

机器人学简介

东林钟声*

武汉，2022 年 11 月

目录

1 绪论	5
2 结构设计	5
3 关节模块	6
4 运动学	6
4.1 正运动学	6
4.1.1 旋转矩阵	6
4.1.2 齐次变换矩阵	7
4.1.3 D-H 参数	8
4.1.4 DROP-A6 D-H 参数	8
4.1.5 雅克比矩阵	10
4.2 逆运动学	11
4.2.1 解析解: Pieper 准则	11
4.2.2 数值解: Newton Raphson 迭代	12
5 轨迹规划	12
5.1 三次多项式拟合	12
5.2 关节速度控制器设计	14
6 动力学	14
6.1 动力学辨识	15
6.1.1 最小惯性参数集	15
6.1.2 最小二乘	15
6.2 计算力矩控制	16
6.2.1 零力控制	16
6.2.2 轨迹跟踪	16
6.3 碰撞检测	16
6.3.1 直接力矩检测	17
6.3.2 动量积分观测器	17
6.4 力控制	18
6.4.1 直接力控制	18

6.4.2 力位混合控制	19
7 视觉感知	20
7.1 2D 感知	21
7.2 3D 感知	21

符号表

- n : 机器人自由度数目
- m : 最小惯性参数数目
- $q \in \mathbb{R}^{n \times 1}$: 关节位置
- $\dot{q} \in \mathbb{R}^{n \times 1}$: 关节速度
- $\ddot{q} \in \mathbb{R}^{n \times 1}$: 关节加速度
- $\tau \in \mathbb{R}^{n \times 1}$: 关节力矩
- $\mathcal{F} \in \mathbb{R}^{6 \times 1}$: 机器人施加给其本身末端的广义力、力矩
- $\mathcal{F}_{ext} \in \mathbb{R}^{6 \times 1}$: 外部环境 (External) 加给机器人末端的广义力、力矩
- $X = [x, y, z, \alpha, \beta, \gamma] \in \mathbb{R}^{6 \times 1}$: 任务空间位姿
- $J(q) \in \mathbb{R}^{6 \times n}$: 雅克比矩阵
- $R \in \mathbb{R}^{3 \times 3}$: 旋转变换矩阵
- $T \in \mathbb{R}^{4 \times 4}$: 齐次变换矩阵
- $T_r : q \in \mathbb{R}^{6 \times 1} \rightarrow X \in \mathbb{R}^{6 \times 1}$: 正向运动学
- $M(q) \in \mathbb{R}^{n \times n}$: 广义质量矩阵
- $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$: 柯氏力和摩擦力矩阵
- $G(q) \in \mathbb{R}^{n \times 1}$: 重力矩阵
- $A(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n \times m}$: 最小惯性参数集回归矩阵
- $\Theta \in \mathbb{R}^{m \times 1}$: 最小惯性参数集
- $t \in \mathbb{R}^+$: 连续时间
- $t_k \in \mathbb{R}^+$: 离散时刻
- $P(t) \in \mathbb{R}^{6 \times 1}$: 任务空间轨迹位姿

1 绪论

本文主要从结构设计、关节模块、运动学、轨迹规划、动力学以及视觉感知这 6 个方面对机器人硬件到算法进行简要的介绍。

2 结构设计

开始设计一款多关节机器人首先要做的是确定自由度的个数以及其关节的具体空间位置以及转动方向。以 Universal Robots 公司推出的 UR5 为例，其具有 6 个可动关节，能够实现任务空间中末端 6 自由度的移动，如图 1 所示。

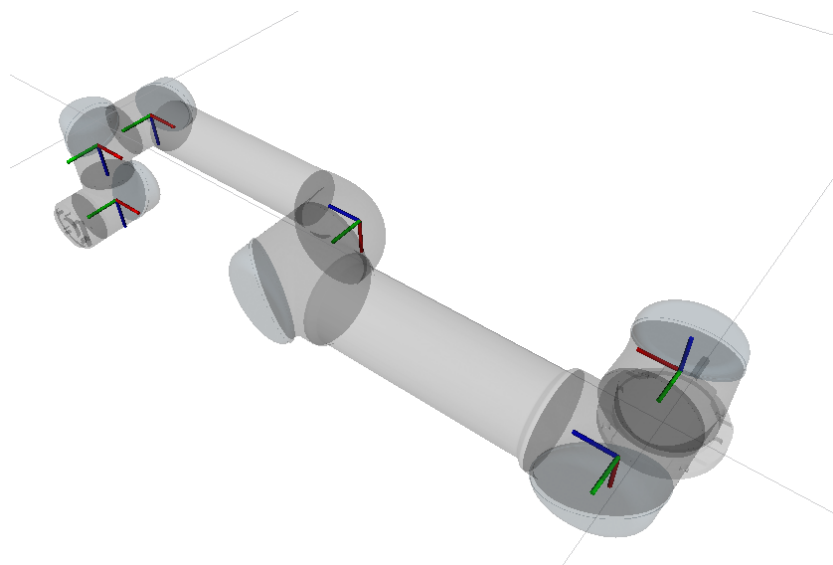


图 1: 6 轴 UR5 机器人

为了验证所设计的结构是否能够满足要求，需要将设计的机器人装配文件通过 [sw2urdf¹](http://wiki.ros.org/sw_urdf_exporter) 转换成 URDF 文件。然后可以基于 [PyBullet²](https://pybullet.org/wordpress/) 仿真环境对 URDF 文件进行仿真验证。

¹http://wiki.ros.org/sw_urdf_exporter

²<https://pybullet.org/wordpress/>

3 关节模块

机器人中关节模块是最为重要的硬件之一，其能够驱动连杆进行运动从而实现机器人在任务空间中的作业。在机器人中关节模块一般分为电机、减速器、编码器、抱闸以及驱动器来构成。减速器主要目的为提高电机的输出力矩，同时提高电机的位置控制精度。编码器主要为电机的位置、速度等基本信息提供反馈。抱闸用来实现断电自锁，这对于工业机器人来说是非常重要的功能。一般关节电机需要驱动器来实现位置、速度以及电流这三环的闭环控制。开源方案可以采用 SimpleFOC³来进行实现。由于机器人一般具有多个关节且需要实现高频率的控制，所以通信协议一般采用 CAN 总线或者 EtherCAT 总线来进行通信控制。其中 CAN 总线的最高波特率为 1M bps，如果想要实现 1KHz 以上的高频闭环控制，需要采用最高波特率可以到达 100M bps 的 EtherCAT 总线。

4 运动学

4.1 正运动学

机器人正运动学主要为描述从关节空间到任务空间的一种坐标变换 $T_r : q \in \mathbb{R}^{n \times 1} \rightarrow X \in \mathbb{R}^{6 \times 1}$ 。简而言之，即为获取机器人各关节角度 q 从而计算出机器人末端在任务空间中的笛卡尔坐标 X 。

4.1.1 旋转矩阵

旋转矩阵 $R \in \mathbb{R}^{3 \times 3}$ 用来表示欧式空间中的旋转变换。通过欧拉角可以写出旋转矩阵为：

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

其中 α, β, γ 分别表示欧拉角中的 roll（横滚）、pitch（俯仰）、yaw（偏航）。旋转矩阵的性质：

³<https://simplefoc.com/>

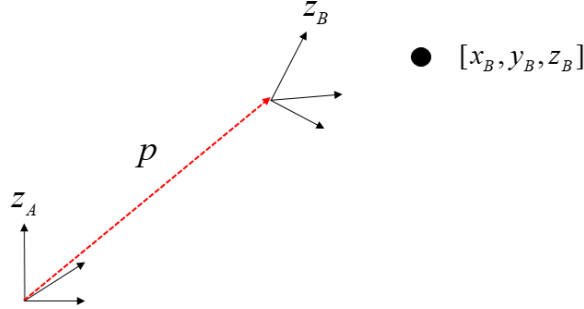


图 2: 齐次变换矩阵坐标变换

- $R^T R = I$, 即 $R^T = R^{-1}$
- $\det R = \pm 1$

4.1.2 齐次变换矩阵

旋转矩阵只表述了原点重合的两个坐标系之间的相对变化关系，没有涵盖原点之间的位移。所以需要齐次变换矩阵 $T \in \mathbb{R}^{4 \times 4}$ 来同时描述笛卡尔坐标系下 6 自由度的刚体位置变换关系。齐次变换矩阵的形式如下：

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}, \quad (1)$$

其中 $R \in \mathbb{R}^{3 \times 3}$ 表示旋转矩阵， $p \in \mathbb{R}^{3 \times 1}$ 表示原点之间的平移向量。齐次变换矩阵可以通过矩阵运算来快速计算坐标系之间的变换，如图 2 所示。记目标点在坐标系 B 下的坐标表示为 $[x_B, y_B, z_B]$ ，那么该点在坐标系 A 下的坐标表示为：

$$[x_A, y_A, z_A, 1]^T = {}^B_A T \times [x_B, y_B, z_B, 1]^T. \quad (2)$$

齐次坐标变换矩阵的几个性质如下：

- $T = \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix}$
- ${}^B_A T = {}^A_B T^{-1}$

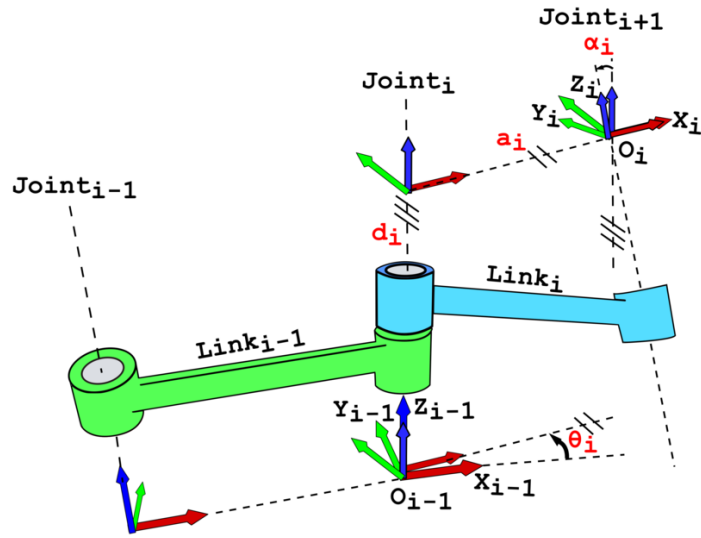


图 3: D-H 参数示意图

4.1.3 D-H 参数

D-H 参数用来描述机器人连杆之间坐标系的变换，如图 3 所示。D-H 参数连杆坐标系的建立方法为：1) z 轴指向关节旋转轴线；2) 基座的 x 轴可以自由选择，其他连杆坐标系的 x 轴与两个连续 z 轴的公垂线共线；3) y 轴由右手定则确定。

在建立好了 D-H 坐标系之后，即可以确认 4 个参数来表示连杆之间的变化：

- d_i : 沿着 z_{i-1} 轴线方向，从原点 o_{i-1} 到 z_{i-1}, z_i 轴的公垂线的距离
- θ_i : 绕着 z_{i-1} 轴，将 x_{i-1} 旋转到 x_i 的角度
- a_i : 沿着 x_i 轴，从 z_{i-1} 移动至 z_i 的距离
- α_i : 绕着 x_i 轴，将 z_{i-1} 旋转到 z_i 的角度

4.1.4 DROP-A6 D-H 参数

自研的 6 轴 DROP-A6 机器人的示意图如图 4 所示，图中表明了相应 D-H 参数，具体如表 1 所示。

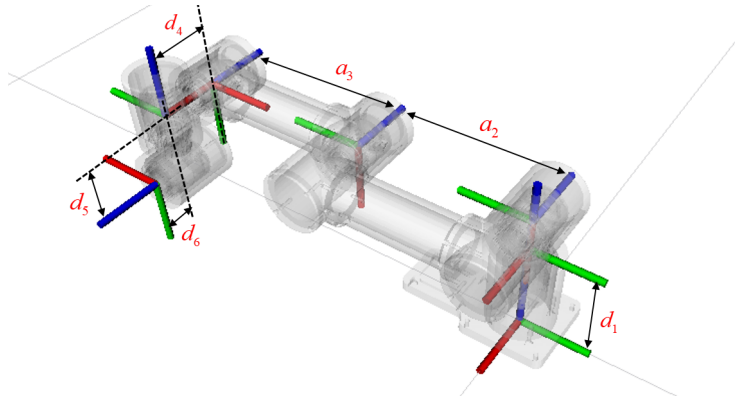


图 4: 自研 DROP-A6 机器人

关节	θ	a (mm)	d (mm)	α
1	0	0	178	$\frac{\pi}{2}$
2	0	-250	0	0
3	0	-250	0	0
4	0	0	88	$\frac{\pi}{2}$
5	0	0	88	$-\frac{\pi}{2}$
6	0	0	43	0

表 1: DROP-A6 D-H 参数表

根据 D-H 参数可以得到相邻两个连杆之间的坐标系齐次变换矩阵 ${}^i{}^{-1}T \in \mathbb{R}^{4 \times 4}$ ，其具体可以表达为：

$${}^i{}^{-1}T = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

所以可以求得 DROP 的正向运动学 6T_0 的具体表达式：

$${}^6T_0 = {}^0T_1 \times {}^1T_2 \times {}^2T_3 \times {}^3T_4 \times {}^4T_5 \times {}^5T_6 \quad (4)$$

由于 ${}^6T_0 \in \mathbb{R}^{4 \times 4}$ 包含了旋转矩阵 R 以及位移向量 p ，所以可以轻易得到任务空间位姿 $X = [x, y, z, \alpha, \beta, \gamma] \in \mathbb{R}^{6 \times 1}$ 。同时由于 D-H 参数中的 θ_i 表示关节轴旋转角，即为 q_i 。所以正向运动学可以记为 $T_r : q \in \mathbb{R}^{n \times 1} \rightarrow X \in \mathbb{R}^{6 \times 1}$ ，其可以表达为：

$$X = T_r(q). \quad (5)$$

4.1.5 雅克比矩阵

雅克比矩阵是机器人学中非常重要的概念，其不仅表示了关节空间和任务空间之间的速度变换关系，也能够进行关节力矩与末端广义力之间的转换。

对公式 (5) 两边对时间 t 求导，得到如下式子：

$$\dot{X} = J(q)\dot{q}, \quad (6)$$

其中 $J(q) \in \mathbb{R}^{6 \times n}$ 表示雅克比矩阵，其表明了关节空间关节速度与任务空间末端速度之间的转换关系。

下面基于虚功原理来推导关节力矩和末端广义力之间的转换关系。关节力矩 τ 会驱使机器人产生运动，等同于在机器人末端施加了一个广义力（力矩） $\mathcal{F} \in \mathbb{R}^{6 \times 1}$ 来驱动末端运动。在关节坐标系下的功率应该和在任务空间下的功率相同，即：

$$\begin{aligned} \tau^T \dot{q} &= \mathcal{F}^T \dot{X} \\ &= \mathcal{F}^T J(q)\dot{q}. \end{aligned} \quad (7)$$

所以可以得到：

$$\tau = J^T(q)\mathcal{F}. \quad (8)$$

其表示关节力矩到末端广义力的转换。

4.2 逆运动学

逆运动学是机器人技术中非常关键的一个环节，其主要为描述从任务空间到关节空间的一种坐标变换 $T_r^{-1} : X \in \mathbb{R}^{6 \times 1} \rightarrow q \in \mathbb{R}^{n \times 1}$. 逆运动学直接告诉了机器人如果要将末端运动到任务空间中的某一指定位姿 X , 则需要将关节位置运动到 $q = T_r^{-1}(X)$ 。

4.2.1 解析解: Pieper 准则

当 6 轴机器人的关节满足 Pieper 准则时, 可以得到逆运动学的解析解形式。Pieper 准则为如下两条:

- 三个连续的转动关节的轴相交于同一点
- 三个连续的转动关节的轴平行

只要满足其中一条, 就可以进行逆运动学求解。可以从图 4 中可以看出, DROP-A6 满足 Pieper 准则中三个连续的转动关节 (2, 3, 4) 的轴线平行。

DROP-A6 的逆运动学解析解一共有 8 组, 如图 5 所示。

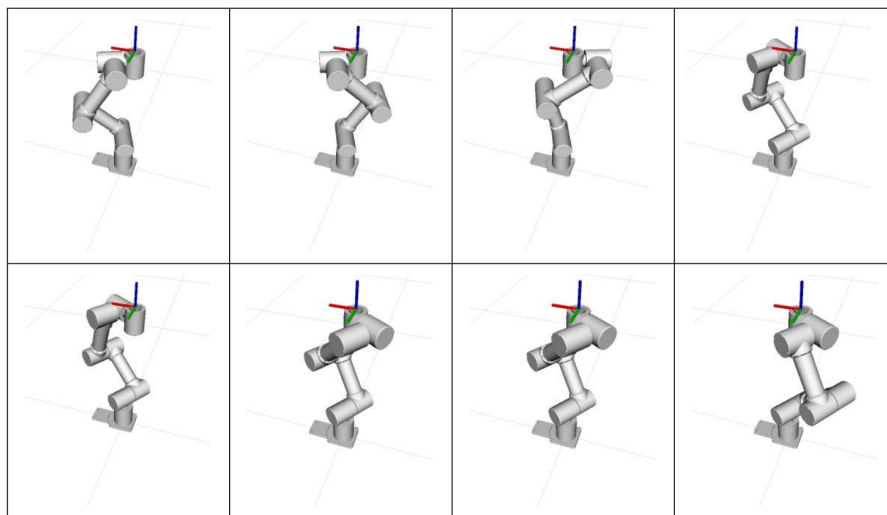


图 5: DROP-A6 逆运动学 8 组解析解

4.2.2 数值解：Newton Raphson 迭代

不是所有的机器人构型都能得到解析形式的逆运动学表达，所以需要基于数值形式的逆运动学迭代求解形式。

记目标函数为：

$$g(q) = X - Tr(q). \quad (9)$$

逆运动学数值解的目的是找到目标函数 $g(q)$ 的零点。然后将 $g(q_{k+1})$ 在上一个时刻的关节位置 q_k 进行一阶泰勒展开，得到：

$$g(q_{k+1}) = X - Tr(q_k) - J(q_k)(q_{k+1} - q_k) + \mathcal{O}(q_{k+1} - q_k). \quad (10)$$

忽略掉高阶小量，令 $g(q_{k+1}) = 0$ 可以得到：

$$q_{k+1} = q_k + J^\dagger(q_k)(X - Tr(q_k)), \quad (11)$$

其中 $J^\dagger(q_k)$ 表示雅克比矩阵的加号广义逆。那么数值迭代求解逆运动学的过程如下：

1. 给定任务空间位姿 X 、初始关节位置 q_0 以及数值精度 ϵ ，令 $k = 0$ ；
2. $q_{k+1} = q_k + J^\dagger(q_k)(X - Tr(q_k))$, $k = k + 1$ ；
3. 误差 $e = \|X - Tr(q_k)\|_2^2$ ，如果 $e \leq \epsilon$ 则输出结果 q_k ，否则跳转 2。

5 轨迹规划

机器人的运动控制除了最基本的点到点的模式以外，另一个重要的运动模式为按照一段给定的轨迹进行运动，同时要满足经过某些指定点并达到某一速度（比如起点终点速度为 0）。为了完成这一目标需要对机器人实现轨迹规划，本节仅介绍最基本的三次多项式拟合生成轨迹的方式以方便读者更加直观地理解。

5.1 三次多项式拟合

规划的轨迹如图 6 所示，记 $P(t_k), \dot{P}(t_k) \in \mathbb{R}^{6 \times 1}$ 表示机器人末端需要经过的 WayPoint 的笛卡尔坐标系位置（姿态）、速度（角速度），即机器人末端需要在 t_k 时刻达到目标位置 $P(t_k)$ 并且具有速度 $\dot{P}(t_k)$ 。

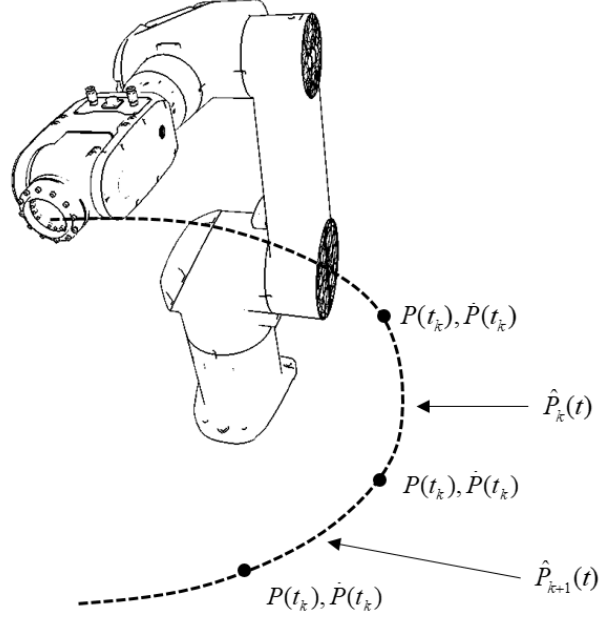


图 6: 轨迹规划

假设有 N 个需要经过的 WayPoint, 则需要拟合 $N - 1$ 段轨迹, 记 $P(t_k), \dot{P}(t_k)$ 到 $P(t_{k+1}), \dot{P}(t_{k+1})$ 之前的这一段拟合轨迹为 $\hat{P}_k(t)$ 。因为多项式函数为光滑函数, 具有平滑无突变的性质, 非常适用于机器人运动的目标轨迹。所以可以采用 3 次多项式的形式来对轨迹进行拟合, $\hat{P}_k(t)$ 的具体形式如下:

$$\hat{P}_k(t) = a_k t^3 + b_k t^2 + c_k t + d_k, \quad (12)$$

其中 a_k, b_k, c_k, d_k 表示 4 个未知参数, 而这段轨迹需要满足起点位置、速度以及终点位置、速度正好 4 个约束, 所以可以求得 4 个未知参数的唯一解。4 个约束如下:

$$\begin{cases} \hat{P}_k(t_k) = a_k t_k^3 + b_k t_k^2 + c_k t_k + d_k = P(t_k) \\ \hat{P}_k(t_{k+1}) = a_k t_{k+1}^3 + b_k t_{k+1}^2 + c_k t_{k+1} + d_k = P(t_{k+1}) \\ \dot{\hat{P}}_k(t_k) = 3a_k t_k^2 + 2b_k t_k + c_k = \dot{P}(t_k) \\ \dot{\hat{P}}_k(t_{k+1}) = 3a_k t_{k+1}^2 + 2b_k t_{k+1} + c_k = \dot{P}(t_{k+1}) \end{cases} \quad (13)$$

为了方便线性方程组的求解，可以写成如下的矩阵形式：

$$\begin{bmatrix} t_k^3 & t_k^2 & t_k & 1 \\ t_{k+1}^3 & t_{k+1}^2 & t_{k+1} & 1 \\ 3t_k^2 & 2t_k & 1 & 0 \\ 3t_{k+1}^2 & 2t_{k+1} & 1 & 0 \end{bmatrix} \times \begin{bmatrix} a_k \\ b_k \\ c_k \\ d_k \end{bmatrix} = \begin{bmatrix} P(t_k) \\ P(t_{k+1}) \\ \dot{P}(t_k) \\ \dot{P}(t_{k+1}) \end{bmatrix} \quad (14)$$

那么 a_k, b_k, c_k, d_k 可以求解为：

$$\begin{bmatrix} a_k \\ b_k \\ c_k \\ d_k \end{bmatrix} = \begin{bmatrix} t_k^3 & t_k^2 & t_k & 1 \\ t_{k+1}^3 & t_{k+1}^2 & t_{k+1} & 1 \\ 3t_k^2 & 2t_k & 1 & 0 \\ 3t_{k+1}^2 & 2t_{k+1} & 1 & 0 \end{bmatrix}^{-1} \times \begin{bmatrix} P(t_k) \\ P(t_{k+1}) \\ \dot{P}(t_k) \\ \dot{P}(t_{k+1}) \end{bmatrix} \quad (15)$$

对 $N - 1$ 段拟合轨迹求解完毕之后，将各段轨迹依次拼接起来生成最终机器人末端要走的轨迹位置、速度的解析解如下：

$$\begin{cases} \hat{P}(t) = \hat{P}_k(t_k) & \text{if } t_k \leq t < t_{k+1} \\ \hat{P}(t) = \hat{P}_k(t_k) & \text{if } t_k \leq t < t_{k+1} \end{cases} \quad (16)$$

5.2 关节速度控制器设计

为了能够同时控制机器人末端的位置和速度，使用速度前馈加位置反馈来设计机器人关节速度控制器（为了记号方便，省略了时间 t ），如下：

$$\dot{q} = J^{-1}(q)\dot{\hat{P}} + K_p \underbrace{\{T_r^{-1}(\hat{P}) - q\}}_{\text{compensation}}, \quad (17)$$

其中 $J^{-1}(\bullet)$ 表示雅克比矩阵的逆， $T_r^{-1}(\bullet)$ 表示逆运动学， K_p 表示位置误差补偿系数。

6 动力学

机器人的动力学主要描述了关节力矩如何影响了关节加速度从而驱使机器人产生运动。机器人的动力学方程可以写为：

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q), \quad (18)$$

其中 $M(q) \in \mathbb{R}^{n \times n}$ 表示广义质量矩阵， $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ 表示柯氏力以及摩擦力， $G(q) \in \mathbb{R}^{n \times 1}$ 表示重力矩阵。

公式 (18) 中的动力学方程会满足如下两个重要性质：

- $M(q) = M^T(q)$
- $\dot{M}(q) = C(q, \dot{q}) + C^T(q, \dot{q})$

6.1 动力学辨识

6.1.1 最小惯性参数集

在机器人中，一个连杆会含有质心位置 $[x_{com}, y_{com}, z_{com}]$ 、质量 m 以及转动惯量矩阵 $\begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$ 一共 10 个参数来进行惯性参数的表达。但是在机器人的运动中，并不是每个连杆的 10 个惯性参数都会影响运动，所以一种机器人结构设计之后，会有一组能够表示最小惯性参数的参数集。

基于最小惯性参数，机器人的动力学方程可以重新写为：

$$\tau = A(q, \dot{q}, \ddot{q}) \times \Theta, \quad (19)$$

其中 $A \in \mathbb{R}^{n \times m}$, $\Theta \in \mathbb{R}^{m \times 1}$ 分别表示最小惯性参数集对应的回归矩阵以及最小惯性参数集。回归矩阵 A 可以通过 Symoro⁴来进行计算。不过 Symoro 需要手动输入 Modified D-H 参数以及连杆惯性参数等，较为不方便。可以采用笔者开发的 RoboticsInOne⁵从 URDF 文件进行自动化调用 Symoro。

6.1.2 最小二乘

为了辨识公式 (19) 中的最小惯性参数 Θ ，可以让机器人运动一段特定的轨迹，然后采集关节位置、速度、加速度以及电流（正比与力矩， $I = K\tau$ ）的 N 个时间序列数据点来进行最小二乘辨识。

由于时间序列的引入，这里将关节位置、速度、加速度以及电流按照时间进行叠加扩维后依次得到 $q_N, \dot{q}_N, \ddot{q}_N, \tau_N \in \mathbb{R}^{n \times N}$ ，然后也对回归矩阵进行扩维后得到 $A_N(q_N, \dot{q}_N, \ddot{q}_N) \in \mathbb{R}^{n \times (m \times N)}$ ，那么最小惯性参数集的辨识结果 $\hat{\Theta}$ 可以通过以下式子进行最小二乘求解：

$$\hat{\Theta} = A_N^\dagger(q_N, \dot{q}_N, \ddot{q}_N) \times \tau_N, \quad (20)$$

其中 \dagger 表示矩阵的加号广义逆。

⁴<https://github.com/symoro/symoro>

⁵<https://github.com/yunlongdong/RoboticsInOne>

6.2 计算力矩控制

在有了辨识之后的动力学矩阵 $\hat{M}(q), \hat{C}(q, \dot{q}), \hat{G}(q)$ 之后, 可以通过计算力矩控制来根据不同的任务要求来计算出相应的关节力矩来进行控制。

6.2.1 零力控制

零力控制是实现无额外力矩传感器的机器人拖动示教的核心技术。其主要实现思路为计算出相应的关节力矩能够平衡掉机器人的重力项 $G(q)$ 以及 $C(q, \dot{q})\dot{q}$ 。即令关节力矩 $\tau = \hat{G}(q) + \hat{C}(q, \dot{q})\dot{q}$, 此时机械臂能够随外力进行任意移动, 并且能够在移动停止后始终固定在同一位置。其控制框图如图 7 所示。

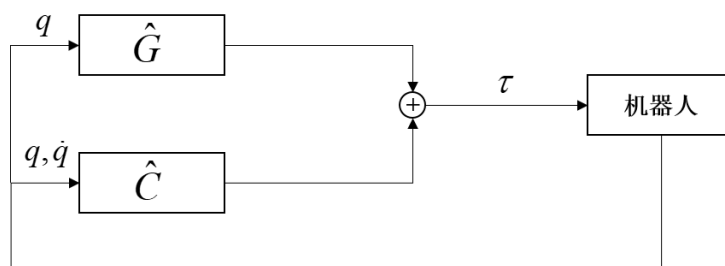


图 7: 零力控制

6.2.2 轨迹跟踪

基于计算力矩的轨迹跟踪动力学前馈 PD 反馈控制器设计如下:

$$\tau = \underbrace{\hat{M}(q)\ddot{q}_d + \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q)}_{\text{feedforward}} + \underbrace{K_p(q_d - q) + K_d(\dot{q}_d - \dot{q})}_{\text{PD feedback}}, \quad (21)$$

其中 $q_d, \dot{q}_d, \ddot{q}_d$ 分别表示关节目标位置、速度、加速度。按照该控制器进行力矩控制, 最终关节实际位置、速度、加速度 q, \dot{q}, \ddot{q} 会收敛到 $q_d, \dot{q}_d, \ddot{q}_d$ 。

6.3 碰撞检测

考虑外部环境接触的机器人动力学方程如下:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + J^T(q)\mathcal{F}_{ext}, \quad (22)$$

其中 $J^T(q) \in \mathbb{R}^{n \times 6}$ 表示雅克比矩阵的转置，用来将末端环境力转换到关节受力矩，如公式 (32) 所示。碰撞检测的主要目的即为实现对外部环境接触力 $\mathcal{F}_{ext} \in \mathbb{R}^{6 \times 1}$ 的准确且快速的估计。为了符号记号方便，记 $\mathcal{F}'_{ext} \triangleq J^T(q)\mathcal{F}_{ext}$ 。

6.3.1 直接力矩检测

直接力矩检测方法非常直观，其通过判断采集的关节位置、速度、加速度以及力矩是否满足辨识的动力学方程来进行 \mathcal{F}'_{ext} 的估计，如下：

$$\hat{\mathcal{F}}'_{ext} = \tau - \hat{M}(q)\ddot{q} - \hat{C}(q, \dot{q})\dot{q} - \hat{G}(q). \quad (23)$$

其中 $\hat{\mathcal{F}}'_{ext}$ 表示对外部接触力 \mathcal{F}'_{ext} 的估计值。

在有了外部接触力的估计值之后，可以基于阈值来构建碰撞检测二分类器 $\text{Collision}(q, \dot{q}, \ddot{q}, \tau)$ 如下：

$$\text{Collision}(q, \dot{q}, \ddot{q}, \tau) \triangleq \begin{cases} 1, & \text{if } \|\hat{\mathcal{F}}'_{ext}\|_2^2 \geq \eta \\ 0, & \text{else} \end{cases}, \quad (24)$$

其中 $\eta \in \mathbb{R}^+$ 表示碰撞检测阈值，其值越小越灵敏。

6.3.2 动量积分观测器

定义机器人的广义动量为 $M(q)\dot{q}$ ，然后根据 $\dot{M}(q) = C(q, \dot{q}) + C^T(q, \dot{q})$ ，则有：

$$\begin{aligned} \dot{p} &= \dot{M}(q)\dot{q} + M(q)\ddot{q} \\ &= C(q, \dot{q})\dot{q} + C^T(q, \dot{q})\dot{q} + \tau - M(q)\ddot{q} - C(q, \dot{q})\dot{q} - G(q) - \mathcal{F}'_{ext} \\ &= \tau + C^T(q, \dot{q})\dot{q} - G(q) - \mathcal{F}'_{ext} \end{aligned}$$

记 \mathcal{F}'_{ext} 的估计值为 r ，那么 \dot{p} 的观测值为：

$$\hat{\dot{p}} = \tau + \hat{C}^T(q, \dot{q})\dot{q} - \hat{G}(q) - r \quad (26)$$

如果让 r 按照如下动力学进行演变：

$$\dot{r} = -K(\hat{p} - \dot{p}) \quad (27)$$

如果动力学的辨识完全准确，则有：

$$\dot{r} = K(\mathcal{F}'_{ext} - r) \quad (28)$$

当 K 为对角正定矩阵时，观测值 r 会收敛到 \mathcal{F}'_{ext} 。

r 的具体计算过程如下：

$$\begin{aligned}
r(t) &= r(0) + \int_0^t K(\dot{\hat{p}} - \dot{p})dt \\
&= r(0) - Kp + K \int_0^t \{\tau + \hat{C}^T(q, \dot{q})\dot{q} - \hat{G}(q) - r\}dt \\
&= r(0) - K\hat{M}(q)\dot{q} + K \int_0^t \{\tau + \hat{C}^T(q, \dot{q})\dot{q} - \hat{G}(q) - r\}dt
\end{aligned} \tag{29}$$

上述方程为连续形式，左右两边都有 r ，并不好求解。实际过程可以离散化时间 t 来进行迭代求解，如下：

$$\begin{aligned}
r_{k+1} &= r_0 - K\hat{M}(q_{k+1})\dot{q}_{k+1} + K \sum_{i=0}^{k+1} \{\tau_i + \hat{C}^T(q_i, \dot{q}_i)\dot{q}_i - \hat{G}(q_i) - r_i\}\Delta t \\
&= r_k + K\hat{M}(q_k)\dot{q}_k - K\hat{M}(q_{k+1})\dot{q}_{k+1} \\
&\quad + K\{\tau_k + \hat{C}^T(q_{k+1}, \dot{q}_{k+1})\dot{q}_{k+1} - \hat{G}(q_{k+1}) - r_{k+1}\}\Delta t
\end{aligned} \tag{30}$$

因此递推形式为：

$$\begin{aligned}
r_{k+1} &= \\
&\frac{r_k + K\hat{M}(q_k)\dot{q}_k - K\hat{M}(q_{k+1})\dot{q}_{k+1} + K\{\tau_k + \hat{C}^T(q_{k+1}, \dot{q}_{k+1})\dot{q}_{k+1} - \hat{G}(q_{k+1})\}\Delta t}{1 + K\Delta_t}
\end{aligned}$$

可以看到，直接力矩法 (23) 需要关节加速度 \ddot{q} ，但是关节加速度通常需要对关节位置编码器进行两阶差分来得到，信噪比。而动量积分观测器的方法相较于直接力矩检测并不需要关节加速度，只需要关节位置、速度以及动力学方程中的各矩阵的辨识结果。

6.4 力控制

6.4.1 直接力控制

直接力控制为直接控制关节电机力矩，让机器人末端对环境产生接触力。设机器人末端对环境产生的接触力为 $\mathcal{F}_d \in \mathbb{R}^{6 \times 1}$ ，那么直接力控制器的设计为：

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + J^T(q) \underbrace{\{\mathcal{F}_d + K_p(\mathcal{F}_d - \mathcal{F}_m) + K_i \int (\mathcal{F}_d - \mathcal{F}_m) dt\}}_{\text{PI feedback}}, \tag{31}$$

其中 \mathcal{F}_m 为通过机器人末端力传感器测得的接触力（需从末端坐标系转换到机器人底座坐标系）。

6.4.2 力位混合控制

本文为了更好的分析力位混合控制，将机械臂的末端视为质点，机器人的关节力矩在克服掉惯性力、柯氏力、摩擦力以及重力之后对质点产生的广义力（力矩）为 \mathcal{F} ，具体如图 8 所示。

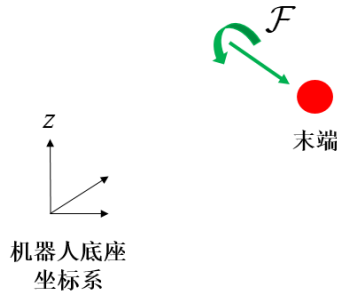


图 8: 末端广义力模型

\mathcal{F} 的具体表达为：

$$\mathcal{F} = J^{-T}(q)\{\tau - M(q)\ddot{q} - C(q, \dot{q})\dot{q} - G(q)\}, \quad (32)$$

其中 $J^{-T}(q)$ 表示雅克比矩阵转置的逆。

那么基于末端广义力模型，来进行末端的位置控制，可以直接设计 PID 控制器如下：

$$\mathcal{F} = \text{PID}(X_d - Tr(q)), \quad (33)$$

其中 $X_d \in \mathbb{R}^{6 \times 1}$ 表示期望末端在机器人底座坐标系下的位姿。

考虑机器人需要实现在复杂曲面零件上进行恒力打磨的案例，这时不仅需要能够控制机器人末端的位姿，还需要控制末端与零件之间的打磨力（接触力），如图 9 所示。

记在 t 时刻需要达到的期望打磨点为 $X_d = [x_d, y_d, z_d, \alpha_d, \beta_d, \gamma_d]^T$ （在机器人底座坐标系下），并且该打磨点所在的零件表面上的法方向记为 $[n_x, n_y, n_z]^T$ 。那么复杂曲面的恒力打磨任务可以表述为，在 t 时刻机器

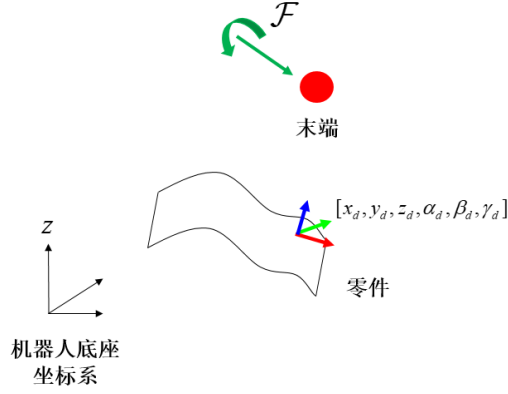


图 9: 复杂曲面零件打磨

人末端需要达到目标打磨点 X_d 并且目标接触力沿着法线的反方向且保持恒定。记恒定打磨力大小为 $F_c \in \mathbb{R}^+$ ，那么目标接触力为：

$$\mathcal{F}_d = [-n_x F_c, -n_y F_c, -n_z F_c, 0, 0, 0]^T.$$

最后复杂曲面零件恒力打磨的力位混合控制器设计为：

$$\mathcal{F} = \underbrace{\text{PID}(X_d - Tr(q))}_{\text{position}} + \underbrace{\mathcal{F}_d + \text{PI}(\mathcal{F}_d - \mathcal{F}_m)}_{\text{force}}, \quad (34)$$

其中 \mathcal{F}_m 为通过机器人末端力传感器测得的接触力（需从末端坐标系转换到机器人底座坐标系）。然后根据公式 (32)，可以计算出关节力矩控制律为：

$$\begin{aligned} \tau = & M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \\ & + J^T(q) \left\{ \underbrace{\text{PID}(X_d - Tr(q))}_{\text{position}} + \underbrace{\mathcal{F}_d + \text{PI}(\mathcal{F}_d - \mathcal{F}_m)}_{\text{force}} \right\}. \end{aligned} \quad (35)$$

复杂曲面恒力打磨另外一个关键点在于零件打磨点以及法方向的正确识别，相关技术会在第 7 节中进行介绍。

7 视觉感知

视觉传感器是机器人在作业中实现目标检测、环境理解的重要传感器。对视觉传感器生成的数据进行处理，能够引导机器人完成抓取、跟踪、放置、装配等复杂的操作。

7.1 2D 感知

2D 感知主要通过 2D 摄像头来进行图像的获取。一般来说获取的图片需要定义像素坐标系，如图 10所示，一般约定图片左上角为坐标原点。

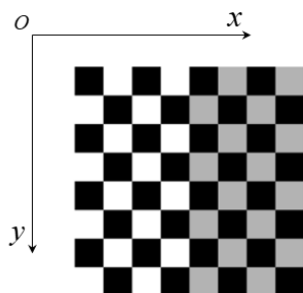


图 10: 像素坐标系

图片里面的像素点可以表示为坐标 (u, v) 的形式，根据小孔成像原理，像素点对应的真实物体在相机坐标系下的三维空间坐标 (X, Y, Z) 和像素坐标 (u, v) 之间的关系如下：

$$\begin{cases} u = f \frac{X}{Z} \\ v = f \frac{Y}{Z} \end{cases} \quad (36)$$

其中 f 表示焦距。关于相机内参的具体细节可以参考文章⁶。

从公式 (36)中可以看到，如果想要从图片像素进行分析，还需要知道物距即 Z 的值，所以需要一些辅助的标定板来进行物距以及相机姿态的确定，比如 Aruco⁷(如图 11所示) 等工具。

7.2 3D 感知

3D 感知相较于 2D 感知，能够直接获取物距 Z 的值，所以能够更加准确且全面的还原场景。3D 感知生成的信息一般为点云信息，即为一系列三维空间的点集 $\{x_i, y_i, z_i | i = 1, \dots, N\}$ ，如图 12所示。

点云数据的处理相较于 2 维图像数据会更加复杂且繁琐，具体可以参考 Open3D⁸进行点云数据处理的学习。较为常用的使用方式为获取到物体

⁶<https://zhuanlan.zhihu.com/p/87334006>

⁷https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

⁸<http://www.open3d.org/>

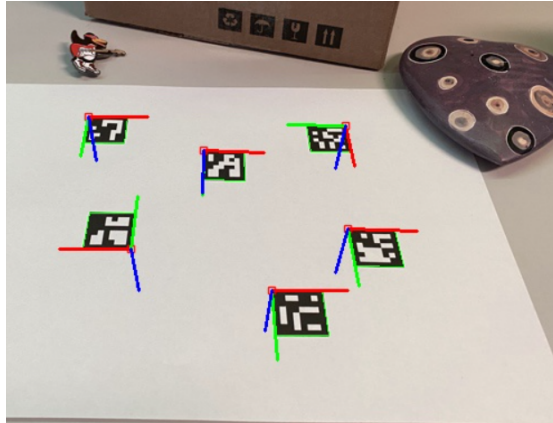


图 11: Aruco 姿态估计

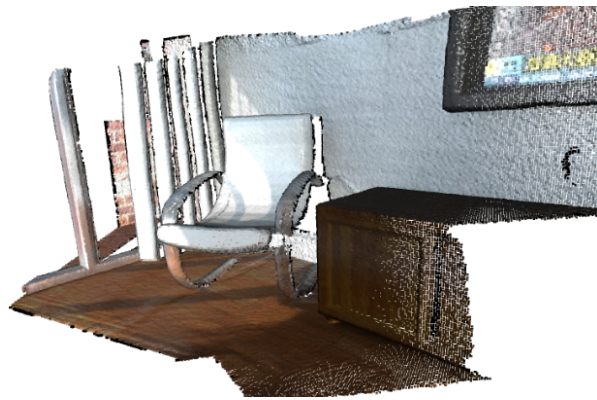


图 12: 点云数据

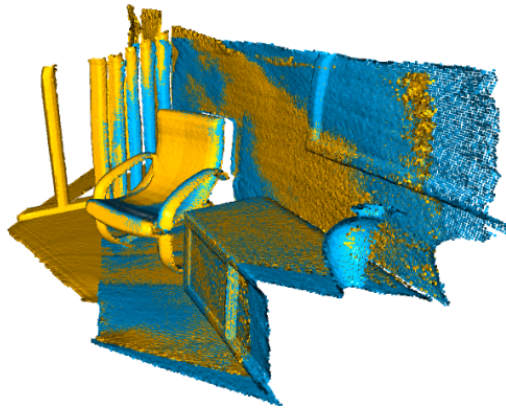
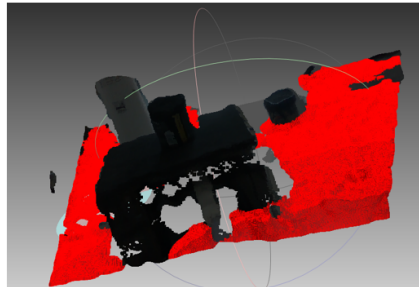


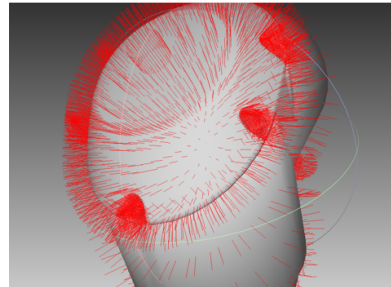
图 13: 点云配准

的点云数据，然后和 CAD 模型生成的理想点云进行配准 (Registration, 如图 13所示) 从而得到物体在相机坐标系下的空间位姿信息。

同时点云数据的处理推荐使用 MeshLab⁹开源软件进行相关的图形化验证操作。



平面检测



法线估计

图 14: MeshLab 操作

⁹<https://www.meshlab.net/>