

# 圆形轨迹规划与实时速度控制实验

## 1. 实验目的

本实验旨在通过控制无人机执行圆形轨迹飞行任务，帮助用户掌握以下技能和知识：

1. 理解圆形轨迹规划的基本原理，掌握连续曲线轨迹的参数化表示方法；
2. 掌握基于实时速度控制的无人机轨迹跟踪技术，学习如何通过速度指令实现平滑的曲线飞行；
3. 学习角速度控制和切向速度计算方法，理解速度与角速度之间的关系；
4. 理解实时轨迹生成的概念，掌握在飞行过程中动态计算轨迹参数的方法；
5. 掌握Offboard模式下实现复杂连续轨迹控制的完整流程，理解实时控制的重要性。

## 2. 实验要求

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

## 3. 实验地址

例程路径：

[安装目录]\RflySimAPIs\6.RflySimExtCtrl\2.AdvExps\e4.TrajPlanExps\3.CircularTrajMotion

- [UAVCtrl.py](#)：主控制程序，实现圆形轨迹规划与实时速度控制逻辑；
- [SITLRun.bat](#)：仿真环境启动脚本，启动QGC地面站、CopterSim和RflySim3D；
- [Python38Run.bat](#)：Python环境运行脚本，用于启动控制程序；

## 4. 实验内容或步骤

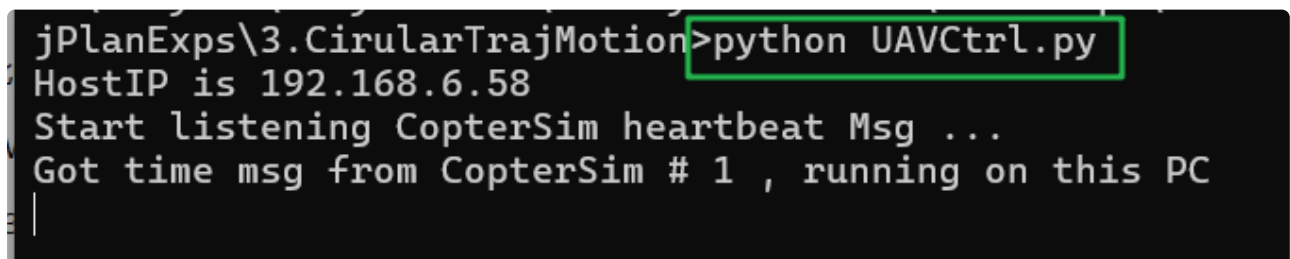
### 4.1 步骤1：初始化仿真环境

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

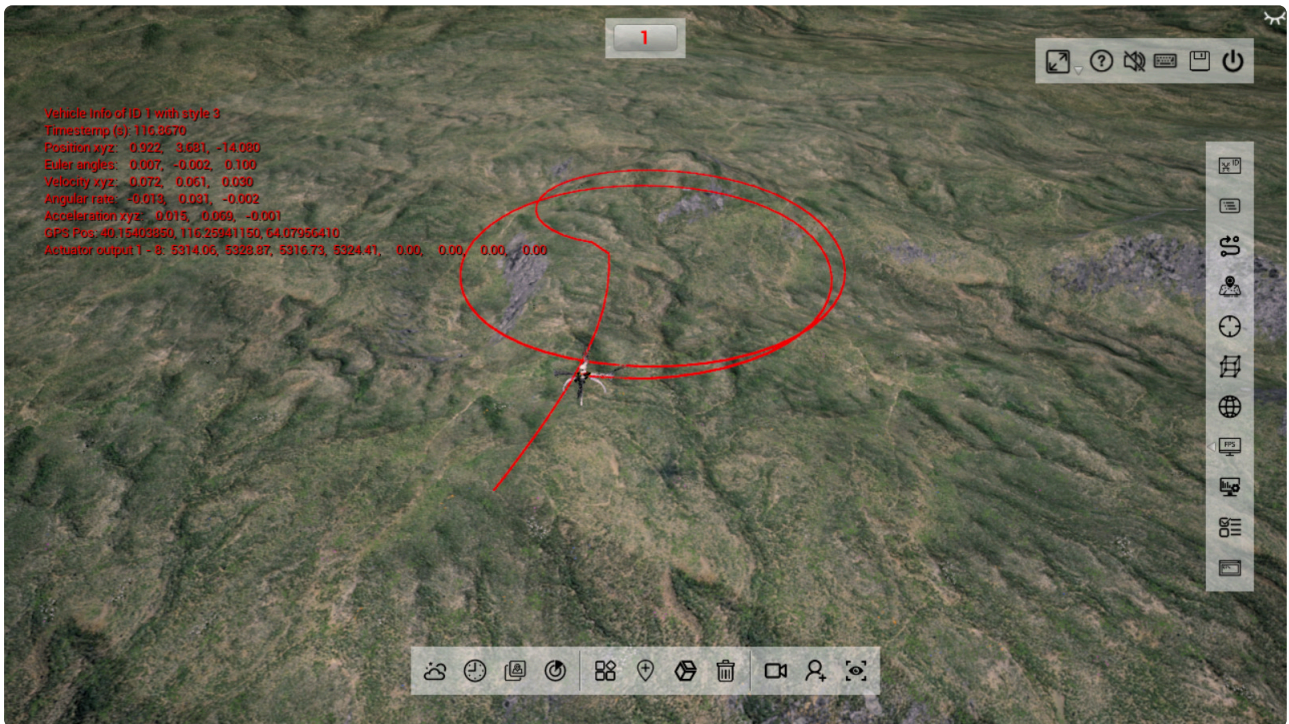
可以看到无人机开始解锁并上升到10米高度后，开始按照圆形轨迹飞行，然后降落，

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。

```

jPlanExps\3.CircularTrajMotion>python UAVCtrl.py
HostIP is 192.168.6.58
Start listening CopterSim heartbeat Msg ...
Got time msg from CopterSim # 1 , running on this PC
End listening CopterSim heartbeat.
Got 1 CopterSim on the LAN.
起飞到起始点: (6.0, 0.0)
开始实时圆形轨迹控制...
轨迹参数: 半径=6.0m, 角速度=22.9°/s, 持续时间=30.0s
时间: 1.0s, 位置: (5.5, -2.4), 速度: 2.40m/s
时间: 2.0s, 位置: (4.1, -4.3), 速度: 2.40m/s
时间: 3.0s, 位置: (2.1, -5.6), 速度: 2.40m/s
时间: 4.0s, 位置: (-0.3, -6.0), 速度: 2.40m/s
时间: 5.1s, 位置: (-2.6, -5.4), 速度: 2.40m/s
时间: 6.1s, 位置: (-4.5, -3.9), 速度: 2.40m/s
时间: 7.1s, 位置: (-5.7, -1.8), 速度: 2.40m/s

```

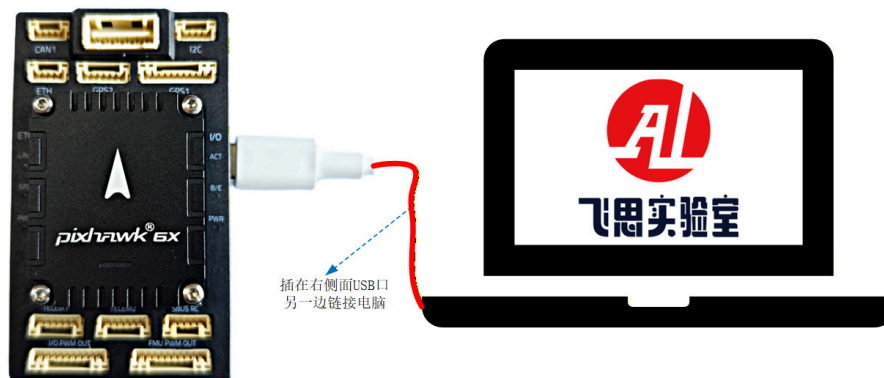


SIL标准流程如下（以下流程用于直接复制使用，若实验中无相关步骤，可删除如下内容）：  
 1、双击运行SITLRun.bat文件，等待仿真环境初始化完成。脚本将会启动1个QGC地面站，1个CopterSim、1个RflySim3D软件，等待CopterSim软件下侧日志栏必须打印出GPS 3D fixed & EKF initialization finished 字样代表初始化完成。如下图所示：



2、本实验在进行之前需要将飞控内部固件进行还原并需确保可正常进行硬件在环仿真，具体还原方式可见例

程：`[RflySim安装路径]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e2.FCUIntro`。按照如下图所示，进行硬件链接并接入电脑。



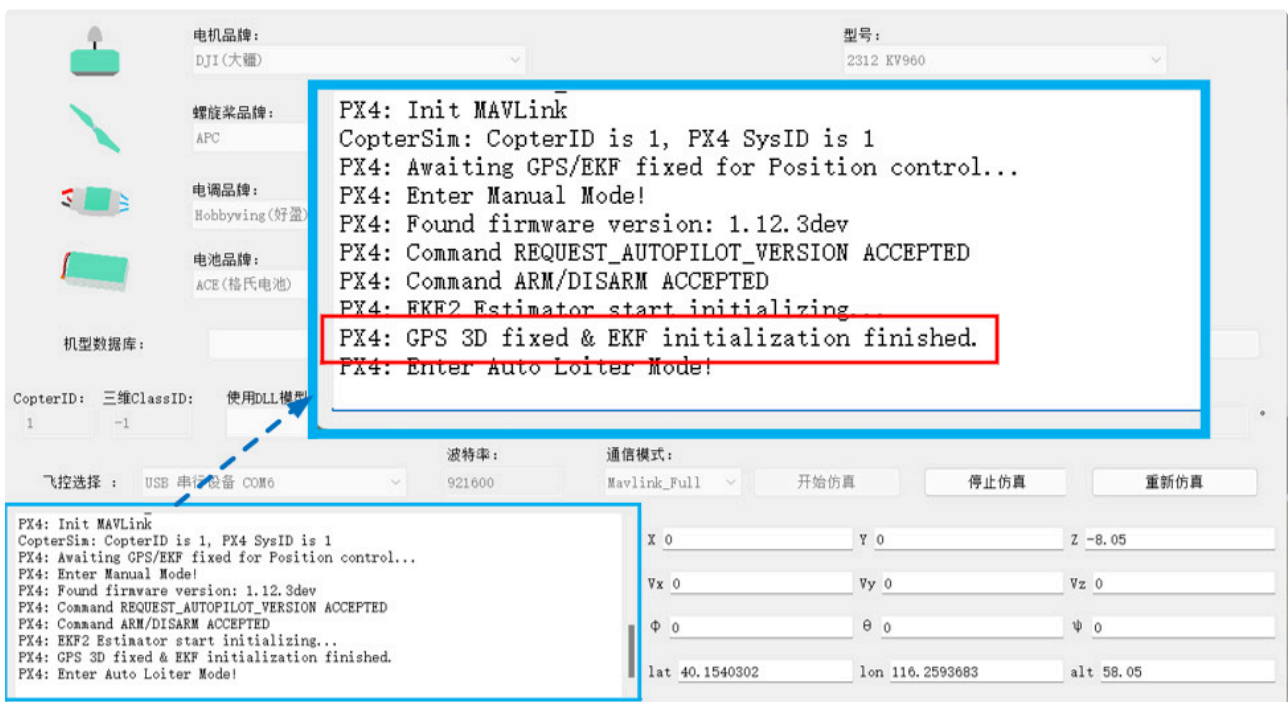
双击运行HITLRun.bat文件，在弹出得对话框中输入飞控COM号如：4。

```
C:\WINDOWS\system32\cmd.exe

-----
Please input the Pixhawk COM port list for HITL
Use ',' as the separator if more than one Pixhawk
E.g., input 3 for COM3 of Pixhawk on the computer
Input 3,6,7 for COM3, COM6 and COM7 of Pixhawks
-----
All COM ports on this computer are:

COM3: Intel(R) Active Management Technology - SOL (unavailable or busy)
COM4: USB 串行设备 * (Pixhawk with SysID=1)
-----
Recommended COM list input is: 4
-----
My COM list for HITL simulation is: _
```

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：



## 5. 关键知识点

### 5.1 关键知识点1：圆形轨迹规划原理与参数化表示

在 `UAVCtrl.py` 程序中，通过 `[generate_circular_trajectory_realtime()]` 函数实现圆形轨迹的参数化表示。该函数定义了圆的半径(radius): 圆形轨迹的半径参数；圆心坐标 (center\_x, center\_y): 定义圆形轨迹的中心位置；飞行高度(height): 飞行高度参数；角速度

(angular\_velocity): 控制无人机在圆周上的角速度。与传统的航点式轨迹不同，圆形轨迹是一种连续曲线轨迹，需要通过参数方程来描述。程序将圆形轨迹表示为时间的函数，通过角速度和时间计算当前角度，进而得到无人机在任意时刻的期望位置和速度。

## 5.2 关键知识点2：实时速度控制策略实现

程序采用实时速度控制而非位置控制来实现轨迹跟踪，这是通过 [mav.SendVelNEDNoYaw()] 函数实现的。在每个控制周期内，程序调用 [get\_circular\_velocity\_at\_time()] 函数计算当前时刻的速度指令。该函数根据角速度和半径计算切向速度大小，然后通过三角函数分解为北向和东向速度分量。这种方法相比位置控制能够实现更平滑的轨迹跟踪，避免了位置控制中可能出现的轨迹跟踪误差和抖动现象。

## 5.3 关键知识点3：切向速度与角速度关系

程序基于物理学中的圆周运动原理，利用切向速度公式  $v = \omega \times r$  计算无人机的速度大小。其中  $v$  是切向速度； $\omega$  是角速度； $r$  是圆半径。通过这种方式，程序可以保持恒定的角速度飞行，使得无人机在圆周上各点的运动速率一致。这种控制方式更符合自然的圆周运动规律，能够实现平稳的圆形轨迹飞行。

## 5.4 关键知识点4：实时轨迹生成与控制频率管理

程序采用实时轨迹生成机制，不需要预先计算所有轨迹点，而是在每个控制周期内根据当前时间计算期望的速度指令。通过 [time.time()] 获取系统时间，并结合轨迹开始时间计算已用时间，进而确定当前应处的位置和速度。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，保证了轨迹跟踪的精度和实时性。

## 5.5 关键知识点5：起始点定位与轨迹初始化

程序首先将无人机飞到圆形轨迹的起始点，即圆上角度为 $0^\circ$ 的点，坐标为 (center\_x + radius, center\_y)。这一步骤确保了无人机从正确的初始位置开始执行圆形轨迹。随后程序记录轨迹开始时间，并进入实时控制循环。这种初始化方式保证了轨迹执行的准确性和连续性。

# 6. 参考资料

1. [RflySim官方文档](#)
2. [PX4官方文档](#)

## 7. 常见问题

### Q1: CopterSim软件没有打印"GPS 3D fixed & EKF initialization finished"信息怎么办?

A1: 检查电脑是否已经连接到互联网，因为首次启动仿真需要下载地图。如果长时间没有显示，请尝试重新运行 [SITLRun.bat](#) 脚本重启仿真环境。

### Q2: 无人机无法按照圆形轨迹飞行，轨迹偏离严重?

A2: 可能的原因包括控制频率过低或参数设置不当。请确认代码中的控制频率为30Hz，并检查圆形轨迹参数（半径、角速度）设置是否合理。同时确保仿真环境中没有其他干扰因素。

### Q3: Python控制程序运行时报错或无法连接到仿真环境?

A3: 请先确认仿真环境完全启动后再运行Python程序。如果仍然报错，检查Python环境是否正确配置，确保所需依赖库（如pymavlink）已正确安装。另外，防火墙设置也可能影响程序与仿真环境的通信。

---

1. <https://rflysim.com/> ↩

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