1 Robotics

FK：正向运动学给定关节角度，计算末端执行器的位置和姿态。IK：给定正向运动学 $T_{s\to e}(\theta)$ 和目标姿态 $T_{target}=\mathbb{SE}(3)$，求解满足以下条件的关节角度 θ：$T_{s\to e}(\theta)=T_{target}$。IK 比 FK 更复杂，因为 T^{-1} 可能很难计算，所以 **通常可能多解或无解**。

三维空间中 (R,t) “完全自由度”配置。至少 6 个自由度可以保证覆盖此空间，从而 IK 有解（但有时候可能得不到解析解，只能得到数值解）。引理：*如果机械臂构型满足 Pieper Criterion，则有解析解（闭式解）*。6DoF 保证有解，但是这个解可能超出了可行空间（如碰撞解），所以额外增加 1 冗余形成 7 DoF，可扩大解空间，更有可能找到可行解。但一味增加自由度会带来工程复杂性并延长反应时间。目前工业界一般 6 或者 7 DoF。

欧拉角：$R_x(\alpha):=\begin{bmatrix}1&0&0\\0&\cos\alpha&-\sin\alpha\\0&\sin\alpha&\cos\alpha\end{bmatrix}$，$R_y(\beta):=\begin{bmatrix}\cos\beta&0&\sin\beta\\0&1&0\\-\sin\beta&0&\cos\beta\end{bmatrix}$，$R_z(\gamma):=\begin{bmatrix}\cos\gamma&-\sin\gamma&0\\\sin\gamma&\cos\gamma&0\\0&0&1\end{bmatrix}$

任意旋转均可拆为 $R=R_z(\alpha)R_y(\beta)R_x(\gamma)$。这个顺序可以变，但一般默认是这个顺序。问题：1. 对于一个旋转矩阵，**其欧拉角可能不唯一**。2. **Gimbal Lock**：如果三次旋转中第二次旋转 β 的角度为 $\pi/2$，那么剩下 2 个自由度会变成 1 个。

欧拉定理/axis-angle：任意三维空间中的旋转都可以表示为绕一个固定轴 $\hat\omega\in\mathbb{R}^3$（单位向量，满足 $\|\hat\omega\|=1$）旋转一个正角度 θ 的结果。问题：**不唯一性**：$(\hat\omega,\theta)$ 和 $(-\hat\omega,-\theta)$ 代表同一个旋转；$\theta=0$ 时对应 $R=I$，任意 $\hat\omega$ 都行；$\theta=\pi$ 时，绕轴 $\hat\omega$ 和绕轴 $-\hat\omega$ 旋转 π 得到的结果是相同的。这种情况对应 $\mathrm{tr}(R)=-1$。将旋转角 θ 限制在 $(0,\pi)$ 内，那么对于大部分旋转，其轴角表示就是唯一的（不考虑不旋转、旋转 π）。

对于一个单位轴向量（axis）$\mathbf{u}=[x,y,z]^\top$，其对应的叉乘矩阵（cross product matrix）$K=\begin{bmatrix}0&-z&y\\z&0&-x\\-y&x&0\end{bmatrix}$。

Rodrigues 旋转公式（向量形式）：向量 $\mathbf{v}$ 沿着单位向量 $\mathbf{u}$ 旋转 θ 角度之后的向量 $\mathbf{v}'$ 为 $\mathbf{v}'=\cos(\theta)\mathbf{v}+(1-\cos(\theta))(\mathbf{u}\cdot\mathbf{v})\mathbf{u}+\sin(\theta)(\mathbf{u}\times\mathbf{v})$ **（矩阵形式）**绕单位轴 $\mathbf{u}$ 旋转 θ 的旋转矩阵 R_θ 可以表示为 $R_\theta=I+(1-\cos\theta)K^2+\sin\theta\cdot K=e^{\theta K}$。证明：拆开级数然后用 $K^2=\mathbf{u}\mathbf{u}^\top-I,K^3=-K$

从旋转矩阵 R 反求 $(\hat\omega,\theta)$：当 $\theta\in(0,\pi)$ 时，$\theta=\arccos\frac{1}{2}[\mathrm{tr}(R)-1]$，$[\hat\omega]=\frac{1}{2\sin\theta}(R-R^\top)$。定义两个旋转矩阵之间的 **旋转距离**：从姿态 R_1 转到姿态 R_2 所需的最小旋转角度。两个旋转的关系是：$(R_2R_1^\top)R_1=R_2$。旋转距离：$\mathrm{dist}(R_1,R_2)=\theta(R_2R_1^\top)=\arccos\left(\frac{1}{2}[\mathrm{tr}(R_2R_1^\top)-1]\right)$

Quaternion：$q=w+xi+yj+zk$。其中 w 实部，x,y,z 虚部。i,j,k 是虚数单位，满足 $i^2=j^2=k^2=ijk=-1$ 和反交换性质（**没有交换律**）：$ij=k=-ji$，$jk=i=-kj$，$ki=j=-ik$。可以表示为向量形式 $q=(w,\mathbf{v})$，$\mathbf{v}=(x,y,z)$。

乘法：$q_1q_2=(w_1w_2-\mathbf{v}_1^\top\mathbf{v}_2,w_1\mathbf{v}_2+w_2\mathbf{v}_1+\mathbf{v}_1\times\mathbf{v}_2)=(w_1w_2-\mathbf{v}_1\cdot\mathbf{v}_2,w_1\mathbf{v}_2+w_2\mathbf{v}_1+\mathbf{v}_1\times\mathbf{v}_2)$，**不可交换**，即 $q_1q_2\neq q_2q_1$。**共轭**：$q^*=(w,-\mathbf{v})$；**模长**：$\|q\|^2=w^2+\mathbf{v}^\top\mathbf{v}=qq^*=q^*q$；**逆**：$q^{-1}=\frac{q^*}{\|q\|^2}$

单位四元数 $\|q\|=1$，可表示三维空间中的旋转，$q^{-1}=q^*$。

旋转表示：绕某个单位向量 $\hat\omega$ 旋转 θ 角度，对应的四元数：$q=\left[\cos\frac{\theta}{2},\sin\frac{\theta}{2}\hat\omega\right]$

注意，旋转到四元数存在 **“双重覆盖”**关系。从四元数恢复轴角表示：$\theta=2\arccos(w)$，$\hat\omega=\begin{cases}\frac{\mathbf{v}}{\sin(\theta/2)},&\theta\neq0\\0,&\theta=0\end{cases}$。

向量旋转：任意向量 $\mathbf{v}$ 沿着以**单位向量**定义的旋转轴 $\mathbf{u}$ 旋转 θ 度得到 $\mathbf{v}'$，那么：令向量 $\mathbf{v}$ 的四元数形式 $v=[0,\mathbf{v}]$，旋转四元数 $q=\left[\cos\left(\frac{\theta}{2}\right),\sin\left(\frac{\theta}{2}\right)\mathbf{u}\right]$，则旋转后的向量 $\mathbf{v}'$ 可表示为：$\mathbf{v}'=qvq^*=qvq^{-1}$。

如果是给定四元数 q 旋转向量 $\mathbf{v}$，那么设 $q=[w,\mathbf{r}]$ 是单位四元数（即 $w^2+\|\mathbf{r}\|^2=1$），向量 $\mathbf{v}$ 的四元数形式为 $v=[0,\mathbf{v}]$。

证明：倒三角函数，利用叉乘展开式：$a\times b\times c=(a\cdot c)b-(a\cdot b)c$ 和 $w^2+\|\mathbf{r}\|^2=1$ 即可。

旋转组合：两个旋转 q_1 和 q_2 的组合等价于四元数的乘法：$q_2(q_1xq_1^*)q_2^*=(q_2q_1)x(q_1^*q_2^*)$，不满足交换律，满足结合律。

令单位四元数 $q=w+x\mathbf{i}+y\mathbf{j}+z\mathbf{k}=[w,(x,y,z)]$，则旋转矩阵 $R(q)$ 为 $R(q)=\begin{bmatrix}1-2y^2-2z^2&2xy-2zw&2xz+2yw\\2xy+2zw&1-2x^2-2z^2&2yz-2xw\\2xz-2yw&2yz+2xw&1-2x^2-2y^2\end{bmatrix}$。根据上一步结果，旋转矩阵 R 的迹满足：$\mathrm{tr}(R)=3-4(x^2+y^2+z^2)=4w^2-1$，所以：$w=\frac{\sqrt{\mathrm{tr}(R)+1}}{2}$ $x=\frac{R_{32}-R_{23}}{4w}$ $y=\frac{R_{13}-R_{31}}{4w}$ $z=\frac{R_{21}-R_{12}}{4w}$。其中 R_{ij} 表示矩阵 R 的第 i 行第 j 列的元素。这些公式在 $w\neq0$ 时有效。

在单位三维球面 S^3 上，或两个四元数 (q_1,q_2) 之间的角度 $\langle p,q\rangle=\arccos(p\cdot q)$。证明：设 $p=(p_w,\mathbf{p}_v)$ 和 $q=(q_w,\mathbf{q}_v)$，那么显然，从 p 旋转到 q 的相对旋转可以由四元数乘法 $\Delta q=qp^*$ 表示。$\Delta q=(q_wp_w+\mathbf{q}_v\cdot\mathbf{p}_v,\dots)$ 的实部是 $p\cdot q$。对应到旋转的距离就是乘个 2：$\mathrm{dist}(p,q)=2\min\{\langle p,q\rangle,\langle p,-q\rangle\}$。**要两个取最小值是因为双倍覆盖**。两个旋转 (R_1,R_2) 的距离与其对应四元数 $q(R_1)$ 和 $q(R_2)$ 在球面上的距离成线性关系（前者是后者的两倍）。

线性插值（Lerp）：$q(t)=(1-t)q_1+tq_2$。**归一化线性插值（Nlerp）**：$q(t)=\frac{(1-t)q_1+tq_2}{\|(1-t)q_1+tq_2\|}$（除个模长恢复为单位四元数）。以上两种插值都有问题，他们实际上是线性切分了弦长而不是弧长，会导致在转动时角速度不均匀。**球面线性插值（Slerp）**：$q(t)=\frac{\sin((1-t)\theta)}{\sin(\theta)}q_1+\frac{\sin(t\theta)}{\sin(\theta)}q_2$，其中 θ 是 q_1 和 q_2 之间的夹角，$\theta=\arccos(q_1\cdot q_2)$。证明：正弦定理。

球面均匀采样：：在 $\mathbb{SO}(3)$ 中均匀采样旋转矩阵等价于从单位四元数的集合 $\mathbb{S}(3)$ 中均匀采样。原因：两个旋转之间的距离与对应的四元数在单位球面上的距离成线性关系。均匀采样 $\mathbb{S}(3)$：从四维**标准正态分布** $\mathcal{N}(0,I_{4\times4})$ 中随机采样一个变量，并将其归一化，从而得到（直接解释为）单位四元数。原因：由于标准正态分布是各向同性的，所以采样得到的单位四元数在 $\mathbb{S}(3)$ 中也是均匀分布的。

- **旋转矩阵**：可逆、可组合（矩阵连乘）、但在 $\mathbb{SO}(3)$ 上移动不直接，**但最适合作为 NN 输出；连续性**。
- **欧拉角**：逆向复杂、组合复杂、因为 Gimbal lock 的存在，与 $\mathbb{SO}(3)$ 不能平滑映射
- **轴角**：可逆、组合复杂、大部分情况下可以与 $\mathbb{SO}(3)$ 平滑映射，但是在边界情况（如旋转 0 度时）不行
- **四元数**：可逆，可组合，平滑映射，但存在双倍覆盖的问题

碰撞检测：**球体包裹法（Bounding Spheres）**。缺点：**保守性导致可行解丢失**，限制了模型对于更精细物体的操作能力。（很小的面片，虚假自碰撞）

运动规划：**PRM 算法**：在 $\mathcal{C}_{\text{free}}$ 中随机 sample（不重），然后连接 k 近邻，剔除碰撞边（这一步用连线上线性采样来处理），然后用搜索或者 Dij 找路。**场景不变时可复用**。**高斯采样**：先随机 q_1，$\mathcal{N}(q_1,\sigma^2)$ 生成 q_2，如果 $q_1\in C_{\text{free}}$ 且 $q_2\notin C_{\text{free}}$，则添加 q_1。**有边界偏好，效率低**。**桥采样**：计算中点 q_3，当 q_1,q_2 都寄才要 q_3。*PRM 具有渐进最优性*。

RRT：exploration（随机采样）and exploitation（向着 goal 走）。探索参数 β、步长 ϵ 和采样点数量 n 都要调。**RRT-Connect**：双向 RRT、定向生长策略（扩展目标选择最近的另一树的叶子）、多种步长贪婪扩展。

Shortcutting：平滑路径。可以多次 RRT 后并行 shortcutting。

控制系统的性能评价：最小化稳态（Steady-State）误差；最小化调节时间，快速达到稳态；最小化稳态附近的振荡。**FeadForward 开环控制**：规划完闭眼做事无反馈纠错；**FeadBack 闭环控制**：带反馈回路，稳定。

期望状态：θ_d（destination），当前状态：θ，误差：$\theta_e=\theta_d-\theta$。

1. **稳态误差（Steady-State Error）**：系统到达稳态后的残余误差 $e_{ss}=\lim_{t\to\infty}\theta_e(t)$。理想系统应满足 $e_{ss}=0$
2. **调节时间（Settling Time）**：误差首次进入并保持在 $\pm2\%$ 误差带所需时间
3. **超调量（Overshoot）**：系统响应超过稳态值的程度，最开始过去不算 $\text{overshoot}=|a/b|\times100\%$，其中，$a$ 表示最大偏移量，b 表示最终稳态值

P 控制 Proportional：$P=K_p\theta_e(t)$。**一阶形式**：当控制信号改变状态的导数（即控制速度信号）时：$\dot\theta(t)=P=K_p\theta_e(t)$，若希望状态以 $\dot\theta_d(t)=c$ 移动，则 $\dot\theta_e(t)+K_p\theta_e(t)=c$，解 ODE 得到 $\theta_e(t)=\frac{c}{K_p}+\left(\theta_e(0)-\frac{c}{K_p}\right)e^{-K_pt}$。当 $c\neq0$（目标移动）时：随着 $t\to\infty$，$e^{-K_pt}\to0$ 稳态误差：$\lim_{t\to\infty}\theta_e(t)=\frac{c}{K_p}$。**系统存在永久稳态误差**，误差大小与目标速度 c 成正比，与比例增益 K_p 成反比，增大 K_p 可以减小稳态误差。**二阶形式**：控制信号改变状态的二阶导数（力或力矩信号）：$\ddot\theta(t)=P=K_p\theta_e(t)$，导致状态振荡且不稳定。

PI 控制 Integral：$PI=K_p\theta_e(t)+K_i\int_0^t\theta_e(\tau)\mathrm{d}\tau$。如果控制信号作用于状态导数（速度）：$\dot\theta(t)=PI=K_p\theta_e(t)+K_i\int_0^t\theta_e(\tau)\mathrm{d}\tau$，$\dot\theta_e(t)=\dot\theta_d(t)-\dot\theta(t),\dot\theta_d(t)=\dot\theta_e(t)+\dot\theta(t)$，两边求导：$\ddot\theta_d(t)=\ddot\theta_e(t)+K_p\dot\theta_e(t)+K_i\theta_e(t)$，如果 $\ddot\theta_d(t)=0$（目标加速度为零），$\ddot\theta_e(t)+K_p\dot\theta_e(t)+K_i\theta_e(t)=0$，为二阶常系数齐次微分方程。解的形式由方程特征根决定，特征方程为：$r^2+K_pr+K_i=0$。

其解的形式决定系统的阻尼特性：**过阻尼（Overdamped）**：两个实根，系统缓慢收敛。**临界阻尼（Critically damped）**：双重实根，快速无振荡收敛。**欠阻尼（Underdamped）**：共轭复根，系统振荡收敛。

PD 控制 Derivative：$PD=K_p\theta_e(t)+K_d\frac{\mathrm{d}}{\mathrm{d}t}\theta_e(t)$。如果 $\ddot\theta_d(t)=0$（目标加速度为零），则 $\ddot\theta_e(t)+K_d\dot\theta_e(t)+K_p\theta_e(t)=0$，解的形式由方程特征根决定，特征方程为：$r^2+K_dr+K_p=0$。

PID 控制：$PID=K_p\theta_e(t)+K_i\int_0^t\theta_e(\tau)\mathrm{d}\tau+K_d\frac{\mathrm{d}}{\mathrm{d}t}\theta_e(t)$。

K_p **控制当前状态**：K_p 增大可 **加快响应速度（Rise Time）**、减少调节时间，因为快速减少 $\theta_e(t)$；**增大超调量**；**单用会产生稳态误差**。

K_i **控制历史累积**：对持续误差进行累积补偿，**消除稳态误差**；**增大超调量**。

K_d **预测未来趋势**：减少调节时间，**抑制超调和振荡**；当误差增加时提供更强的控制作用；当误差减小时提供更温和的控制作用。# Grasp

DexGraspNet：合成数据（Synthetic Data）+ 深度学习

1. 场景理解：预测每个点 **抓取可能性（Graspness）**，是否是 **物体（Objectness）**
2. 局部特征：不用全局特征（关联性弱、泛化性差），选择 Graspness 高的地方附近的点云，提取局部特征（**几何信息**）
3. 条件抓取生成模块：条件生成处理 (T,R) 多峰分布，然后采样后直接预测手指形态 θ

仅处理包覆式抓取（Power Grasp），没处理指尖抓取（Precision Grasp）；主要使用力封闭抓取；透明（Transparent）或高反光（Highly Specular/Shiny）物体有折射（Refraction）/ 镜面反射（Specular Reflection），导致点云质量差。

ASGrasp：深度修复，合成数据 + 监督学习。域随机化、多模态立体视觉、立体匹配（Stereo Matching）。

Affordance：指一个物体所能支持或提供的交互方式或操作可能性，哪个区域、何种方式进行交互。

Where2Act：大量随机尝试 + 标注。学习从视觉输入预测交互点 a_p、交互方向 $R_{z|p}$ 和成功置信度 $s_{R|p}$。**VAT-Mart**：预测一整条操作轨迹。

利用视觉输入进行预测：

- **物体位姿（Object Pose）**：需要模型、抓取标注。
- **抓取位姿（Grasp Pose）**：直接预测抓取点和姿态，无模型或预定义抓取。
- **可供性（Affordance）**

启发式（Heuristic）规则：预抓取 Pre-grasp，到附近安全位置再闭合，避免碰撞

1. **操作复杂度有限**：难以处理复杂任务，受启发式规则设计限制。
2. **开环执行（Open-loop）**：规划一次，执行到底，闭眼做事。高频重规划可近似闭环。

2 Policy

策略学习：学习 $\pi(a_t|s_t)$ 或 $\pi(a_t|o_t)$，实现 **闭环控制**。

BC：将 $D=\{(s_i,a_i^*)\}$ 视为监督学习任务，学习 $\pi_\theta(s)\approx a^*$。

Distribution shift：策略 π_θ 错误累积，访问训练数据中未见过的状态（$p_\pi(s)$ 与 $p_{\text{data}}(s)$ 不匹配），策略失效。

1. **改变 $p_{\text{data}}(o_t)$：Dataset Aggregation（DAgger）**
   训练 $\pi_i\Rightarrow$ 用 π_i **执行（Rollout）** 收集新状态 $\Rightarrow$ 查询专家在此状态下的 $a^*\Rightarrow D\leftarrow D\cup\{(s,a^*)\}\Rightarrow$ 重新训练 π_{i+1}。但是出错才标注，也会影响准确性。
2. **改变 $p_\pi(o_t)$（更好拟合）**：从（传统算法）最优解中获取；从教师策略中学习（有 **Privileged Knowledge** ）

遥操作数据（Teleoperation）：贵，也存在泛化问题。

非马尔可夫性：引入历史信息，但可能过拟合，因果混淆（Causal Confusion）。

目标条件化（Goal-conditioned）：$\pi(a|s,g)$，共享数据和知识。但 g 也有分布偏移问题。

IL 局限性：依赖专家数据、无法超越专家、不适用于需要精确反馈的高度动态 / 不稳定任务。

Offline Learning：固定数据集学习，无交互。

Online Learning：边交互边学习。

策略梯度定理：$\nabla_\theta J(\theta)\approx\frac{1}{N}\sum_{i=1}^N\nabla_\theta\log p_\theta(\tau^{(i)})R(\tau^{(i)})$
$\nabla_\theta\log p_\theta(\tau)=\sum_{t=0}^{T-1}\nabla_\theta\log\pi_\theta(a_t|s_t)$，奖励函数无需可导。

环境模型：包括状态转移概率 $p(s_{t+1}|s_t,a_t)$ 和奖励函数 $r(s_t,a_t)$

- **Model-Free**：不需要知道环境的模型
- **Model-Based**：利用神经网络学习环境的模型

REINFORCE：按照整条轨迹的总回报 $R(\tau^{(i)})$ 加权，**On-Policy**。BC 是平权。

On-Policy：数据来自当前策略。效果好，**样本效率低**，每次都得重新采样。
Off-Policy：数据可来自不同策略。**样本效率高**，可能不稳定。

Reward-to-Go：降方差，用未来回报 $\hat Q(s_t,a_t)=\sum_{t'=t}^T r_{t'}$ 加权梯度。

Baseline：降方差，减去 a_t 无关状态基线 $b(s_t)$，$\hat Q(s_t,a_t)-b(s_t)$。梯度无偏。

Advantage $A^{\pi_\theta}(s_t,a_t)=Q^{\pi_\theta}(s_t,a_t)-V^{\pi_\theta}(s_t)$：动作相对平均的优势，可替换 $R(\tau^{(i)})$ 做权值，$\hat A(s_t,a_t)=r(s_t,a_t)+\gamma\hat V(s_{t+1})-\hat V(s_t)$

Bootstrap：使用基于当前函数估计的值 $\hat V_\phi^\pi(s_{i,t+1})$ 来更新 **同一个函数** 在另一个点 $s_{i,t}$ 的估计 $\hat V_\phi^\pi(s_{i,t})$

Batch AC：收集一批完整轨迹或转换数据后，统一更新 A/C。梯度估计更稳定，但更新频率低。

Online AC：每一步交互（或极小批量）后，立即更新 A/C。更新快，数据利用率高，但梯度估计方差较大。

Parallelization：多 worker 采样，提速增稳，异步快。