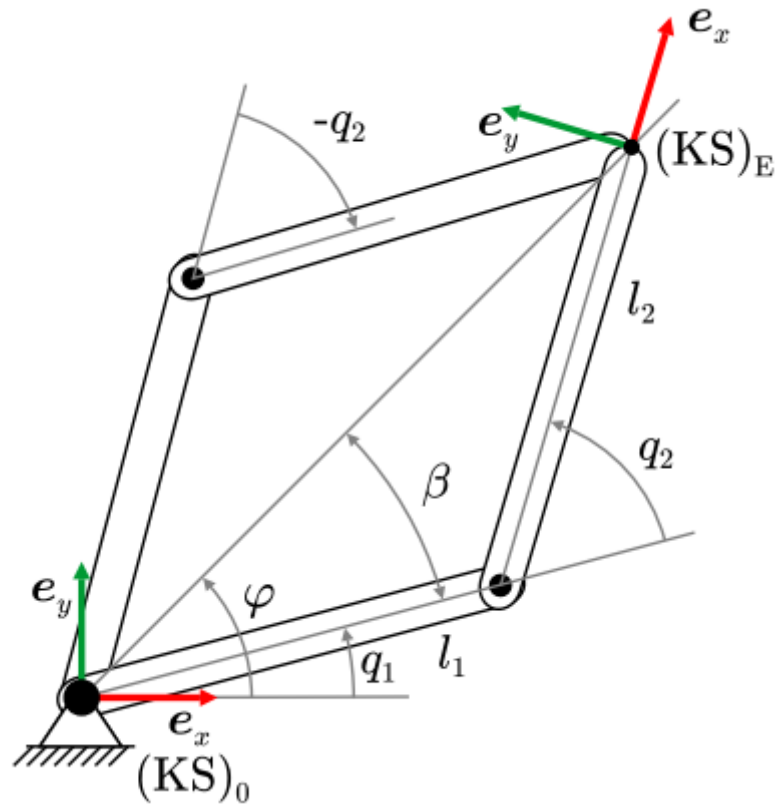
$$c_{\theta_E} = \cos(q_1 + q_2),$$

$$s_{\theta_E} = \sin(q_1 + q_2).$$

$$q_2 = \arctan2 \left(\pm \sqrt{1 - \left(\frac{({}^{(0)}x_E^2 + ({}^{(0)}y_E^2 - l_1^2 - l_2^2)}{2 l_1 l_2} \right)^2}, \frac{({}^{(0)}x_E^2 + ({}^{(0)}y_E^2 - l_1^2 - l_2^2)}{2 l_1 l_2} \right)$$

$$q_1 = \arctan2 \left(({}^{(0)}y_E, ({}^{(0)}x_E) \right) - \arctan2 \left(l_2 s_2, l_1 + l_2 c_2 \right)$$

- 例2：简单的 2DOF 平面连杆机械臂，通过几何关系分析求解逆运动学



$$({}_0)x_E^2 + ({}_0)y_E^2 = l_1^2 + l_2^2 - 2 l_1 l_2 \cos(\pi \pm q_2).$$

$$q_2 = \pm \arccos \left(\frac{{}_{(0)}x_E^2 + {}_{(0)}y_E^2 - l_1^2 - l_2^2}{2 l_1 l_2} \right)$$

$$\varphi = \arctan2 \left({}_{(0)}y_E, {}_{(0)}x_E \right)$$

$$\beta = \arccos \left(\frac{{}_{(0)}x_E^2 + {}_{(0)}y_E^2 + l_1^2 - l_2^2}{2l_1 \sqrt{{}_{(0)}x_E^2 + {}_{(0)}y_E^2}} \right)$$

$$q_1 = \varphi \pm \beta$$

- 其他例子：6自由度机械臂的分析法逆解
 - [工业机械臂逆运动学算法详解](#)
 - [UR和iwa协作机械臂逆运动学解析解算法](#)
 - [UR3机械臂正逆运动学详解](#)

- [傲博Aubo-i10机器人正逆运动学公式推导](#)

数值法求解逆运动学的例子

- cprobotics demo

其他软件工具

- Matlab Robotic Toolbox
 - 基于雅克比矩阵的迭代法
 - [MATLAB机器人工具箱中机器人逆解是如何求出来的](#)
- ROS MoveIt 中的工具
 - KDL
 - MoveIt 的默认运动学求解器，基于牛顿优化法
 - Track-IK
 - MoveIt 中基于雅克比矩阵的迭代法
- IKFAST - 获取 dof <= 7 自由度的解析解
 - [openrave-IKFAST](#)
 - [IKFAST-Generator \(超好用的 IKFAST 在线工具\)](#)
 - [逆运动学库：IKFAST库的使用：ikfast](#)

微分运动学（雅克比矩阵）

雅克比矩阵定义

- 机械臂的微分运动学描述了末端的速度或加速度与关节的速度或加速度之间的关系。
- 雅克比矩阵描述了末端与关节在微小形变下的线性关系：

$$\begin{aligned}\mathbf{x}_E &= \mathbf{f}(\mathbf{q}) \\ \Delta \mathbf{x}_E &= \mathbf{J}(\mathbf{q}) \Delta \mathbf{q} \\ \Delta \mathbf{q} &= \mathbf{J}^{-1}(\mathbf{q}) \Delta \mathbf{x}_E\end{aligned}$$

- 末端与关节的速度关系（一阶偏微分）：

$$\begin{aligned}\dot{\mathbf{x}}_E &= \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \\ \dot{\mathbf{q}} &= \mathbf{J}^{-1}(\mathbf{q}) \dot{\mathbf{x}}_E\end{aligned}$$

- 末端与关节的加速度关系（二阶偏微分）：

$$\begin{aligned}\ddot{\mathbf{x}}_E &= \mathbf{J}(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} &= \mathbf{J}^{-1}(\mathbf{q}) \ddot{\mathbf{x}}_E - \mathbf{J}^{-1}(\mathbf{q}) \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{J}^{-1}(\mathbf{q}) \dot{\mathbf{x}}_E\end{aligned}$$

- 雅克比矩阵的广义逆 \mathbf{J}^\dagger : 由于雅可比矩阵不一定是方阵，根据关节数量的不同，求逆方式如下：

$$\mathbf{J}^\dagger = \begin{cases} (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T, & n < 6 \\ \mathbf{J}^{-1}, & n = 6 \\ \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1}, & n > 6 \end{cases}$$

雅克比矩阵的分类

几何雅克比矩阵 J_G

- 描述的末端速度为相对于基地坐标系的速度向量以及角速度向量

$$\Delta \mathbf{x}_E = \begin{bmatrix} (0)\mathbf{v}_E \\ (0)\boldsymbol{\omega}_E \end{bmatrix} = \begin{bmatrix} (0)v_{E,x} \\ (0)v_{E,y} \\ (0)v_{E,z} \\ (0)\omega_{E,x} \\ (0)\omega_{E,y} \\ (0)\omega_{E,z} \end{bmatrix}$$

- 推导过程基于DH参数传递链：

$${}_{(0)}\dot{\mathbf{x}}_{E,G} = \begin{bmatrix} (0)\dot{\mathbf{x}}_{E,\text{trans}} \\ (0)\dot{\mathbf{x}}_{E,\text{rot}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{j}_{t,1} & \cdots & \mathbf{j}_{t,n} \\ \mathbf{j}_{r,1} & \cdots & \mathbf{j}_{r,n} \end{bmatrix}}_{J_G(\mathbf{q})} \dot{\mathbf{q}}$$

$$\text{translation joint: } \begin{bmatrix} \mathbf{j}_{t,i} \\ \mathbf{j}_{r,i} \end{bmatrix} = \begin{bmatrix} (0)\mathbf{e}_z^{i-1} \\ \mathbf{0} \end{bmatrix}$$

$$\text{rotation joint: } \begin{bmatrix} \mathbf{j}_{t,i} \\ \mathbf{j}_{r,i} \end{bmatrix} = \begin{bmatrix} (0)\mathbf{e}_z^{i-1} \times (0)\mathbf{r}_{i-1,E} \\ \mathbf{0} \end{bmatrix}$$

with:

$${}_{(0)}\mathbf{e}_z^{(i-1)} = \underbrace{{}_0\mathbf{R}_1^{-1}\mathbf{R}_2 \cdots {}^{i-2}\mathbf{R}_{i-1}}_{{}_0\mathbf{R}_{i-1}} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} (0)\mathbf{r}_{(i-1),E} \\ 0 \end{pmatrix} = \underbrace{{}_0\mathbf{T}_1^{-1}\mathbf{T}_2 \cdots {}^{n-1}\mathbf{T}_E}_{{}_0\mathbf{T}_E} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} - \underbrace{{}_0\mathbf{T}_1^{-1}\mathbf{T}_2 \cdots {}^{i-2}\mathbf{T}_{i-1}}_{{}_0\mathbf{T}_{i-1}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

解析雅克比矩阵 J_A

- 通过对正运动学公式进行偏微分推导而来

$$\mathbf{x}_E = \mathbf{f}(\mathbf{q})$$

$$\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} = \begin{pmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{pmatrix} \in \mathbf{R}^{m \times n}$$

- 描述的末端速度为相对于基底坐标系的速度分量以及**选择的空间旋转方式的角速度**，比如欧拉角内旋表达 $R_x(\alpha)R_y(\beta)R_z(\gamma)$ （即RPY），所以正运动学公式 \mathbf{f} 中需要包括从旋转矩阵 \mathbf{R} 推导到欧拉角旋转角度 α 、 β 、 γ 的内容：

$$\Delta \mathbf{x}_E = \begin{bmatrix} {}^{(0)}\mathbf{v}_E \\ {}^{(0)}\boldsymbol{\omega}_{E,\text{RPY}} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ {}^{(0)}\omega_{E,\alpha} \\ {}^{(0)}\omega_{E,\beta} \\ {}^{(0)}\omega_{E,\gamma} \end{bmatrix}$$

- 由于旋转速度的表达方式不同，解析雅克比矩阵 \mathbf{J}_A 与几何雅克比矩阵 \mathbf{J}_G 存在如下关系
 - 在某些特殊情况下，如平面连杆机构：

$$\mathbf{J}_A = \mathbf{J}_G$$

- 针对末端旋转为欧拉角内旋表达 $R_x(\alpha)R_y(\beta)R_z(\gamma)$ （即RPY）

- 角速度向量转换为RPY表达有以下关系：

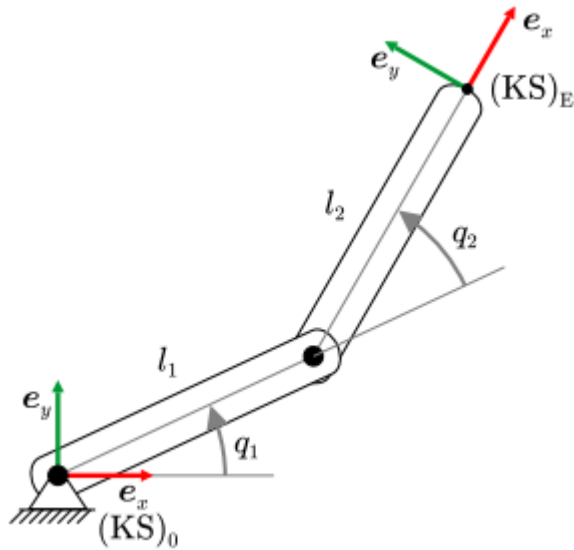
$$\begin{aligned} {}^{(0)}\boldsymbol{\omega}_E &= \begin{bmatrix} \dot{\alpha} \\ 0 \\ 0 \end{bmatrix}_{\text{Roll}} + \mathbf{R}_x(\alpha) \cdot \begin{bmatrix} 0 \\ \dot{\beta} \\ 0 \end{bmatrix}_{\text{Pitch}} + \mathbf{R}_x(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \begin{bmatrix} 0 \\ 0 \\ \dot{\gamma} \end{bmatrix}_{\text{Pitch}} \\ &= \underbrace{\begin{bmatrix} 1 & 0 & \sin(\beta) \\ 0 & \cos(\alpha) & -\sin(\alpha)\cos(\beta) \\ 0 & \sin(\alpha) & \cos(\alpha)\cos(\beta) \end{bmatrix}}_{\boldsymbol{\Omega}_{\text{RPY}}} \cdot \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}_{\text{RPY}} \end{aligned}$$

- 则解析雅克比矩阵 \mathbf{J}_A 与几何雅克比矩阵 \mathbf{J}_G 转换关系如下：

$$\mathbf{J}_G = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega} \end{bmatrix} \cdot \mathbf{J}_A$$

$$\mathbf{J}_A = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}^{-1} \end{bmatrix} \cdot \mathbf{J}_G$$

- 例：一个平面2轴机械臂的雅克比矩阵



$$\begin{aligned} {}^{(0)}x_E &= l_1 \cos(q_1) + l_2 \cos(q_1 + q_2), \\ {}^{(0)}y_E &= l_1 \sin(q_1) + l_2 \sin(q_1 + q_2). \end{aligned}$$

- 不涉及末端角速度的解析雅克比矩阵：

$$\mathbf{J} = \begin{pmatrix} -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{pmatrix}$$

- 涉及末端角速度的几何雅克比矩阵，因为是平面连杆机构，所以与解析雅克比矩阵等效

$$\mathbf{J}_g = \begin{pmatrix} \dot{\mathbf{j}}_{t_1} & \dot{\mathbf{j}}_{t_2} \\ \dot{\mathbf{j}}_{r_1} & \dot{\mathbf{j}}_{r_2} \end{pmatrix} = \begin{pmatrix} -l_2 \sin(q_1 + q_2) - l_1 \sin(q_1) & -l_2 \sin(q_1 + q_2) \\ l_2 \cos(q_1 + q_2) + l_1 \cos(q_1) & l_2 \cos(q_1 + q_2) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$$

雅克比矩阵的应用

- 用于机械臂轨迹插值以及逆运动学的数值解

$$\begin{aligned} \mathbf{q}_{i+1} &= \mathbf{q}_i + \Delta \mathbf{q}_i \\ &= \mathbf{q}_i + \mathbf{J}^{-1}(\mathbf{q}_i) \Delta \mathbf{x}_{E,i} \\ &= \mathbf{q}_i + \mathbf{J}^{-1}(\mathbf{q}_i) (\mathbf{x}_{E,i+1} - \mathbf{x}_{E,i}), \end{aligned}$$

- 用于计算机机械臂的操控性以及奇异性分析
 - 操控性越高，同样关节变化造成的末端变换更大，同时末端误差也会越大
 - 针对满秩的方阵雅克比矩阵，通过对雅克比矩阵进行特征分解(ED: Eigen Decomposition); 对于普通矩阵，进行奇异值分解 (SVD: Singular Value Decomposition) (参考 [矩阵分解——特征值分解与奇异值分解](#)) 获取各特征向量上的特征值 λ_i

$$\text{ED:} \quad \mathbf{J} = \mathbf{B} \mathbf{D} \mathbf{B}^{-1} = \mathbf{B} \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} \mathbf{B}^{-1}$$

$$\text{SVD:} \quad \mathbf{J} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \mathbf{U} \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & & \lambda_n \\ 0 & \cdots & 0 \end{pmatrix} \mathbf{V}^T$$

- 特征值 λ_i 表示机械臂末端在对应的特征向量上的操控性或灵活度：
 - $|\lambda_i|$ 大表示大的操作性或灵活度
 - $|\lambda_i|$ 小表示小的操作性或灵活度
 - $|\lambda_i| = 0$ 表示在该方向上没有可操作性或灵活度，表示机械臂在该方向上损失了一个自由度，遭遇了奇异性问题。
- 整体的操作性，注意当操作性为0时说明机械臂处于奇异性状态，机械臂末端损失了自由度，无法完成对应的位移。

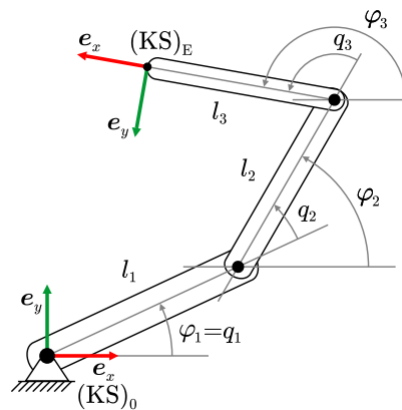
$$\mu(\mathbf{q}) = |\Pi_i^n \lambda_i| = |\det(\mathbf{J}(\mathbf{q}))|$$

- 机械臂在处于奇异性位置时的特点：

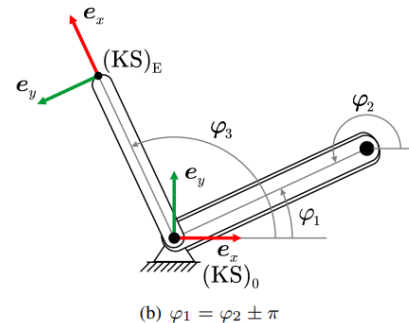
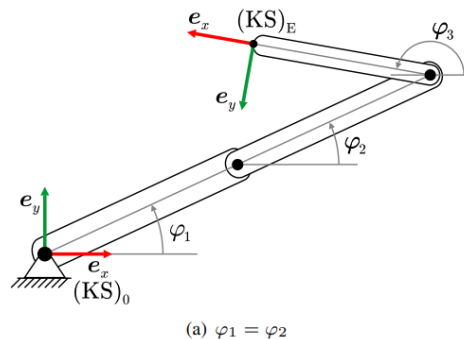
- 此时雅克比矩阵的行列式为 0

$$|\det(\mathbf{J}(\mathbf{q}))| = 0$$

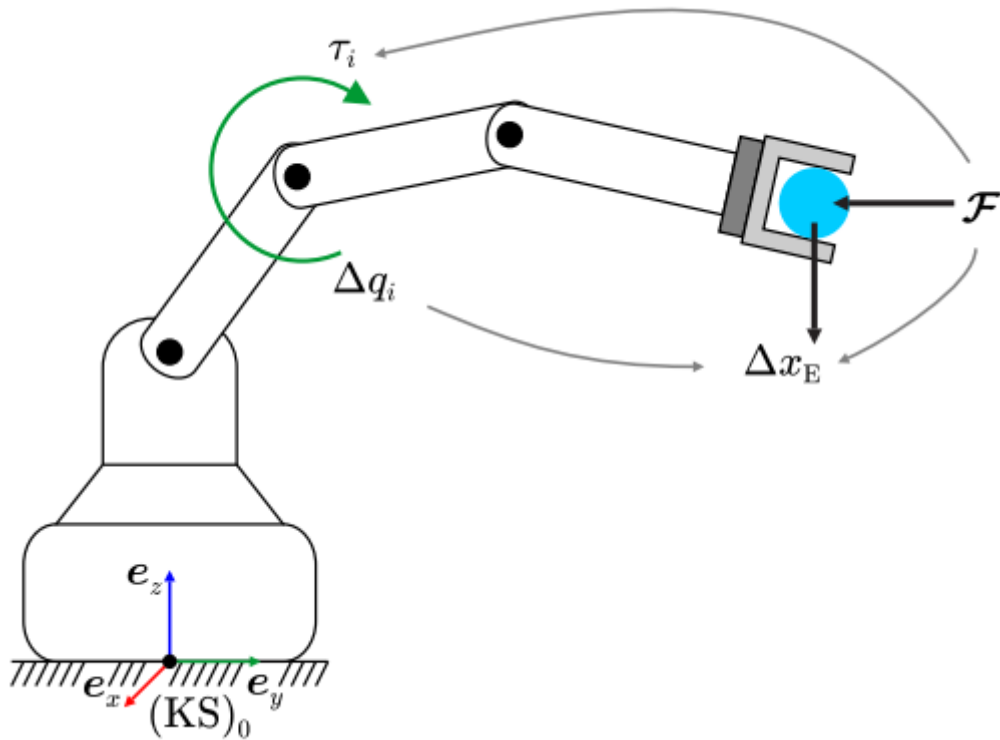
- 损失自由度
- 微小的末端位置改变，造成巨大的关节变化，造成关节速度突变
- 微小的末端负载会造成巨大的关节受力
- [What are robot singularities?](#)
- 例：平面三连杆机构的奇异性配置：
 - 非奇异性状态



- 奇异性状态：通过计算雅克比矩阵的行列式为0的情况，发现当关节2的角度为0或180度时，该机构陷入奇异性状态，此时末端在沿着第一关节的方向损失一个自由度。



- 解决奇异性的方法
 - 轨迹优化
 - 关节冗余设置
 - 力学控制，避免过大的受力
- 用于描述末端负载与关节力矩之间的关系
 - 例：机械臂末端受到的力与力矩 \mathcal{F} 对末端产生微小变化 $\Delta \mathbf{x}_E$ ，同时对关节产生力或力矩 $\boldsymbol{\tau}$ ，并发生形变 $\Delta \mathbf{q}$



- 通过虚功分析，可以得到以下关系：

$$\mathcal{F} = \begin{pmatrix} F \\ M \end{pmatrix} \in \mathbb{R}^6$$

$$\Delta x_E^T \mathcal{F} = \Delta q^T \tau$$

$$(J(q)\Delta q)^T \mathcal{F} = \Delta q^T \tau$$

$$\tau = J^T(q) \mathcal{F}$$

$$\mathcal{F} = (J^T(q))^{-1} \tau$$

- 用于刚性分析

- 假设每个关节的弹性模量为 k_i ，整个系统的关节弹性矩阵为

$\mathbf{K} = \text{diag}(k_1, \dots, k_n) = \text{diag}(\mathbf{k})$ ，则可以得出整个系统的末端弹性矩阵 \mathbf{C}_x ：

$$\tau = \mathbf{K} \Delta q$$

$$J^T(q) \mathcal{F} = \mathbf{K} \Delta q$$

$$J^T(q) \mathcal{F} = \mathbf{K} J^{-1}(q) \Delta x_E$$

$$\Delta x_E = \underbrace{J(q) \mathbf{K}^{-1} J^T(q)}_{\mathbf{C}_x} \mathcal{F}$$

$$\mathbf{C}_x = J(q) \mathbf{K}^{-1} J^T(q)$$

$$\Delta x_E = \mathbf{C}_x \mathcal{F}$$

$$\mathbf{C}_x = \mathbf{C}_x^T$$

- 对末端的弹性矩阵 \mathbf{C}_x 进行特征分解，有以下特点：

- 特征值越大，对应的特征向量表示了在该方向上其末端弹性越大，其刚性越小
- 特征值越小，对应的特征向量表示了在该方向上其末端弹性越小，其刚性越大

冗余机械臂

- 冗余机械臂定义

$$\dim(\mathbf{q}) > \dim(\mathbf{x}_E)$$

- 冗余机械臂的雅克比矩阵:

$$\mathbf{J}_{\text{redundant}} \in \mathbb{R}^{m \times n}, m < n$$

- 求逆:

$$\mathbf{J}^\dagger = \begin{cases} (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T, & n < 6 \\ \mathbf{J}^{-1}, & n = 6 \\ \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1}, & n > 6 \end{cases}$$

- 操控性计算，以及奇异性判断

$$\mu(\mathbf{q}) = \det(\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}))$$

- 用途

- 避障

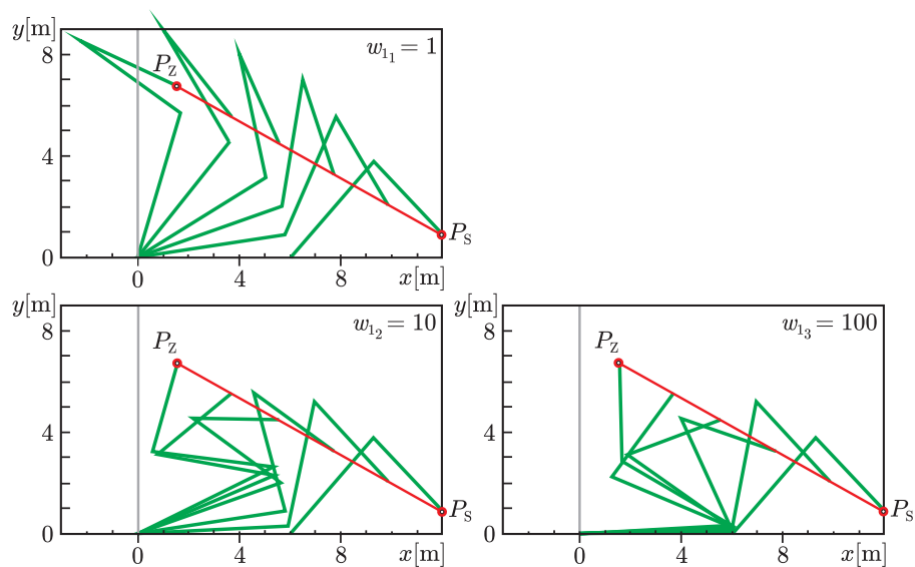
- 例: cprobotics: Demo04

- 避免奇异性

- 优化关节角度配置，零空间运动

- 末端位姿不变，改变关节配置，有无数解

- 例:



- 零空间运动的推导:

- 视频演示: [7 DOF Robotic Arm Null space motions](#)

- 通过对冗余机械臂的雅克比矩阵进行高斯变换，可以求解当前末端变化对应的关节角度变化的特解 (particular solution) 和通解 (homogenous solution)。当此刻关节角度的变化满足通解的条件时，将不会对末端产生影响。

$$\dot{\mathbf{x}}_E = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad \text{with } \mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n} (m < n) \quad \text{und} \quad \text{Rang}(\mathbf{J}(\mathbf{q})) = m$$

after Gauss transformation :

$$\mathbf{J}^* = \begin{pmatrix} \boxed{j_{11}} & j_{12} & 0 & 0 \\ 0 & 0 & \boxed{j_{23}} & 0 \\ 0 & 0 & 0 & \boxed{j_{34}} \end{pmatrix}$$

$$\boldsymbol{\delta} = \mathbf{J}^* \dot{\mathbf{q}}$$

$$\left(\begin{array}{cccc|c} \boxed{j_{11}} & j_{12} & 0 & 0 & \delta_1 \\ 0 & 0 & \boxed{j_{23}} & 0 & \delta_2 \\ 0 & 0 & 0 & \boxed{j_{34}} & \delta_3 \end{array} \right) \quad (\equiv \mathbf{J}^* \dot{\mathbf{q}} = \boldsymbol{\delta})$$

$$\dot{\mathbf{q}} = \underbrace{\begin{pmatrix} \frac{\delta_1}{j_{11}} \\ 0 \\ \frac{\delta_2}{j_{23}} \\ \frac{\delta_3}{j_{34}} \end{pmatrix}}_{\text{particular solution}} + \underbrace{\lambda \begin{pmatrix} -\frac{j_{12}}{j_{11}} \\ 1 \\ 0 \\ 0 \end{pmatrix}}_{\text{homogenous solution}} = \dot{\mathbf{q}}_p + \dot{\mathbf{q}}_h$$

$$\mathbf{J}^* \dot{\mathbf{q}}_h = \mathbf{0}$$

机械臂运动规划

- 路径规划（只考虑路径，不考虑时间、速度、加速度等）
 - 基于采样
 - 基于采样的方法不需要完整表示位型空间，所以非常适合高维和复杂环境问题，且相比 search-based（A* D*等）方法，能更有效的处理非凸和杂乱的环境。但是 sampling-based 方法只具有概率完备性，不一定总能找到最优解，解的质量取决于采样点的数量和分布；很难处理动力学等复杂约束问题；不能保证解的有一致性，比如相似的起点或者终点状态，生成的解不能保证是相似的，无法对规划结果进行预判，这就使得自动规划的机器人无法进入极端追求稳定性的工业领域。
 - 例：PRM, RRT, RRT*, OMPL (MoveIt 默认机械臂路径规划库)
 - [RRT, RRT-Star 和 RRT-Star-Smart](#)
 - [Rapidly exploring random tree](#)
 - demo: cprobotics/ArmNavigation/RRTStarSevenJointArmControl

- 基于优化
 - 将运动规划问题表述为一个优化问题，目标是在满足机器人环境约束和动力学约束下，找到优化某个成本函数的轨迹。可以设计各种成本函数来实现不同的优化目标，比如最短路径、最优时间、最优能量等。基于优化的方法可以找到高质量的解决方案，且解具有一致性，可以处理机器人动力学和环境的复杂约束。但是 optimization-based 方法计算成本很高，尤其是对于高维系统；需要准确的机器人动力学和环境模型，这在实践中可能很难得到；在复杂环境下容易陷入局部最优解。
 - 例：CHOMP、STOMP
- 轨迹规划（考虑时间、速度、加速度）

机械臂动力学