

多点轨迹规划与速度控制实验

1. 实验目的

本实验旨在通过控制无人机执行多点轨迹飞行任务，帮助用户掌握以下技能和知识：

1. 理解多点轨迹规划的基本原理，掌握多个航点组成的轨迹路径规划方法；
2. 掌握基于速度控制的无人机轨迹跟踪技术，理解速度控制与位置控制的区别与联系；
3. 学习无人机偏航角控制策略，特别是动态偏航角调整的实现方法；
4. 理解飞行时间计算、控制频率设置和飞行器运动学约束等关键概念；
5. 掌握在Offboard模式下实现复杂轨迹控制的完整流程。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链^[1]。
- 硬件要求：笔记本/台式电脑1台^[2]。

3. 实验地址

例程目录：

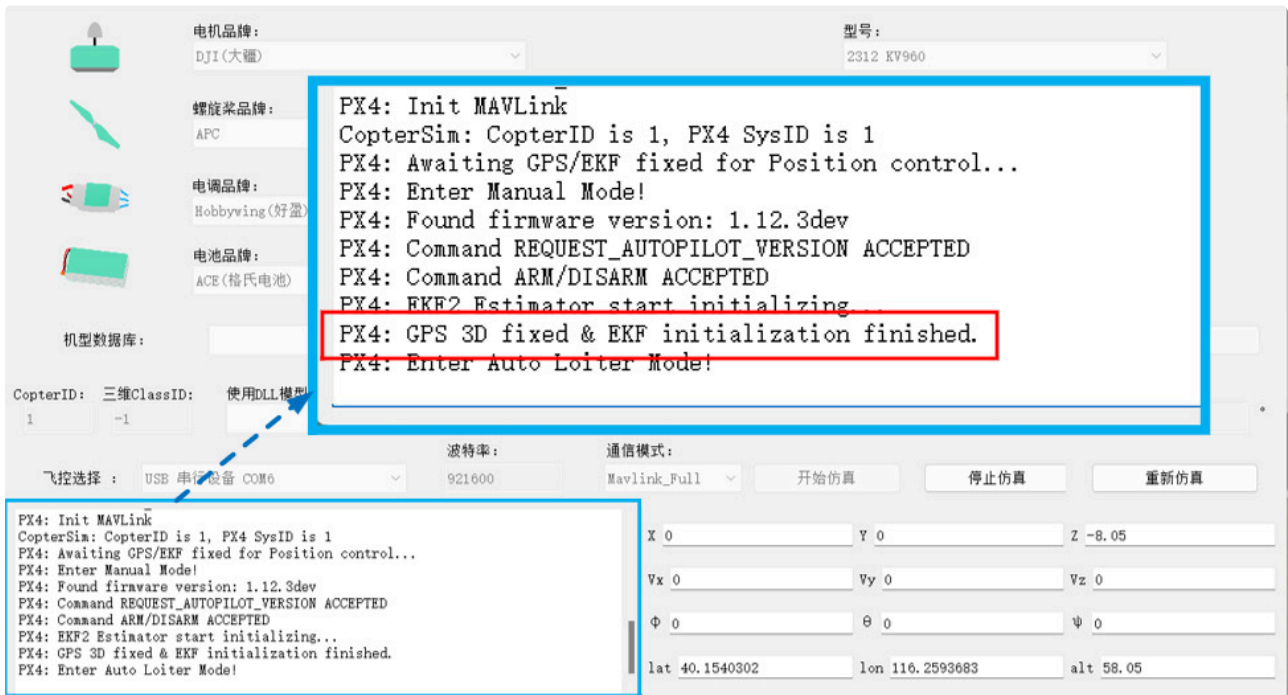
[安装目录]\RflySimAPIs\6.RflySimExtCtrl\2.AdvExps\4.TrajPlanExps\2.MutilPointTrajMotion

- [Python38Run.bat](#)：Python运行脚本，用于启动Python控制程序。
- [SITLRun.bat](#)：仿真环境启动脚本，用于启动SITL仿真。
- [UAVCtrl.py](#)：主要控制程序，实现多点轨迹规划与速度控制算法，包括航点管理、速度控制策略、动态偏航角控制和减速机制等功能模块。

4. 实验内容或步骤

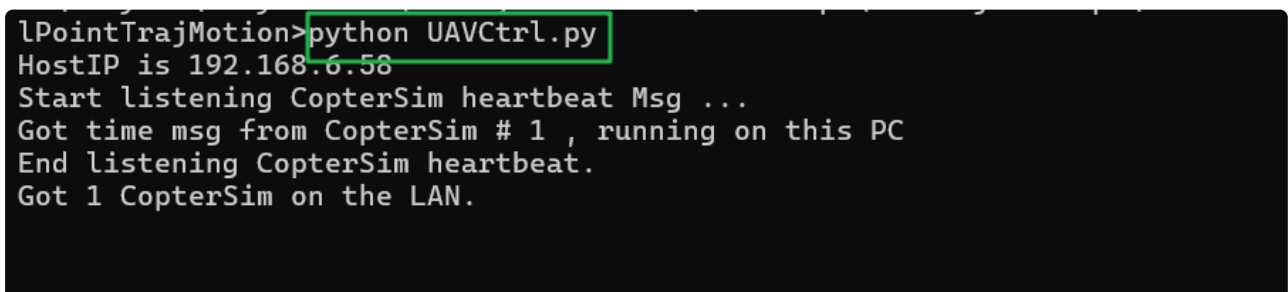
4.1 步骤1：初始化仿真环境

双击运行 `SITLRun.bat` 文件，等待仿真环境初始化完成。脚本将会启动 1 个 QGC 地面站，1 个 CopterSim、1 个 RflySim3D 软件，等待CopterSim软件下侧日志栏必须打印出 GPS 3D fixed & EKF initialization finished 字样代表初始化完成。如下图所示：



4.2 步骤2：运行控制程序

双击运行 `Python38Run.bat` 脚本，在弹出的对话框重视输入：`python UAVCtrl.py`。



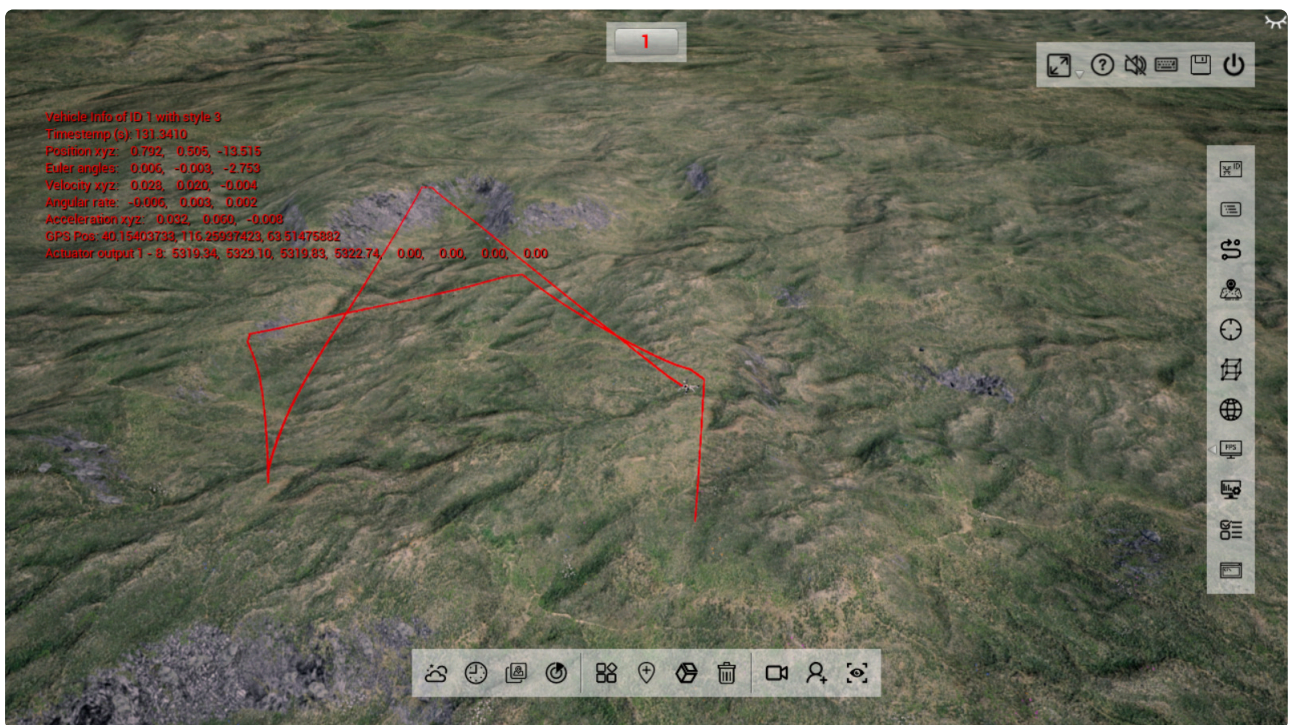
4.3 步骤3：观察仿真

可以看到无人机开始解锁并上升到10米高度后，依次按照(10, 5, -10)、(15, -3, -10)、(8, -8, -10)、(20, 10, -10)、(0, 0, -10) 点飞行，然后降落，

在RflySim3D界面下依次按下"D"、"T"将显示无人机飞行状态量和飞行轨迹。

```
C:\WINDOWS\system32\cmd. x + v

进度：35.9%，偏航：26.5°，偏航率：0.2°/s
进度：53.9%，偏航：26.6°，偏航率：0.0°/s
进度：71.9%，偏航：26.6°，偏航率：0.0°/s
进度：89.8%，偏航：26.6°，偏航率：0.0°/s
到达航点 2
第 2 段：(10, 5) -> (15, -3)
距离：9.4m，时间：4.7s
当前偏航：26.6°，目标偏航：-58.0°
进度：0.0%，偏航：25.0°，偏航率：-45.8°/s
进度：21.3%，偏航：-20.8°，偏航率：-45.8°/s
进度：42.6%，偏航：-55.1°，偏航率：-9.8°/s
进度：63.8%，偏航：-57.9°，偏航率：-0.4°/s
进度：85.1%，偏航：-58.0°，偏航率：-0.0°/s
到达航点 3
第 3 段：(15, -3) -> (8, -8)
距离：8.6m，时间：4.3s
当前偏航：-58.0°，目标偏航：-144.5°
进度：0.0%，偏航：-59.5°，偏航率：-45.8°/s
进度：23.3%，偏航：-105.4°，偏航率：-45.8°/s
进度：46.5%，偏航：-141.1°，偏航率：-11.2°/s
进度：69.8%，偏航：-144.3°，偏航率：-0.5°/s
进度：93.0%，偏航：-144.5°，偏航率：-0.0°/s
到达航点 4
第 4 段：(8, -8) -> (20, 10)
距离：21.6m，时间：10.8s
当前偏航：-144.5°，目标偏航：56.3°
进度：0.0%，偏航：-146.0°，偏航率：-45.8°/s
进度：9.3%，偏航：168.2°，偏航率：-45.8°/s
进度：18.5%，偏航：122.3°，偏航率：-45.8°/s
```



4.4 步骤4：实验总结

通过RflySim3D可视化界面，可以清楚地观察到无人机的飞行轨迹呈现出由多个直线段组成的复杂路径，验证了多点轨迹规划算法和速度控制策略的有效性。相较于简单的直线轨迹，该实验展示了更复杂的轨迹跟踪能力。

5. 关键知识点

5.1 关键知识点1：多点轨迹规划与航点管理

在 `UAVCtrl.py` 程序中，通过定义一个包含6个三维坐标点的 `waypoints`列表来实现多点轨迹规划。程序使用循环遍历相邻的航点对，将整个轨迹分解为5个独立的直线段进行处理。每个航点都包含北向(N)、东向(E)和地向(D)三个坐标分量，其中D分量保持恒定(-10米)以维持飞行高度。

5.2 关键知识点2：速度控制策略实现

程序采用速度控制而非位置控制来实现轨迹跟踪，这是通过 `mav.SendVelNED()`函数实现的。在每个轨迹段中，程序首先计算起点到终点的方向向量(dx, dy)，然后将其归一化得到单位方向向量(dir_x, dir_y)。通过将单位方向向量与巡航速度 `CRUISE_SPEED`相乘，得到速度指令(vx, vy)。这种方法相比位置控制能够实现更平滑的轨迹跟踪，特别是在航点转换处避免了急停急启现象。

5.3 关键知识点3：动态偏航角控制算法

程序实现了复杂的动态偏航角控制机制。首先通过 `math.atan2(dy, dx)`计算目标偏航角，使其始终指向运动方向。然后通过P控制器计算偏航角速度指令：`yaw_rate = yaw_error * 3.0`，并使用 `max(min(yaw_rate, MAX_YAW_RATE), -MAX_YAW_RATE)`限制最大偏航角速度。程序还特别处理了角度环绕问题，确保偏航角在 $[-\pi, \pi]$ 范围内。

5.4 关键知识点4：航点减速与平滑停止机制

在每个航点到达前，程序实现了一个10步的减速过程。通过 `decel_factor = 1.0 - (decel_step / 10.0)`计算递减因子，使速度逐步降低至0。同时在减速过程中继续调整偏航角，确保无人机在航点处的姿态正确。这种机制显著提高了飞行的平稳性和安全性，避免了突然停止带来的不稳定现象。

5.5 关键知识点5：控制频率与时序管理

程序采用30Hz的控制频率，通过`time.sleep(1/control_freq)`实现精确的时间控制。在每个轨迹段开始前，根据距离和巡航速度计算飞行时间，并据此确定控制步数。程序还实现了进度监控和状态打印功能，便于调试和观察飞行状态。

6. 参考资料

1. 《RflySim用户手册》：提供RflySim平台的全面使用指导，包括仿真环境搭建、API接口说明等内容。
2. 《PX4开发者指南》：详细介绍了PX4开源飞控系统的架构、参数设置和开发方法。
3. 《Python编程：从入门到实践》：Python语言编程参考书籍，涵盖基础语法和实际应用。

7. 常见问题

Q1：运行Python控制程序时出现连接错误，无法控制无人机飞行？

A1：检查SITL仿真环境是否已完全启动，确认CopterSim日志中显示"GPS 3D fixed & EKF initialization finished"后再运行控制程序。同时确认端口设置正确，默认为UDP端口9002。

Q2：无人机在航点切换时反应迟钝或轨迹偏离预期路线？

A2：这可能是由于控制频率过低或速度设定过高导致的。尝试调整代码中的`control_freq`参数提高控制频率，或适当降低`CRUISE_SPEED`值以改善轨迹跟踪精度。

Q3：无人机在航点附近未能正确减速或停留时间过长？

A3：检查代码中`decel_step`参数的设置，确保减速过程的步数适中。若减速过程太短可能导致急停现象，太长则影响效率。可根据实际飞行情况调整`decel_factor`的计算公式。

1. <https://rflysim.com/> ↩

2. 推荐配置请见: <https://rflysim.com/doc/zh/HowToInstall.pdf> ↩