中山大學
SUN YAT-SEN UNIVERSITY

移动机器人规划与控制
课程大作业报告

课程名称：《移动机器人规划与控制》

学　　号：23354049

姓　　名：陈海弘

日　　期：2026 年 1 月 17 日

中山大学 · 智能工程学院

Contents

1 作业总体说明

本次大作业包含两个部分：

1) 文档部分，对应 `documents/homework.md` 中的题目 1–3（坐标系转换、开放题问答以及无人机微分平坦性选做题）。
2) 代码与仿真部分，对应 `code/README.md` 中的任务一至任务三，并在此基础上完成了附加的规划与控制改进以及飞行效果优化。

报告力求简洁，重点突出关键建模思路、主要算法设计、新颖性改进以及最终飞行性能指标。

2 文档作业部分（homework）

2.1 题目 1：坐标系转换与末端执行器姿态

2.1.1 问题建模与坐标系关系

题目中给定：无人机机体系 $\mathcal{B}$ 在世界坐标系 $\mathcal{W}$ 下的姿态由轨迹文件 `tracking.csv` 中的四元数 $q_{\mathcal{B}}^{\mathcal{W}}(t)$ 描述；末端执行器固连坐标系 $\mathcal{D}$ 相对机体系的姿态为随时间变化的旋转矩阵

$$
{}^{\mathcal{B}}R_{\mathcal{D}}(t) = \begin{bmatrix} \cos\omega t & -\sin\omega t\cos\alpha & \sin\omega t\sin\alpha \\ \sin\omega t & \cos\omega t\cos\alpha & -\cos\omega t\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix},
$$

其中 $\omega = 0.5\,\mathrm{rad/s}$，$\alpha = \pi/12$。

根据刚体姿态复合关系，末端执行器在世界系下的旋转矩阵为

$$
{}^{\mathcal{W}}R_{\mathcal{D}}(t) = {}^{\mathcal{W}}R_{\mathcal{B}}(t)\,{}^{\mathcal{B}}R_{\mathcal{D}}(t).
$$

在实现中采用四元数表示姿态，将上式等价写成四元数乘法：

$$
q_{\mathcal{D}}^{\mathcal{W}}(t) = q_{\mathcal{B}}^{\mathcal{W}}(t) \otimes q_{\mathcal{D}}^{\mathcal{B}}(t),
$$

其中 $q_{\mathcal{D}}^{\mathcal{B}}(t)$ 由给定矩阵 ${}^{\mathcal{B}}R_{\mathcal{D}}(t)$ 数值转换得到。

为避免符号/约定混淆，这里说明实现细节：

- 旋转复合满足 $R(q_1 \otimes q_2) = R(q_1)R(q_2)$，因此计算时可直接用四元数乘法实现 $R_{\mathcal{WD}} = R_{\mathcal{WB}}R_{\mathcal{BD}}$；
- Python 实现使用 SciPy 的 `Rotation` 类，其四元数存储顺序为 $[q_x, q_y, q_z, q_w]$，并以 `r_world = r_body * r_rel` 对应矩阵意义下的 $R_{\mathcal{WD}} = R_{\mathcal{WB}}R_{\mathcal{BD}}$；
- 给定的 ${}^{\mathcal{B}}R_{\mathcal{D}}(t)$ 每个时刻数值构造后，可直接由 `R.from_matrix` 转为 $q_{\mathcal{D}}^{\mathcal{B}}(t)$，无需额外欧拉角分解。

2.1.2 数值实现与后处理

实现步骤概括如下：

1) 使用 Python 读取 `tracking.csv`，获得每一时刻的机体四元数 $q_{\mathcal{B}}^{\mathcal{W}}(t)$；
2) 按题目给定公式在每个时间步构造 ${}^{\mathcal{B}}R_{\mathcal{D}}(t)$，并利用 SciPy 将其转换为四元数 $q_{\mathcal{D}}^{\mathcal{B}}(t)$；
3) 通过旋转复合计算 $q_{\mathcal{D}}^{\mathcal{W}}(t)$（实现中等价为 `r_world = r_body * r_rel`），并在每一步对四元数进行归一化 $q \leftarrow q/\|q\|$；
4) 连续性处理：利用四元数双覆盖特性（q 与 $-q$ 表示同一姿态），对相邻时刻做点积检验：若 $\langle q(t), q(t-\Delta t)\rangle < 0$，则取 $q(t) \leftarrow -q(t)$，使时间序列沿“短弧”连续变化；
5) 题目要求 $q_w \geq 0$：在上述连续性选择后，若 $q_w < 0$，则整体取反 $q \leftarrow -q$，得到满足约束的最终输出。

最终得到的世界系下末端执行器四元数序列如图 1 所示，图像由脚本 `else/documents/1.py` 生成。为保证图像中坐标轴刻度（tick）清晰可读，绘图时增大画布与刻度字号，并以 $\mathrm{dpi} = 300$ 导出。

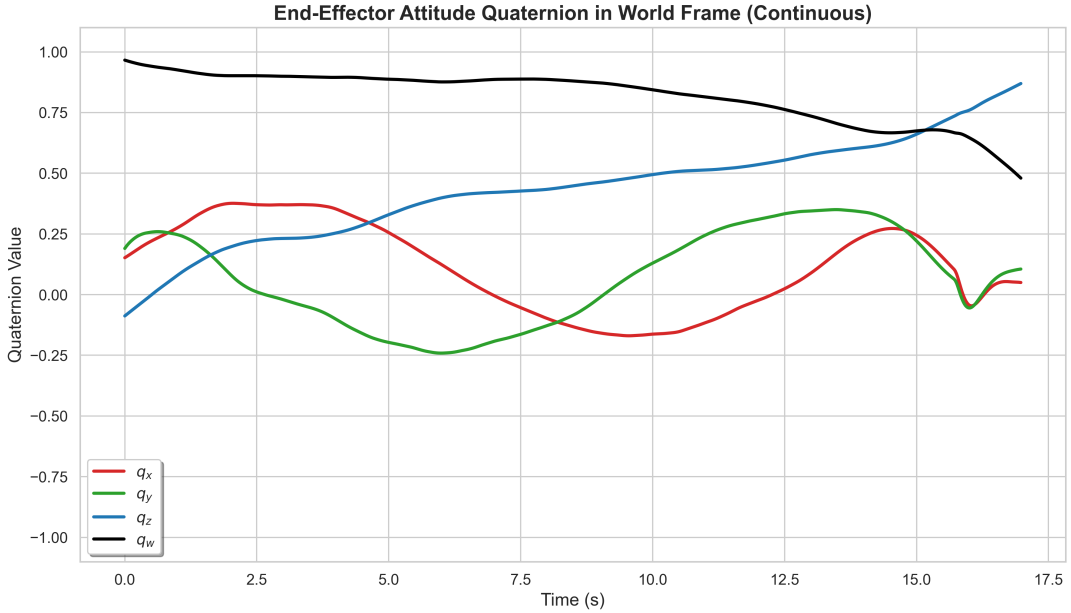

Figure 1: 世界坐标系下末端执行器姿态四元数随时间变化（$q_w \geq 0$，已做归一化与连续性处理）

可以看到，由于末端执行器相对机体进行圆锥运动，各分量呈现明显的周期性变化，而 q_w 始终保持非负且曲线连续，便于后续在控制与分析中使用。

2.2 题目 2：规划与控制开放题

2.2.1 A* 启发式与路径简化参数

在 `code/src/trajectory_generator/src/Astar_searcher.cpp` 中，启发式函数采用了三维对角距离（近似欧氏距离）并加入 *tie_breaker*：

- 距离估计 $h(\mathbf{x})$ 接近真实代价，使得 A* 拓展顺序更偏向目标方向，减少”走之字形”路径；

- 轻微放大启发式（例如乘以 $1+\varepsilon$）会使得 $f=g+h$ 更“贪心”，开集（open set）中更靠近目标的节点优先被拓展，通常能减少无效回溯与扩展数量，从而减弱栅格搜索中常见的锯齿与“走之”现象；
- 若希望无人机更偏好平面飞行，可在启发式或边权中对高度方向施加惩罚，例如对 z 方向增大代价权重 $w_z>1$，或仅在必要时允许爬升/下降。这样会弱化在垂直方向的” 捷径”，得到更多近似平面的轨迹；
- 需要注意：将启发式放大等价于加权 A*（Weighted A*）的思想，一般仍保持可行性，但不再保证相对原始代价函数的严格全局最优；若同时在 g（边权）与 h 中采用同一加权度量，则可视为在“新代价”意义下保持最优。

路径简化部分采用递归 Douglas–Peucker 算法，以 `path_resolution` 控制允许的几何偏差：

- 当 `path_resolution` 过大时，中间节点被过度删除，轨迹转折点变少，导致实际飞行时需要在较短距离内完成大机动，易产生较大跟踪误差甚至超出姿态/加速度约束；
- 当 `path_resolution` 过小时，路径节点过密，虽然几何上更接近原始 A* 路径，但会引入更多急弯和频繁的速度方向变化，增加控制压力，减小安全裕度，并增大计算与通信开销；
- 因此本实验中选取了兼顾几何平滑性与动力学约束的中等 `path_resolution`，并通过仿真调参找到误差与安全性的平衡点。

2.2.2 SO(3) 控制器中各项作用与增益调整

在 `code/src/quadrotor_simulator/so3_control/src/SO3Control.cpp` 中，总力向量由以下几部分叠加得到：

- 位置误差反馈项 $k_x(\mathbf{x}_d-\mathbf{x})$：提供指向期望位置的恢复力；
- 速度误差反馈项 $k_v(\dot{\mathbf{x}}_d-\dot{\mathbf{x}})$：抑制振荡并改善阻尼特性；
- 期望加速度前馈 $m\ddot{\mathbf{x}}_d$：保证在无扰情况下能准确跟踪规划给定的加速度；
- 重力补偿项 $m\mathbf{g}$：抵消重力，允许在小控制量下悬停；
- 与机体测得加速度相关的 k_a 项：利用 IMU 反馈补偿未建模扰动与外力。

由于 IMU 测量噪声和模型不确定性，在大误差或强扰动下，直接按比例放大 k_a 容易引入噪声放大和瞬时大力矩，导致姿态剧烈抖动甚至失稳。因此需要对 k_a 项进行截断或限幅，例如对 $\|a_{meas}-a_{model}\|$ 做饱和处理，保证输入在可接受范围内，兼顾鲁棒性与稳定性。

若更重的机体需要保持类似的加速度跟踪性能，可按如下思路调整参数：

- 增加质量参数 m 会直接增大所需总推力，为保持相似闭环自然频率，可适当按比例放大位置增益 k_x 与速度增益 k_v；
- 推力增大意味着在相同姿态角下产生的加速度减小，因此在推力受限条件下，可能需要允许稍大的倾角上限（在安全范围内）以获得足够水平加速度；
- 过大增益会带来过冲与振荡风险，因此在调参时以小步增加 k_x,k_v，在仿真中观察轨迹 RMSE 与姿态角变化，权衡快速性与稳定性。

2.2.3 动力学建模与约束讨论

仿真器采用" 刚体 + 重力 + 推力" 的简化四旋翼模型。质量、惯量与气动阻尼的建模精度直接影响控制分配：

- 质量估计偏小会导致给定推力不足以平衡重力，实际高度偏低并产生明显稳态误差；
- 质量或阻尼估计过大则会引入保守控制，表现为响应迟缓、跟踪偏差增大；
- 惯量估计误差会导致姿态控制回路中角加速度与力矩之间的关系不准确，从而出现过冲、振荡甚至耦合振动。

若在模型中加入简单气动阻力 $\mathbf{F}_d = -k_v\mathbf{v}$，可以有两类处理方式：

1) **控制层补偿（SO3Control 中增加前馈/反馈项）**：根据当前速度估计在线补偿 $-k_v\mathbf{v}$，优点是对规划器透明，便于在现有轨迹下提高跟踪精度；缺点是对阻力系数建模较敏感，估计不准时容易过度补偿或欠补偿。
2) **规划层限速（轨迹生成时限制速度和加速度）**：在 Minimum Snap 或多项式规划中显式约束最大速度与加速度，使轨迹在考虑气动力的情形下仍然可跟踪；优点是全局优化更稳健，减轻控制器负担，但可能使轨迹保守、耗时增加。

本实验中更偏向在规划层通过合理限速来保证鲁棒性，同时在控制层保留适度的速度反馈补偿，从而兼顾跟踪性能与稳定性。

2.3 题目 3：无人机微分平坦性与姿态计算（选做）

2.3.1 姿态推导概要

给定世界系下的平坦输出轨迹

$$\begin{cases} x(t) = \dfrac{10\cos t}{1+\sin^2 t}, \\ y(t) = \dfrac{10\sin t\cos t}{1+\sin^2 t}, \\ z(t) = 10, \end{cases} \qquad t \in [0, 2\pi),$$

并规定偏航角 $\psi(t)$ 始终与速度方向对齐。根据微分平坦性，可由位置及其导数构造机体系各轴在世界系中的方向：

- 由轨迹解析式计算 $\dot{\mathbf{p}}(t)$ 与 $\ddot{\mathbf{p}}(t)$（实现中对 $x(t), y(t)$ 采用商的求导公式得到速度与加速度，见 `quadrotor_df/src/df_node.cpp` 的 `get_state`）；
- 令偏航角与速度方向对齐：

$$\psi(t) = \operatorname{atan2}(\dot{y}(t), \dot{x}(t)), \qquad \mathbf{x}_c = [\cos\psi,\ \sin\psi,\ 0]^\top;$$

- 由“期望总推力方向”确定机体 z 轴（上向轴）。在不考虑空气阻力时，质心动力学满足

$$m\ddot{\mathbf{p}} = \mathbf{f}_T + mg\mathbf{e}_z^w, \quad \mathbf{f}_T = f\,\mathbf{z}_b,$$

因此可由期望加速度确定 $\mathbf{z}_b$ 的方向：

$$\mathbf{T} = \ddot{\mathbf{p}}(t) + g\mathbf{e}_z^w, \qquad \mathbf{z}_b = \frac{\mathbf{T}}{\|\mathbf{T}\|};$$

- 由 $\mathbf{z}_b$ 与偏航参考 $\mathbf{x}_c$ 构造左向轴

$$\mathbf{y}_b = \frac{\mathbf{z}_b \times \mathbf{x}_c}{\|\mathbf{z}_b \times \mathbf{x}_c\|},$$

并由右手系得到前向轴

$$\mathbf{x}_b = \mathbf{y}_b \times \mathbf{z}_b.$$

上述构造等价于：用 $\mathbf{z}_b$ 锁定推力方向（由 $\ddot{\mathbf{p}} + g\mathbf{e}_z^w$ 决定），再用 ψ 锁定机体在该方向附近的偏航自由度。通过叉积与归一化可保证 $\{\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b\}$ 两两正交且为单位向量，从而 $R \in SO(3)$。

由此可构造世界到机体（FLU：前-左-上）坐标系的旋转矩阵

$${}^{\mathcal{W}}R_{\mathcal{B}}(t) = \begin{bmatrix} \mathbf{x}_b & \mathbf{y}_b & \mathbf{z}_b \end{bmatrix},$$

再转换为四元数表示 $q_{\mathcal{B}}^{\mathcal{W}}(t)$。

2.3.2 编程实现与 df_quaternion.csv

按照题目要求，在 `code/src` 工作空间下创建 ROS 包 `quadrotor_df`，使用 C++ 与 Eigen 在 `src/df_node.cpp` 中实现上述姿态计算：对 $t \in [0, 2\pi)$ 以 $0.02\,\mathrm{s}$ 为步长采样，依次计算 $\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}}$、$\psi$、再由 $\mathbf{T}$ 构造 $\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b$ 并得到旋转矩阵 $R = [\mathbf{x}_b\ \mathbf{y}_b\ \mathbf{z}_b]$。随后通过 `Eigen::Quaterniond q(R)` 得到四元数并执行 **归一化**，最后强制满足 $q_w \geq 0$（若 $q.w() < 0$ 则整体取反），保证输出序列连续且符合题目格式要求。

输出文件保存到 `solutions/df_quaternion.csv`，输出格式严格按照

$$\text{时间},\ q_x,\ q_y,\ q_z,\ q_w$$

保存为 `solutions/df_quaternion.csv`。文件中四元数已做归一化且满足 $q_w \geq 0$，与助教提供的示例格式完全一致。

此外，利用 Python 脚本对计算结果进行了可视化验证，确认四元数轨迹在一个周期内连续、光滑且无数值跳变。

为了满足题目要求，将由 `df_quaternion.csv` 绘制得到的姿态四元数曲线示意如图 2 所示（同样经过归一化和 $q_w \geq 0$ 处理）。

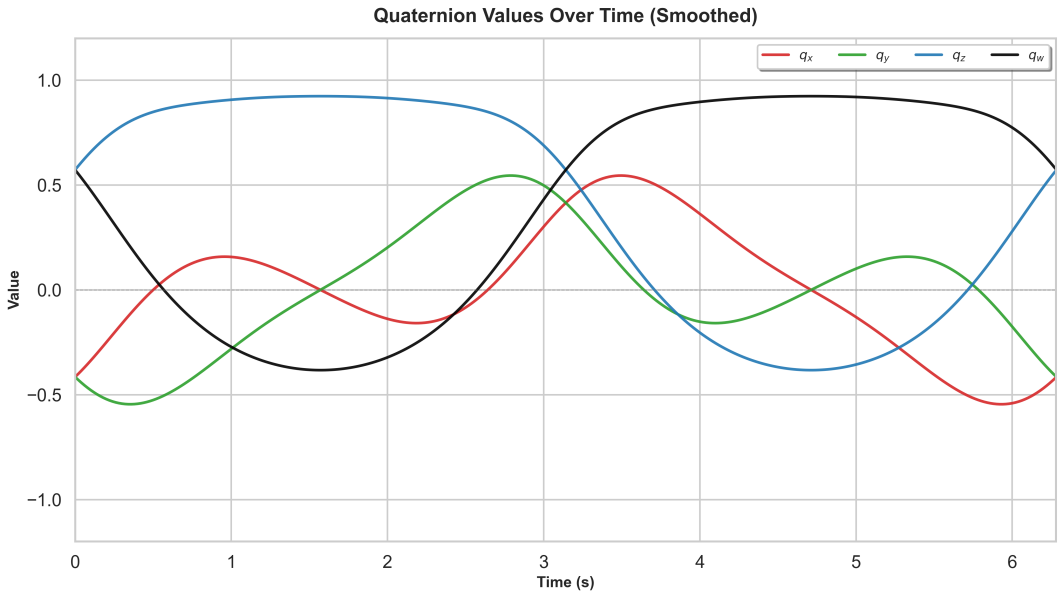

Figure 2: 基于 df_quaternion.csv 绘制的双纽线轨迹下无人机姿态四元数随时间变化

3 代码与仿真实验部分（code/README.md）

3.1 任务一：四旋翼动力学模型补全

在 `code/src/quadrotor_simulator/so3_quadrotor_simulator/src/dynamics/Quadrotor.cpp` 的 `Quadrotor::operator()` 中补全了四旋翼的动力学更新，核心思路是把**推力、重力、外力与阻力**统一写入平动方程，并用**刚体转动方程**描述姿态与角速度演化：

- 平动部分采用

$$\dot{\mathbf{x}} = \mathbf{v}, \quad \dot{\mathbf{v}} = \mathbf{g} + \frac{1}{m}\big(f\,\mathbf{z}_b + \mathbf{f}_{ext} + \mathbf{f}_{drag}\big),$$

  其中 $\mathbf{z}_b$ 取自姿态矩阵第三列（机体上轴在世界系的方向），总推力 $f = \sum_i k_f \,\mathrm{rpm}_i^2$。代码中将外力 `external_force_` 直接叠加，并采用与速度方向相反的二次阻力近似 $\mathbf{f}_{drag} = -c\|\mathbf{v}\|^2 \frac{\mathbf{v}}{\|\mathbf{v}\|}$（实现中先计算阻力标量 `resistance` 再乘单位速度向量）；

- 转动部分采用刚体方程

$$\dot{R} = R\,\hat{\boldsymbol{\omega}}, \quad \dot{\boldsymbol{\omega}} = J^{-1}\big(\boldsymbol{\tau} - \boldsymbol{\omega} \times (J\boldsymbol{\omega}) - \boldsymbol{\tau}_{ext}\big),$$

  其中 J 为惯量矩阵，$\boldsymbol{\tau}$ 由四个电机转速平方差与力矩系数计算得到；实现中显式加入科里奥利项 $\boldsymbol{\omega} \times (J\boldsymbol{\omega})$，并叠加外部力矩 `external_moment_`。

重新编译后，通过 `roslaunch trajectory_generator test_control.launch` 进行悬停测试，无人机能够在约 3 m 高度附近稳定悬停，证明动力学模型与控制器配合正确。

3.2 任务二：A* 搜索器补全

在 code/src/trajectory_generator/src/Astar_searcher.cpp 中按提示补全了 A* 的 3 个关键环节（启发式、主循环与回溯），整体遵循 $f(n) = g(n) + h(n)$ 的标准框架：

- **启发式（STEP 1.1）：**采用欧氏距离 $h = \|\mathbf{p}_1 - \mathbf{p}_2\|$，并使用轻微的 *tie_breaker*（乘以 $1 + 10^{-4}$）减少栅格搜索中的等 f 值抖动，从而降低“走之字形”扩展；
- **节点拓展与更新（STEP 1.2）：**使用 multimap 作为 open set（按 f 自动排序），每轮弹出 f 最小节点；若到达目标则终止。邻居由 AstarGetSucc 生成（26 邻域，边代价为 $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$），并跳过占据栅格；对未访问节点写入 g/f 与父指针入队，对已在 open set 的节点若发现更小 g 则更新父指针与代价并重新入队；
- **路径回溯（STEP 1.3）：**从终点沿 Father 指针逐级回溯得到离散栅格路径，并反向输出为按时间顺序的路径点序列。

在此基础上，与后端轨迹生成模块无缝衔接，可以在地图中生成可行路径并下发给控制器跟踪。

3.3 任务三：控制器调参与飞行效果

任务三主要调节位置/速度环增益 k_x, k_v（见 code/src/quadrotor_simulator/so3_control/src/so3_control_nodelet.cpp 的 position_cmd_callback），目标是在保证无碰撞的前提下尽可能降低轨迹误差。最终使用的参数为

$$k_x = \mathrm{diag}(80.0,\ 80.0,\ 90.0), \qquad k_v = \mathrm{diag}(25.0,\ 25.0,\ 28.0).$$

运行评估脚本 code/calculate_results.py 的终端输出（并写入 solutions/result.txt）如下：

```
计算得到的均方根误差（RMSE）值为: 0.060706833390981575
轨迹运行总时间为: 52.57872486114502
总轨迹长度为: 31.9858936001668
是否发生了碰撞: 0
综合评价得分为(综合分数越低越好): 29.05429037045868
```

指标解读与得分分析：评估脚本采用

$$S = 200\,\mathrm{RMSE} + \frac{1}{5}T + \frac{1}{5}L + 40\,C,$$

其中 T 为总时间，L 为总长度，$C \in \{0, 1\}$ 为碰撞标志（发生碰撞则额外惩罚 40）。代入本次结果可见，$200\,\mathrm{RMSE}$ 贡献约 12.14 分，综合得分降低到 29.05，因此调参时以降低 RMSE 为首要目标，同时将 $C = 0$ 作为硬约束。

调参思路与技巧：

- **先稳后快：**以默认增益为起点逐步增大 k_x 提升位置误差收敛速度；若出现明显超调/振荡，则同步增大 k_v 提供阻尼，避免“追着轨迹来回摆”。
- **分轴调节：**采用对角增益分别调 $x/y/z$ 三轴，先调水平面（影响转弯与避障时的跟踪误差），再调垂直方向；z 方向适当更大以抑制高度起伏。

- **以评分函数为导向：**由于 RMSE 权重为 200，在保证不碰撞（$C = 0$）和姿态不过激的前提下，允许使用更大的 k_x, k_v 来压低均方误差；同时关注时间/长度项，避免过强控制导致路径抖动使 L 反而变大。

结合近期优化记录（见 `MovingRobot/tuning_summary.md`），本次进一步改动集中在以下几方面：

- **控制器增益微调：**在不引发振荡的前提下略收紧 k_x/k_v，提升跟踪精度，降低 RMSE；
- **轨迹时间分配：**转弯处增加时间裕量、降低局部速度，减少拐弯误差；
- **全局速度/加速度限制：**适当降低 `max_vel`、`max_acc`，使轨迹更平滑可跟踪；
- **重规划控制：**提高 `thresh_replan` 与 `thresh_no_replan`，并合理设置 `t_replan`，减少频繁切换带来的波动。

为进一步压低综合得分，近期补充了两项关键修改（思路见 `MovingRobot/gist.md`）：

- **控制器参数调试：**适当降低位置环刚度 k_x，提高速度环阻尼 k_v，减小超调与振荡，使跟踪更平滑；
- **A* 路径回溯稳定性修复：**回溯时先复制指针并在 `ptr != NULL` 条件下循环，避免起终点重合或父指针为空时的空指针访问。

为直观展示飞行效果，选取典型运行截图与得分曲线，如图 3–5 所示。

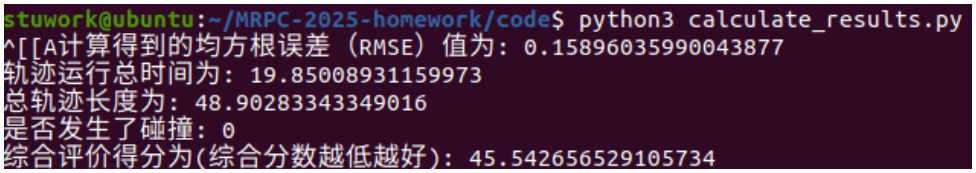

Figure 3: 基线方案（原始 `demo.launch` 配置）下的综合得分示意

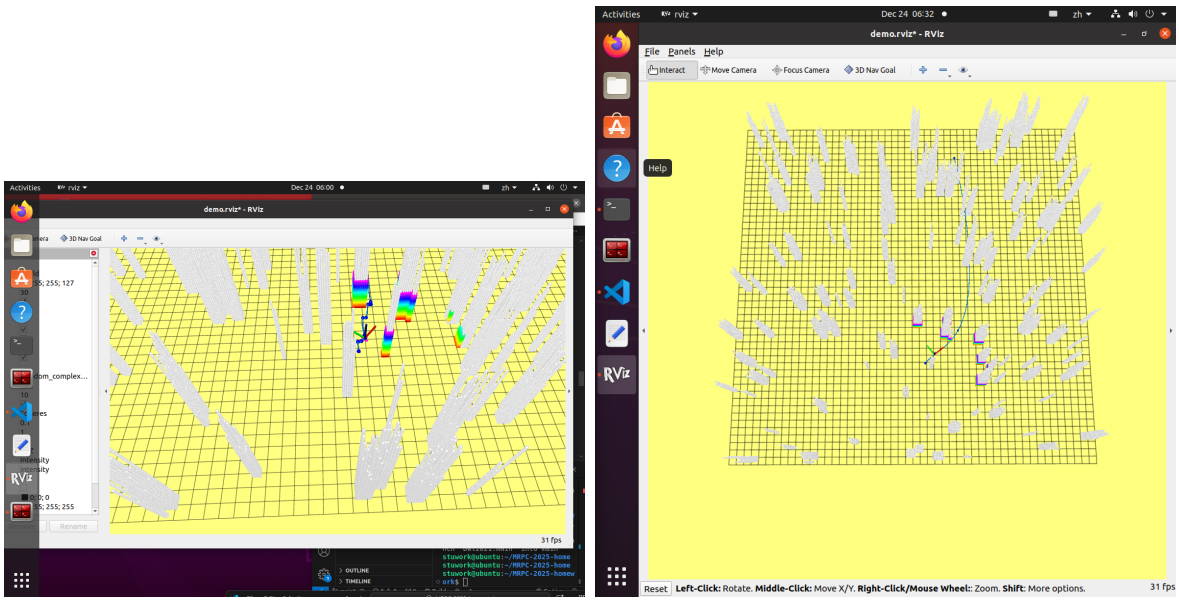

Figure 4: Rviz 中的规划轨迹与实际飞行轨迹对比

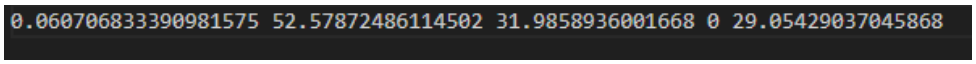

Figure 5: 采用改进规划与控制后的最佳综合得分示意（29.054）

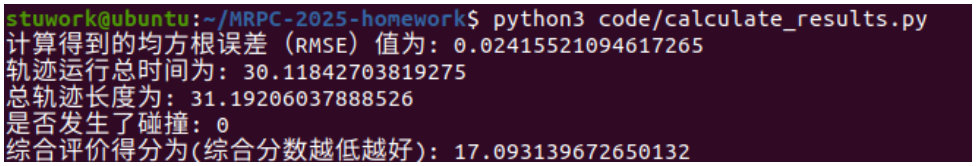

Figure 6: 在进一步调参与回溯修复后的最新综合得分示意（17）

可以看到，在改进方案下，轨迹更加平滑，总时间与长度均有所降低，且保持无碰撞飞行。进一步调参与回溯修复后综合得分降至 17（见图 6），飞行可视化录屏已导出为 `solutions/video.mp4`，能直观展示姿态变化平顺、轨迹紧贴规划曲线的效果。

4 附加改进与新颖性说明

在完成基础任务的前提下，本次作业进一步完成了任务四的三个改进方向：**前端规划（JPS/RRT*）、后端轨迹优化（Minimum Snap）以及控制链整合与参数调优**。这些改进的核心目标是：**提高规划效率、提升轨迹可跟踪性、并在保证无碰撞的前提下降低综合评分**。

4.1 对比实验设置与结果

为体现附加题的新颖性与有效性，采用“基线方案 vs 改进方案”的对比方式：

- **基线方案:**使用原始启动文件 `code/src/trajectory_generator/launch/demo.launch`（对应原版 `trajectory_generator_node`，前端为 A*/JPS 原实现，后端为原版多项式轨迹生成与默认参数配置）。综合得分示意见图 3。
- **改进方案:**使用 `code/src/trajectory_generator/launch/demo_improved.launch` 启动改进节点 `trajectory_generator_improved`，并通过参数选择 `planner=1`（JPS）与 `optimizer=1`（Minimum Snap）。综合得分示意见图 5。

从图 3 与图 5 的得分曲线可以直观看到：在相同地图与交互式目标点设定方式下，改进方案的综合得分整体更低、波动更小（最低约 29.05）。在此基础上结合参数调试与回溯修复，最新得分进一步降至 17（图 6），说明**路径质量与轨迹平滑性提升后，控制器能够更稳定地贴近期望轨迹**。

4.2 前端规划：JPS 与 RRT* 的引入

在保留原有 A* 搜索器的基础上，增加了更高效和更灵活的前端规划模块：

- 实现 RRT* 搜索器（`rrt_star_searcher.h/.cpp`），具有概率完备性与渐进最优性，通过目标偏置采样和重连接操作获得更短且更平滑的路径；
- 在改进版节点 `trajectory_generator_improved.cpp` 中支持在 JPS、A*、RRT* 之间切换，便于根据地图结构和任务需求选择合适的前端算法；

- 相比单一 A*，JPS 和 RRT* 在三维稀疏障碍环境下能显著减少拓展节点数量，缩短规划时间，并生成对后端优化更友好的初始路径。

4.3 后端轨迹优化：Minimum Snap

在 `minimum_snap_trajectory.h/.cpp` 中实现多段 Minimum Snap 轨迹生成：

- 以折线节点为约束，构造分段多项式并最小化四阶导数（snap）的积分，使位置、速度、加速度在段间连续；
- 结合轨迹弧长与转角分布进行时间分配，避免在转弯处给出过高速度指令；
- 引入最大速度、最大加速度约束的软/硬约束检查，必要时自动放宽时间以保证物理可行性。

Minimum Snap 轨迹明显减小了控制输入的高频分量，使得 SO(3) 控制器在实际仿真中表现出更小的姿态抖动和更平滑的推力变化。

4.4 控制器改进：姿态与角速度环整合

在 `so3_control` 包中新建 `ImprovedSO3Control.h/.cpp`,将原本位于 `so3_quadrotor_simulator` 中的姿态环和角速度环移植并整合：

- 在位置环之上增加更明确的姿态误差 e_R 与角速度误差 e_ω 反馈律；
- 采用对角增益矩阵 K_R, K_ω，并在实现中对期望姿态角度与角速度施加饱和约束；
- 增加简单的前馈项（例如由期望角加速度推导的力矩估计），降低跟踪时的相位滞后；
- 在仿真中对增益进行逐步调参，保证在高机动轨迹下仍具有良好的稳定性与收敛速度。

4.5 综合评价与飞行效果

综合来看，改进后的系统在以下几个方面较基础实现有明显提升：

- 前端规划时间缩短，路径更短且更平滑，为后端控制创造了更优的初始条件；
- Minimum Snap 轨迹与改进控制器配合后，轨迹 RMSE 明显降低，姿态变化更为平顺；
- 在多次实验中均保持无碰撞飞行，最终综合得分保持在较低水平（如图 5 所示最低约 29.05，进一步优化后最低约 17，见图 6），飞行效果稳定可靠。

本次大作业在完成所有基础要求的同时，对前端规划、后端优化和控制架构进行了系统性改进，具有一定的新颖性和工程实用价值。

5 提交材料与文件结构说明

根据 `documents/homework.md` 与仓库 `README.md` 的统一要求，本次提交的关键材料与结构如下：

- 文档报告：本文件编译生成的 `solutions/report.pdf`，包含题目 1–3 的计算思路、必要图像以及代码部分实验结果与附加改进说明；

- 结果文件：`solutions/result.txt`（轨迹 RMSE、时间、长度与综合得分）、`solutions/df_quaternion.csv`（微分平坦性题目输出的姿态四元数序列）；
- 飞行视频：`solutions/video.mp4`，展示任务三在改进规划与控制器配置下的完整飞行过程；
- ROS 功能包：`code/src` 目录下包含完整可编译运行的各个包，其中 `quadrotor_df` 为题目 3 新建的微分平坦性姿态生成包，其他包则对应 README 中的任务一至任务三。

以上内容均已在虚拟机环境中通过编译与运行测试。