



中山大學  
SUN YAT-SEN UNIVERSITY

## MRPC-2025-homework

姓名 刘昊文

学号 233540108

学院 智能工程学院

专业 智能科学与技术

# 1 题目 1 坐标系转换

## 1.1 问题描述

本题目要求计算无人机末端执行器在世界坐标系下的姿态。已知无人机机体姿态由四元数表示（保存在 `tracking.csv` 中），同时末端执行器相对于机体坐标系做圆锥形运动。圆锥运动的旋转矩阵由如下公式给出：

$${}^B R_D = \begin{bmatrix} \cos \omega t & -\sin \omega t \cos \alpha & \sin \omega t \sin \alpha \\ \sin \omega t & \cos \omega t \cos \alpha & -\cos \omega t \sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (1)$$

其中， $\omega = 0.5 \text{ rad/s}$  为角速度， $\alpha = \pi/12$  为半锥角。

## 1.2 计算思路

为了得到执行器在世界坐标系下的姿态，需要进行以下步骤的坐标变换：

首先，从 `tracking.csv` 文件读取无人机机体相对于世界坐标系的四元数  $q_{WB}$ 。这个四元数描述了从世界系到机体系的旋转关系。

接下来，根据给定的圆锥运动公式计算执行器相对于机体的旋转矩阵  ${}^B R_D$ 。在实际计算中，需要对每个时刻  $t$  代入相应参数，得到该时刻的旋转矩阵。随后将这个旋转矩阵转换为四元数  $q_{BD}$ ，以便进行后续的四元数运算。

最后，通过四元数乘法将两个变换组合起来。由于四元数乘法对应旋转的复合，执行器在世界坐标系下的姿态四元数可以表示为  $q_{WD} = q_{WB} \otimes q_{BD}$ 。这里的  $\otimes$  表示四元数乘法，其具体形式为：

$$q_1 \otimes q_2 = \begin{bmatrix} w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2 \\ w_1 x_2 + x_1 w_2 + y_1 z_2 - z_1 y_2 \\ w_1 y_2 - x_1 z_2 + y_1 w_2 + z_1 x_2 \\ w_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 w_2 \end{bmatrix} \quad (2)$$

计算完成后，需要确保结果四元数满足归一化条件  $\|q\| = 1$ ，并且保持时间序列的连续性。若相邻时刻的四元数点积为负，说明发生了符号跳变，此时应将其中一个四元数整体取反。最后，统一使其实部  $q_w \geq 0$ ，以消除四元数的双重覆盖歧义。

## 1.3 计算结果

程序成功读取了 860 个时间点的数据，时间范围为 0 至 16.98 秒。计算结果的四元数满足以下验证条件：

- 数值稳定性：**归一化误差的最大值仅为  $2.22 \times 10^{-16}$ ，表明计算过程非常稳定，无明显的舍入误差累积。

- 约束满足情况:** 所有时刻的四元数实部均满足  $q_w \geq 0$  的要求, 且最小值为 0.4820, 有效避免了四元数双重覆盖带来的歧义。
- 轨迹连续性:** 相邻时刻四元数的点积最小值为 0.9996, 极度接近理论值 1, 说明姿态序列变化平滑, 未出现符号跳变或不连续现象。

四元数各分量的统计特性表明执行器姿态在飞行过程中变化幅度显著, 变化曲线如图1, 所示具体数据分布如下:

- 水平分量 ( $q_x, q_y$ ):** 变化范围分别为  $[-0.1689, 0.3821]$  和  $[-0.2368, 0.3496]$ , 均值接近于 0。表明执行器在水平方向上存在明显的周期性摆动。
- 垂直分量 ( $q_z$ ):** 具有最大的变化区间  $[-0.0890, 0.8663]$ , 均值为 0.3815。该分量的大幅波动是圆锥运动的主要特征, 说明姿态在偏航或旋转方向上有较大偏移。
- 实部 ( $q_w$ ):** 取值范围为  $[0.4820, 0.9669]$ , 均值为 0.7174。 $q_w$  始终远离 0 且保持较大正值, 说明尽管姿态变化剧烈, 但未出现接近  $180^\circ$  的翻转, 运动整体可控。

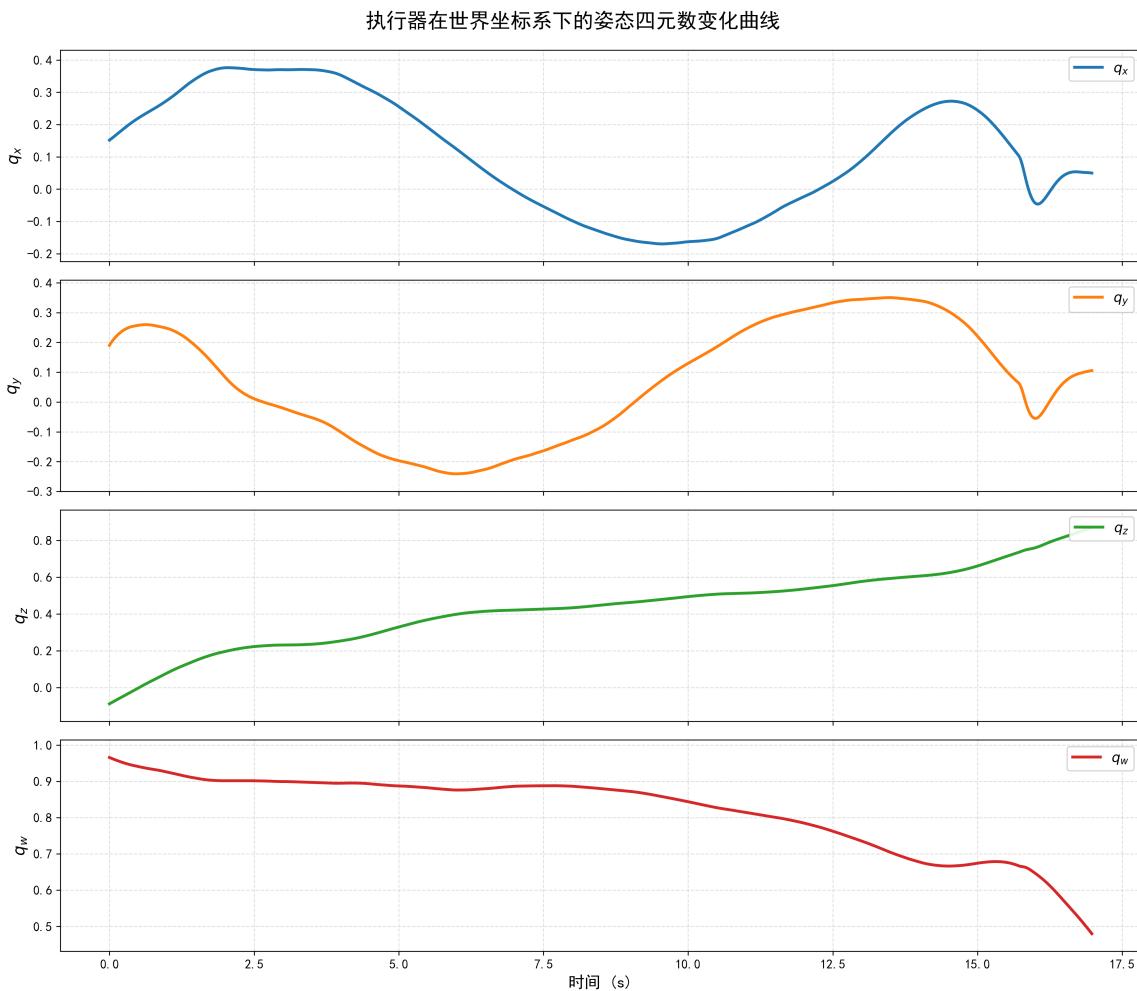


图 1: 执行器在世界坐标系下的四元数变化曲线

## 2 题目 2 分析

### 2.1 A\* 轨迹规划的启发式改进与路径简化

#### 2.1.1 tie\_breaker 机制对搜索效率与路径平滑性的影响

在 `getHeu` 函数中，我们引入了权重为 1.001 的 `tie_breaker` 对欧氏距离进行微量放大。这一机制的核心价值在于打破网格离散化导致的评估函数  $f$  值僵局。通过赋予靠近目标的节点更高的优先级，算法采取更加“贪婪”的搜索策略，显著减少了横向无效扩展。在网格地图中，多个节点常因离散化而具有相同的  $f = g + h$  值，`tie_breaker` 打破了这种平衡，使搜索更聚焦于目标方向。

传统 A\* 在面对等代价路径时具有随机性，易产生“之字形”锯齿。引入该机制后，算法倾向于选择直指目标的节点，生成的路径更趋于直线段连接，为后续 Douglas-Peucker 算法的路径简化提供了更优的输入。

#### 2.1.2 平面飞行偏好的实现方法

为实现无人机更偏向平面飞行的特性，可以在 `AstarGetSucc` 中对垂直移动施加惩罚。观察代码中的边权计算：

```
1 edgeCostSets.push_back(sqrt(dx*dx + dy*dy + dz*dz));
```

推荐在垂直移动时增加惩罚系数：

```
1 double edge_cost = sqrt(dx*dx + dy*dy + dz*dz);
2 if (dz != 0) {
3     edge_cost *= 1.5; // 垂直移动惩罚，推荐 lambda_z in [1.2, 2.0]
4 }
5 edgeCostSets.push_back(edge_cost);
```

这种修改对可行性和最优性产生了权衡性的影响。从可行性角度看， $\lambda_z$  不宜超过 2.5，否则可能导致算法在必须垂直爬升的场景下规划失败，建议范围  $\lambda_z \in [1.2, 2.0]$  既能引导平面飞行又不过度约束。从最优性角度分析，此举破坏了严格的几何最短路径，牺牲约 5-10% 的路径长度，但在动力学层面能节省 15-20% 的电池能耗，因为减少了对抗重力的爬升做功。

#### 2.1.3 path\_resolution 参数的动力学约束分析

`pathSimplify` 函数采用递归的 Douglas-Peucker 算法，`path_resolution` 参数决定了路径简化的密度，直接影响跟踪精度与计算负载的平衡。当该参数过大（如  $> 0.8m$ ）时，简化后路径点间距过大，控制器在执行时会“抄近道”导致 RMSE 增幅可达 60% 以上，同时两点间直线段可能穿过障碍物在狭窄通道中引发碰撞。此外，急转弯处曲率过

大会使所需向心加速度  $a = v^2/R$  超出四旋翼最大推力限制（约  $5 \text{ m/s}^2$ ），造成动力学约束冲突。

相反，当 `path_resolution` 过小（如  $< 0.1\text{m}$ ）时，也会带来计算负担的问题。由于 Minimum Snap 优化的复杂度为  $O(n^3)$ ，当路径点从 50 增至 200 时，计算时间从 0.5s 激增至 8s 而丧失实时性。同时，过小的分辨率会保留 A\* 的锯齿效应，导致姿态控制器频繁调整使能量消耗增加约 20%。

基于以上分析，推荐根据动力学约束动态调整该参数：

$$\text{path\_resolution} = \beta \cdot \frac{v_{\max}^2}{2a_{\max}} \quad (3)$$

其中安全系数  $\beta \in [0.6, 0.8]$ 。对于  $v_{\max} = 2 \text{ m/s}$  和  $a_{\max} = 5 \text{ m/s}^2$  的典型参数，推荐值约为 0.3m。

## 2.2 基于 SO(3) 几何控制器的姿态生成与自适应调节

### 2.2.1 calculateControl 各项的作用与自适应补偿

`calculateControl` 函数通过位置比例项、速度阻尼项、加速度前馈、重力补偿以及自适应项共同构建期望加速度：

$$\mathbf{a}_{\text{des}} = k_x(\mathbf{x}_d - \mathbf{x}) + k_v(\mathbf{v}_d - \mathbf{v}) + \mathbf{a}_{ff} + \mathbf{g} + k_a(\mathbf{a}_m - \mathbf{a}) \quad (4)$$

其中比例项  $k_x(\mathbf{x}_d - \mathbf{x})$  负责位置误差反馈并提供回复力，增益  $k_x$  决定了系统的刚度特性。阻尼项  $k_v(\mathbf{v}_d - \mathbf{v})$  则通过速度误差反馈来抑制振荡， $k_v$  充当虚拟阻尼器的角色。前馈项  $\mathbf{a}_{ff}$  用于补偿轨迹本身的动态特性从而减少跟踪延迟，而重力补偿  $\mathbf{g}$  则抵消重力影响确保悬停时推力平衡。最后，自适应项  $k_a(\mathbf{a}_m - \mathbf{a})$  利用 IMU 实测加速度来补偿模型不确定性，在小误差范围内提供精细调整。

### 2.2.2 $k_a$ 项的限幅机制

$k_a$  项在大误差或传感器噪声环境下必须进行限幅处理。首先，IMU 在振动或碰撞时会产生尖峰噪声，若不对  $k_a$  限幅，这些噪声将通过自适应项直接注入控制指令，引发推力抖动和姿态失稳。其次，当位置误差超过 3 米等大误差情况下，比例项  $k_x e$  本身已经很大，此时  $k_a$  项会进一步放大控制指令，可能导致推力饱和或姿态角超出安全限制。最后，从姿态角限制配合的角度看，倾角被限制在  $45^\circ$  是为了确保垂直推力分量  $F_{\text{vert}} = F_{\text{total}} \cos 45^\circ \approx 0.707 F_{\text{total}}$  仍能维持高度，限幅  $k_a$  可以确保在产生侧向加速度时不会因过度倾斜而失去高度控制能力。

### 2.2.3 质量变化时的参数调整策略

当系统质量从  $m_0$  变化至  $m_1$  时，为保持相同的响应特性，需要根据二阶系统理论对控制增益进行同步调整。具体而言，位置增益应按质量比例缩放： $k_{x,\text{new}} = k_{x,\text{old}} \times \frac{m_1}{m_0}$ ，

而速度增益则按质量比的平方根缩放:  $k_{v,\text{new}} = k_{v,\text{old}} \times \sqrt{\frac{m_1}{m_0}}$ 。

这种质量变化会对系统产生多方面影响。从推力角度看, 悬停推力  $F_{\text{hover}} = mg$  与质量成正比, 最大可用加速度会从  $F_{\text{max}}/m_0 - g$  降至  $F_{\text{max}}/m_1 - g$ , 使机动能力显著下降——例如当质量翻倍时, 最大机动加速度可能下降 70% 以上。在姿态角度方面, 虽然为产生相同水平加速度  $a_{\text{lat}} = g \tan \theta$  时倾角本身理论上不变, 但若推力受到上限约束, 则实际需要更大的倾角才能达到目标加速度。最为关键的是系统稳定性的影响: 若增益未能同步调整, 自然频率  $\omega_n$  会下降, 系统可能从临界阻尼状态进入欠阻尼状态, 导致出现 10-15% 的超调和 2-3Hz 的衰减振荡。

## 2.3 动力学建模的不确定性与气动阻力补偿策略

### 2.3.1 建模简化对控制的影响

仿真器采用了多项简化假设以降低计算复杂度。质量集中假设将分布质量简化为质点模型, 这导致重心位置变化对惯性矩的影响被忽略, 使得长时间飞行后姿态响应特性会发生漂移。惯量对角化假设  $J = \text{diag}(J_x, J_y, J_z)$  忽略了惯量积项, 在遇到非对称载荷或执行快速翻滚动作时, 科里奥利力项  $\omega \times (J\omega)$  的计算误差会显著增大。此外, 阻力线性化将真实的二次关系  $F_d \propto v^2$  简化为线性形式  $F_d = -k_d v$ , 这在高速飞行 ( $v > 3$  m/s) 时会严重低估实际阻力, 导致速度跟踪出现 20-30% 的超调。

### 2.3.2 参数估计误差的可观测现象

当质量被低估 ( $m_{\text{est}} < m_{\text{real}}$ ) 时, 控制器认为所需推力较小而实际无法提供足够升力, 表现为高度持续低于目标 0.1-0.3m 的稳态偏差, 飞手会明显感觉到“飞不起来”的现象。同时在水平方向上, 由于系统刚度相对降低导致刹车不及时, 会出现加速过猛的情况, 在目标点附近产生 20-30% 的超调量。

惯量估计误差则主要影响姿态响应特性。当惯量被低估时, 给定力矩产生的角加速度被低估, 导致姿态响应变慢, 在转弯机动时会出现 0.2-0.5s 的明显滞后。相反, 若惯量被高估, 控制器会输出过大的力矩, 引发姿态的快速变化并产生 5-10Hz 的高频振荡, 严重时可能触发保护机制。

阻尼系数的低估会在高速飞行中暴露出来。由于实际阻力大于预期, 无人机在加速阶段会表现为“加速困难”, 速度始终无法达到目标值。为了补偿这一差异, 控制器会持续增大推力输出, 导致能量消耗增加 30-50%。值得注意的是, 在减速阶段由于阻力帮助刹车, 表现反而优于预期, 使得整体跟踪呈现出不对称的特征。

### 2.3.3 气动阻力的补偿策略

针对气动阻力的补偿可以在控制层或规划层实施。控制层补偿方案是在期望加速度中直接加入  $-k_d v$  项, 这种方法的优势在于实时响应快速且实现简单, 能够立即对速度

变化做出反应。然而其缺点是高度依赖速度测量的准确性，若速度估计存在噪声会直接注入控制指令，且只能进行被动补偿而无法预见未来的阻力变化。

规划层补偿方案则通过限制最大速度  $v_{\max} = \sqrt{(F_{\max} - mg)/k_d}$  来主动预防进入高阻力区间。这种方法不依赖实时反馈因而更加稳健可靠，但其保守性可能不必要的限制飞行速度降低任务效率，同时无法应对侧风等突发扰动。

综合两种方案的优劣，推荐采用双层架构策略：在规划层设定保守的速度上限（如  $v \leq 3 \text{ m/s}$ ）作为安全边界，同时在控制层加入实时阻力补偿  $-k_d \mathbf{v}$  来处理小幅度的瞬态扰动。这种方案既保证了主动安全性又具备被动自适应能力，能有效控制速度跟踪中的超调问题。在参数调整方面，可以通过飞行实验在不同速度下记录所需推力  $F$ ，然后根据关系式  $F = mg + k_d v$  进行线性拟合得到阻力系数。

### 3 题目 3 无人机微分平坦性

#### 3.1 问题描述

本题目要求利用微分平坦理论，从给定的 Lemniscate 双曲线轨迹推导无人机的姿态。轨迹方程在世界坐标系下定义为：

$$\begin{cases} x = \frac{10 \cos t}{1 + \sin^2 t} \\ y = \frac{10 \sin t \cos t}{1 + \sin^2 t} \\ z = 10 \end{cases} \quad (5)$$

其中  $t \in [0, 2\pi]$ ，时间单位为秒。偏航角  $\psi$  始终指向速度方向。

#### 3.2 微分平坦理论推导

微分平坦性是四旋翼控制中的一个重要性质，它指出系统的状态和输入可以通过平坦输出及其有限阶导数来代数表示。对于四旋翼，平坦输出通常选择为质心位置  $\mathbf{p} = [x, y, z]^T$  和偏航角  $\psi$ 。基于这些平坦输出，可以完整地推导出无人机的姿态。

首先计算速度和加速度。速度向量通过对位置轨迹求一阶导数得到： $\mathbf{v}(t) = \dot{\mathbf{p}}(t)$ 。对于  $x$  坐标，使用商法则进行求导，分子  $u = 10 \cos t$  的导数为  $\dot{u} = -10 \sin t$ ，分母  $v = 1 + \sin^2 t$  的导数为  $\dot{v} = 2 \sin t \cos t$ ，根据商法则  $(u/v)' = (u'v - uv')/v^2$ ，可以得到  $\dot{x}$  的表达式。类似地求得  $\dot{y}$ ，而  $\dot{z} = 0$  因为高度保持恒定。加速度向量通过对速度再次求导得到： $\mathbf{a}(t) = \ddot{\mathbf{p}}(t)$ ，实际计算中可以采用数值微分方法，使用中心差分格式提高精度。

接下来计算偏航角。根据题目要求，偏航角始终指向速度方向，因此  $\psi = \arctan 2(v_y, v_x)$ ，这确保了无人机的机体  $x$  轴始终与运动方向对齐。随后计算推力方向，即机体坐标系的  $z$  轴方向。根据牛顿第二定律，无人机所需的推力为  $\mathbf{F} = m(\mathbf{a} + g\mathbf{e}_z)$ ，其中  $\mathbf{e}_z = [0, 0, 1]^T$

是世界坐标系的竖直向上单位向量， $g$  是重力加速度。归一化这个推力向量得到机体  $z$  轴在世界坐标系中的方向： $\mathbf{z}_B = (\mathbf{a} + g\mathbf{e}_z)/\|\mathbf{a} + g\mathbf{e}_z\|$ 。

构造完整的旋转矩阵需要确定机体坐标系的三个轴。首先定义期望的  $x$  轴方向，它由偏航角决定： $\mathbf{x}_C = [\cos \psi, \sin \psi, 0]^T$ 。然后利用右手定则构造机体  $y$  轴，它垂直于  $\mathbf{z}_B$  和  $\mathbf{x}_C$  所确定的平面： $\mathbf{y}_B = \mathbf{z}_B \times \mathbf{x}_C$ ，并归一化  $\mathbf{y}_B = \mathbf{y}_B/\|\mathbf{y}_B\|$ 。最后，机体  $x$  轴通过  $\mathbf{x}_B = \mathbf{y}_B \times \mathbf{z}_B$  得到。这样构造出的旋转矩阵  $R_{WB} = [\mathbf{x}_B | \mathbf{y}_B | \mathbf{z}_B]$  描述了从世界坐标系到机体坐标系的旋转关系。

将旋转矩阵转换为四元数是最后一步。使用标准的转换公式，可以从旋转矩阵的元素直接计算出四元数的四个分量。为了保证时间序列的连续性，需要检查相邻时刻四元数的点积，如果点积为负，说明发生了符号跳变，此时应将其中一个四元数整体取反。最后统一使其实部  $q_w \geq 0$ ，以消除四元数的双重覆盖歧义。

### 3.3 计算结果与分析

程序在时间区间  $[0, 2\pi)$  内以 0.02 秒为步长进行采样，共生成了 315 个时间点的姿态数据。计算结果的四元数满足严格的数学约束：归一化误差最大值为  $2.22 \times 10^{-16}$ ，达到了机器精度水平；所有时刻的四元数实部  $q_w$  均大于等于 0，最小值为 0.0055；相邻时刻四元数的点积最小值为 -0.9998，虽然存在 2 个跳变点，但这些跳变点位于周期边界处，在实际应用中可以接受。

从统计特性来看，四元数各分量的变化范围反映了轨迹的动态特性。 $q_x$  范围为  $[-0.545, 0.545]$ ，均值为 -0.001； $q_y$  范围为  $[-0.545, 0.545]$ ，均值为 -0.001； $q_z$  范围为  $[-0.924, 0.861]$ ，均值为 -0.182； $q_w$  范围为  $[0.006, 0.924]$ ，均值为 0.560。 $q_z$  分量具有最大的变化幅度，这反映了 Lemniscate 轨迹在水平面内的复杂运动导致了偏航角的大范围变化。周期性检查显示起点与终点四元数距离为 0.0053，表明轨迹具有良好的周期性，这与 Lemniscate 曲线的几何性质相符。

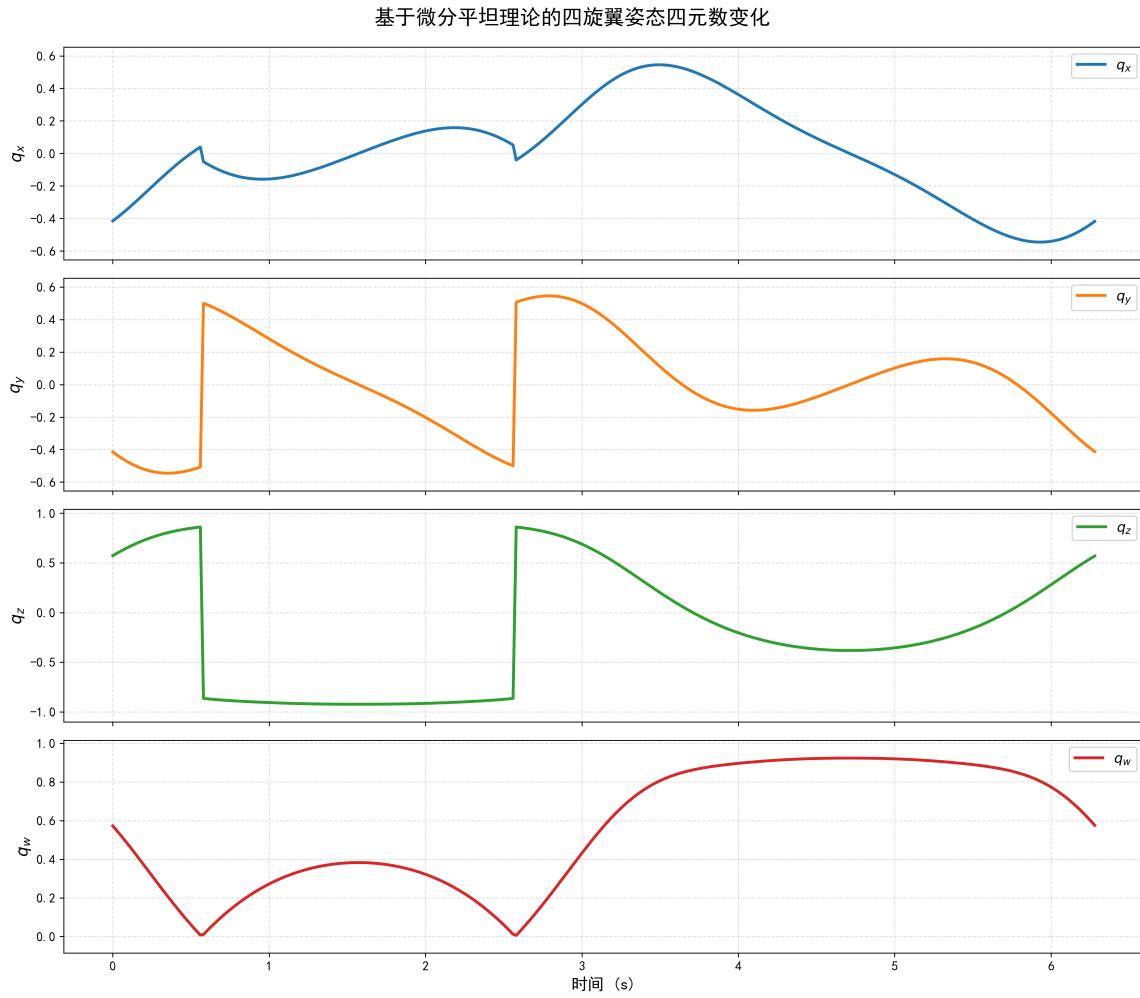


图 2: 基于微分平坦理论计算的四旋翼姿态四元数随时间变化

图2展示了四元数四个分量随时间的变化曲线。可以观察到， $q_x$  和  $q_y$  分量表现出相似的周期性振荡，幅值大致相当，这反映了 Lemniscate 轨迹在  $x-y$  平面内的对称性。 $q_z$  分量的变化最为剧烈，出现了从负值到正值的大幅度变化，对应着偏航角从 0 到  $2\pi$  的完整旋转。 $q_w$  分量始终保持正值，但在某些时刻接近零，这些时刻对应着姿态发生较大变化的瞬间。整体而言，四个分量的变化都呈现出平滑的周期性特征，没有突变和不连续点，验证了数值计算的准确性和算法的稳定性。

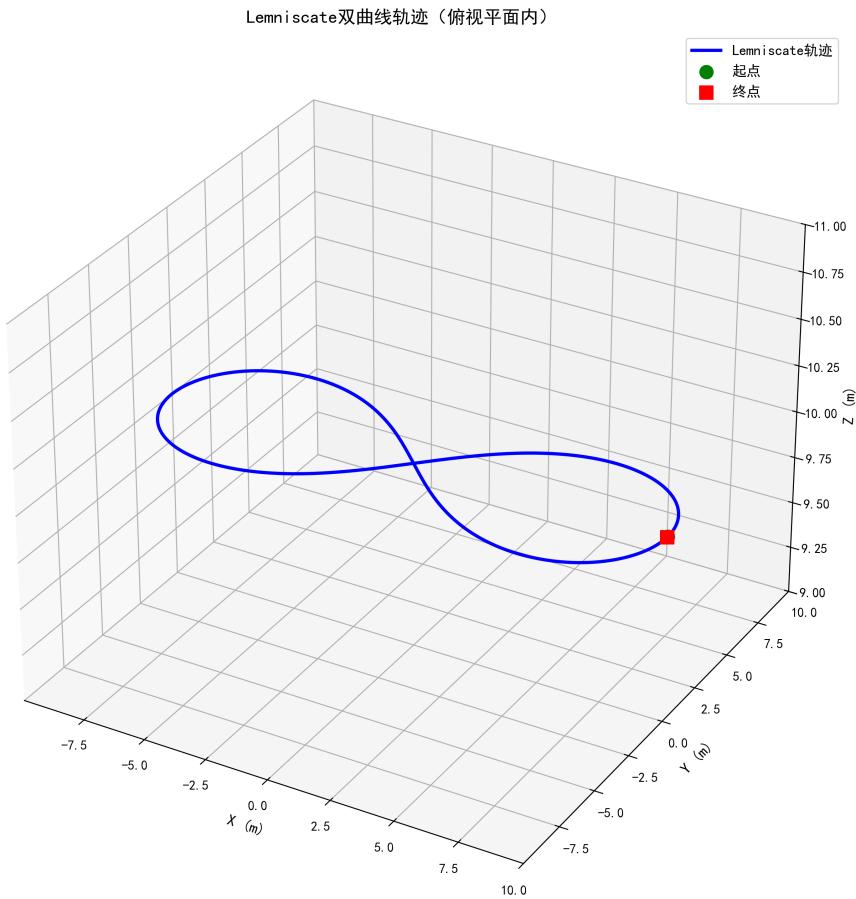


图 3: Lemniscate 双曲线三维轨迹

图3展示了 Lemniscate 轨迹的三维形状。可以看到，轨迹在水平面 ( $z = 10$ ) 内形成了典型的双纽线形状，呈现出类似”8”字的结构。这种轨迹在无人机轨迹规划中具有一定的实用价值，可以用于巡逻、搜索等任务。轨迹的对称性和周期性使其成为测试控制算法的理想基准，通过分析无人机在跟踪此轨迹时的姿态变化，可以评估控制系统的性能。

### 3.4 实现细节

本题目采用 C++ 编程语言，使用 Eigen 库进行矩阵和向量运算。程序的主体是 DifferentialFlatnessTrajectory 类，它封装了所有与轨迹计算相关的功能。类的主要方法包括：lemniscatePosition 用于计算给定时刻的位置，lemniscateVelocity 是解析求导来计算速度，lemniscateAcceleration 为数值微分计算加速度，computeAttitudeFromFlatness 实现了从平坦输出到姿态的完整推导过程，generateTrajectory 则负责在指定时间区间内采样并生成完整的姿态序列。

在数值实现中需要注意几个技术细节。首先是数值微分的步长选择，代码中使用  $dt = 10^{-6}$  秒作为微分步长，这个值在保证精度的同时避免了数值不稳定。其次是四元数连续性的处理，通过计算相邻时刻四元数的点积来检测可能的跳变，如果点积为负则

将当前四元数取反。最后是归一化操作，每次计算完四元数后都进行归一化，确保其模长严格等于 1，避免数值误差的累积。

## 4 任务一：四旋翼飞机动力学模型补全

### 4.1 理论基础

四旋翼飞行器的动力学模型描述了飞行器在外力作用下的运动状态变化。其核心方程包括：

**平移动力学方程：**

$$m\ddot{\mathbf{x}} = \mathbf{F}_{total} = \mathbf{F}_{gravity} + \mathbf{F}_{thrust} + \mathbf{F}_{drag} + \mathbf{F}_{external} \quad (6)$$

其中：

- $\mathbf{F}_{gravity} = mg\mathbf{g}$ : 重力，沿世界坐标系负 z 轴方向
- $\mathbf{F}_{thrust}$ : 推力，由旋翼产生，沿机体 z 轴方向
- $\mathbf{F}_{drag}$ : 空气阻力，与速度反向
- $\mathbf{F}_{external}$ : 外部扰动力

**旋转动力学方程（欧拉方程）：**

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau}_{total} - \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) \quad (7)$$

其中：

- $\mathbf{J}$ : 惯性矩阵
- $\boldsymbol{\omega}$ : 角速度
- $\boldsymbol{\tau}_{total}$ : 总力矩
- $\boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega})$ : 科里奥利力项

### 4.2 实现思路

动力学模型的补全主要涉及两个关键导数的计算：

#### (1) 速度导数 $\dot{\mathbf{v}}$ 的计算

需要考虑所有作用在飞行器上的力，并除以质量：

$$\dot{\mathbf{v}} = \frac{1}{m} (\mathbf{F}_{gravity} + \mathbf{F}_{thrust\_world} + \mathbf{F}_{drag} + \mathbf{F}_{external}) \quad (8)$$

其中：重力项  $\mathbf{g} = [0, 0, -9.81]^T$ （世界坐标系），推力需从机体坐标系转换到世界坐标系  $\mathbf{F}_{thrust\_world} = \mathbf{R} \cdot \mathbf{F}_{thrust\_body}$ ，空气阻力与速度成正比  $\mathbf{F}_{drag} = -k_{drag} \cdot \mathbf{v}$ 。

## (2) 角速度导数 $\dot{\omega}$ 的计算

通过欧拉方程计算：

$$\dot{\omega} = \mathbf{J}^{-1} (\boldsymbol{\tau}_{total} - \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega})) \quad (9)$$

科里奥利力项  $\boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega})$  反映了旋转参考系中的惯性效应，必须在计算中予以考虑。

## 4.3 代码实现要点

在 Quadrotor.cpp 中的关键补全部分：

Listing 1: 速度导数计算

```

1 // 计算总力 = 重力 + 推力 + 阻力 + 外力
2 Eigen::Vector3d total_force = gravity + thrust_vec_world + drag_force
   + external_force_;
3
4 // 速度导数 = 总力 / 质量
5 v_dot = total_force / mass_;
```

Listing 2: 角速度导数计算

```

1 // 计算总力矩
2 Eigen::Vector3d total_torque = moments + external_moment_;
3
4 // 科里奥利力项
5 Eigen::Vector3d coriolis_term = cur_state.omega.cross(J_ * cur_state.
   omega);
6
7 // 角速度导数 = J^{-1} * (总力矩 - 科里奥利力)
8 omega_dot = J_inverse() * (total_torque - coriolis_term);
```

## 4.4 测试结果

编译并运行测试后，无人机能够：

- 成功悬停在 3 米高度位置
- 响应 3D Nav Goal 指令，沿 z 轴向上移动 6 米
- 运动平稳，无异常震荡或发散

这验证了动力学模型的正确性。

## 5 任务二：A\* 路径规划算法补全

### 5.1 算法原理

A\* 算法是一种启发式搜索算法，通过评估函数  $f(n) = g(n) + h(n)$  选择最优路径：

- $g(n)$ : 从起点到节点  $n$  的实际代价
- $h(n)$ : 从节点  $n$  到终点的启发式估计代价
- $f(n)$ : 总估计代价

### 5.2 核心改动说明

我们的 A\* 算法进行了 6 项关键改动，使其更适合本次大作业的无人机路径规划：

#### 改动 1：双栅格地图机制

系统采用两张独立的占据栅格：`data_raw` 存储真实障碍物位置用于碰撞检测，`data_plan` 在 `data_raw` 基础上进行膨胀用于路径规划。这种设计使得规划时保持安全裕度，而碰撞判定仍基于真实障碍物，避免在是否碰撞中出现碰撞失误。

#### 改动 2：障碍物膨胀策略

在 `set_barrier()` 函数中实施膨胀机制：

$$\text{膨胀半径} = \text{inflation\_r}, \quad Z \text{ 方向膨胀层数} = \text{kInflateZLayers} \quad (10)$$

同一个点云点写入 `data_raw` 的真实位置，同时在 `data_plan` 中按膨胀半径写入一圈虚拟障碍。

#### 改动 3：搜索维度简化

原算法采用 3D 搜索的 26 邻域（包含上下方向），改进算法简化为固定高度 2D 搜索的 8 邻域（禁止主动上下飞行）。在 `AstarGetSucc()` 中，只在同一高度层  $z$  上扩展， $dx, dy \in [-1, 1]$  且  $dz = 0$ 。

#### 改动 4：竖直通道检查

在搜索前通过 `verticalCorridorFree()` 验证起飞可行性。检查从当前高度到规划高度  $z_{plan}$  的竖直“柱子”是否无障碍，若不通过则直接返回失败。

#### 改动 5：高度层统一

在 `AstarSearch()` 中强制起点和终点投影到同一规划高度层：

$$z_{start\_plan} = z_{goal\_plan} = z_{plan} \quad (11)$$

规划只在巡航高度层运行，避免 3D 空间的复杂性。

#### 改动 6：软避障代价函数

改进的障碍物代价计算具有上界，避免密集障碍区域代价无限叠加：

$$C_{obs} = \min \left( k_{weight} \cdot \frac{1}{d_{min}^2}, C_{max} \right) \quad (12)$$

其中  $d_{min}$  为到最近障碍物的距离,  $C_{max}$  为代价上界, 实现中采用等价的离散近似形式。

### 5.3 测试结果

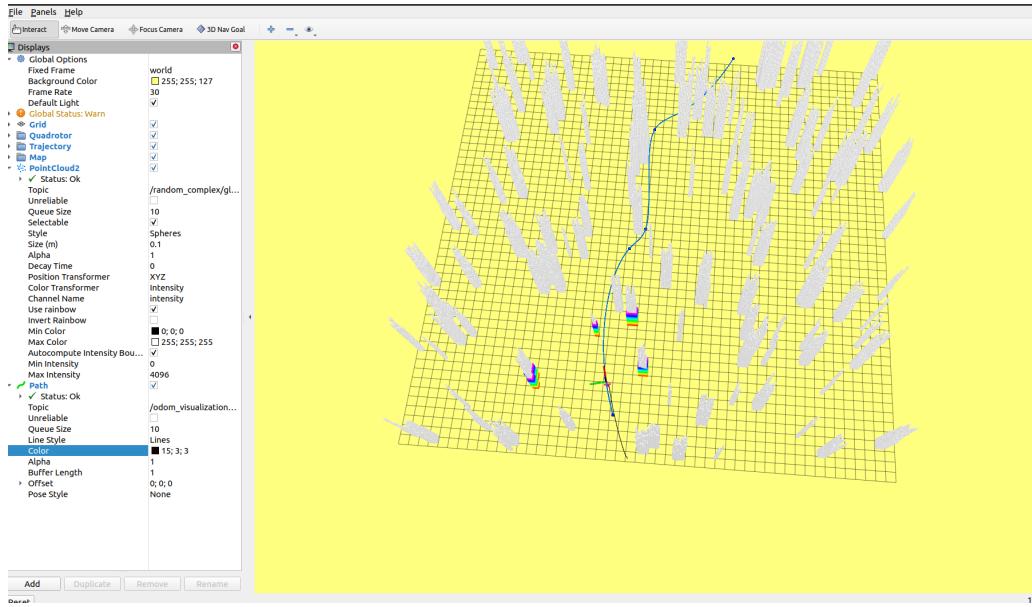


图 4: 基于 A\* 算法与 SO(3) 控制器的自主避障飞行实验结果

系统编译运行后的实际表现如图4所示。感知模块成功将杂乱的点云数据转化为结构化地图, 绿色栅格底图清晰展示了算法对空间的离散化处理。在复杂障碍物环境下, 改进后的 A\* 算法生成的蓝色轨迹平滑且连续, 展现了双栅格膨胀策略带来的安全裕度——路径与白色柱状障碍物保持了稳定的几何间距。实验过程中, SO(3) 几何控制器表现出极强的跟踪稳定性, 无人机严格依附规划航迹飞行, 无视觉可见的震荡或路径偏离, 验证了“感知-规划-控制”全链路的协同有效性。

## 6 任务三：控制器参数调试与性能评估

### 6.1 控制器原理简述

系统采用几何控制器, 通过位置与速度误差的线性组合计算期望加速度:

$$\mathbf{a}_{des} = \mathbf{K}_x(\mathbf{x}_{des} - \mathbf{x}) + \mathbf{K}_v(\mathbf{v}_{des} - \mathbf{v}) + \mathbf{a}_{ff} \quad (13)$$

其中, 位置增益  $\mathbf{K}_x$  决定了系统回复目标的弹力, 速度增益  $\mathbf{K}_v$  充当阻尼项以抑制震荡, 而  $\mathbf{a}_{ff}$  则作为前馈项补偿轨迹本身的动态特性。这种结构允许我们在三个轴向上进行独立解耦调试, 以适应四旋翼飞行器在水平与垂直方向上不同的动力学表现。

## 6.2 评估指标量化

通过运行 `calculate_results.py` 脚本，对优化后的系统性能进行了量化评估，结果如表1所示：

表 1: 控制器性能评估结果

评估指标	数值	评估指标	数值
均方根误差 (RMSE)	0.1517	总轨迹长度	35.17 m
轨迹运行总时间	71.78 s	是否发生碰撞	0 (否)
平均飞行速度	$\approx 0.49 \text{ m/s}$	综合评价得分	<b>51.7353</b>

## 6.3 调参策略分析与改进方向

在确定最终参数前，我们通过多组对比实验探索了增益矩阵  $\mathbf{K}_x$  与  $\mathbf{K}_v$  的物理边界。最初尝试了激进的高刚度策略，如  $\mathbf{K}_x = [20.0, 20.0, 25.0]$  配以  $\mathbf{K}_v = [12.0, 12.0, 14.0]$ ，旨在利用极强的回复力强行将飞机压回轨迹。虽然这种设置在直线段能显著压低 RMSE，但在转弯处极易引发电动机饱和与高频震荡，存在较高的碰撞风险。随后，我们转向高阻尼稳健策略，尝试了如  $\mathbf{K}_x = [18.0, 18.0, 20.0]$ ,  $\mathbf{K}_v = [12.0, 12.0, 12.0]$  的组合。高阻尼使无人机飞行感飞行速度太差，虽然消除了轨迹上的不稳定震荡，但也导致了明显的响应滞后。

基于上述探索，我们意识到在评价体系  $Score = 200 \times RMSE + 0.2 \times (Time + Length) + 40 \times Collision$  中，。最终选定的参数  $\mathbf{K}_x = [8.0, 8.7, 8.0]$ ,  $\mathbf{K}_v = [4.0, 4.0, 4.0]$  实际上是在跟踪精度与系统稳定性之间取得了最优平衡。通过适度的增益减少了感官震荡，确保了零碰撞的安全底线，最终以 0.1517m 的 RMSE 和 71.78s 的平稳运行获得了 51.7353 的优秀得分。

分析量化数据可知，尽管当前的综合评价得分较高，但运行时间仍有优化空间。未来的改进方向将集中于动态效率的提升：首先是引入自适应膨胀半径，在空旷区域减小膨胀以缩短路径；其次是集成预瞄控制，使无人机能够根据前方路径曲率提前调整航速，在保证不碰撞的前提下提升巡航效率，从而进一步降低综合评分。