

1. 实验名称及目的

1.1 实验名称

200机质点模型集群实验（仅限完整版及以上版本）

1.2 实验目的

本实验旨在通过结合高精度的6DOF模型（CopterSim）与真实飞控系统（PX4），采用软/硬件在环仿真闭环方式，提高模型的准确性和可信度，从而减少仿真结果与实际飞行实验之间的差距。实验将基于RflySim平台实现30架质点模型的四旋翼飞机的起飞和圆形飞行任务。通过这一方法，旨在评估高精度模型与真实飞控系统结合对四旋翼飞机飞行性能和控制精度的提升效果，并对比仿真结果与实际飞行数据的差异。

1.3 关键知识点

从模型精度的角度，使用高精度6DOF模型（CopterSim）+真实飞控系统（PX4）的软/硬件在环仿真闭环的方式，能够有效提高模型可信度，从而减小仿真与真机实验的差距。本实验基于RflySim平台实现在局域网内两台电脑200架质点模型的四旋翼飞机起飞和画圆飞行。

为了提高单台电脑仿真集群飞机的数量，就需要降低模型精度并使用简化飞控模型。因此本平台在Python下开发出了质点多旋翼模型，只需Python和RflySim3D两个软件即可在单台电脑上实现百驾级别的无人机集群仿真。

2. 实验效果

可实现局域网内两台电脑200个飞机先起飞并悬停，接着开始同步画圆。



图 1 实验效果

3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\10.RflySimSwarm\3.CustExps\e4.NoPX4GMLSSwarm\e2_NoPX4SITL200Swarm\3.NoPX4SITL200Swarm2PC
 \3.NoPX4SITL200Swarm2PC\](file:///C:\Users\uavcs\Desktop\1.SwarmLogGet)

表 1 文件目录

文件夹/文件名称	说明
NoPX4SITL200.bat	启动仿真配置文件
NoPX4SITL200PC1.py	电脑A实现功能主文件
NoPX4SITL200PC2.py	电脑B实现功能主文件
Readme.pdf	用户指南

4. 运行环境

表 2 运行环境

4.1 软件要求

Win 10/Win11系统；RflySim工具链。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4_fmu-v6x_default，推荐PX4固件版本为：1.12.3。其他配套飞控及编译命令请见：

<https://rflysim.com/doc/zh/1/Hardware.html>

4.2 硬件要求

笔记本/台式电脑① 2台。

①：推荐配置请见：<https://rflysim.com/>

1. **： **推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

5. 实验步骤

5.1 必做实验

Step 1: 电脑A仿真配置

在电脑A上执行NoPX4SITL200.bat文件。只会出现RflySim3D软件。



图 2 电脑A仿真配置

Step 2: 电脑B仿真配置

在电脑B上执行NoPX4SITL200.bat文件。只会出现RflySim3D软件。



图 3 电脑B仿真配置

Step 3: 电脑B启动仿真

在电脑B文件夹中，双击Python38Run.bat，打开集成好的python环境，在该环境下运行NoPX4SITL200PC2.py文件，输入

python

NoPX4SITL200PC2.py，接着按回车运行，即可在RflySim3D中看到生成的100架飞机，编号是从101~200。并在Python38Run中显示：Waiting for start Msg。



图 4 电脑B初始化

Step 4: 电脑A启动仿真

在电脑A文件夹中，双击Python38Run.bat，打开集成好的python环境，在该环境下运行NoPX4SITL200PC1.py文件，输入

python

NoPX4SITL200PC1.py，接着按回车运行，即可在RflySim3D中看到生成的100架飞机，编号是从1~100，同时启动仿真。



图 5 电脑A初始化

Step 5: 观察实验效果

仿真开始后，可以在电脑A、B中看到200架飞机同时起飞并进行同时各自画圆。



图 6 实验效果

5.2 选做实验（VS Code调试运行）

Step 1: 准备工作

先确保已经按

[\[RflySim安装目录\]/RflySimAPIs/1.RflySimIntro/2.AdvExps/e3.PythonConfig/Readme.pdf](#)

步骤，正确配置VS

Code环境。或者配置了自己的Pycharm等自定义Python环境。

Step 2: VS code调试运行

其他步骤与上文相同，在Step 3电脑B启动仿真时，在电脑B打开VS

code找到本实验路径文件夹，运行NoPX4SITL200PC2.py文件，会在RflySim3D中看到生

成的100架飞机，编号是从101~200。并在VS

code终端中显示：Waiting for start Msg。

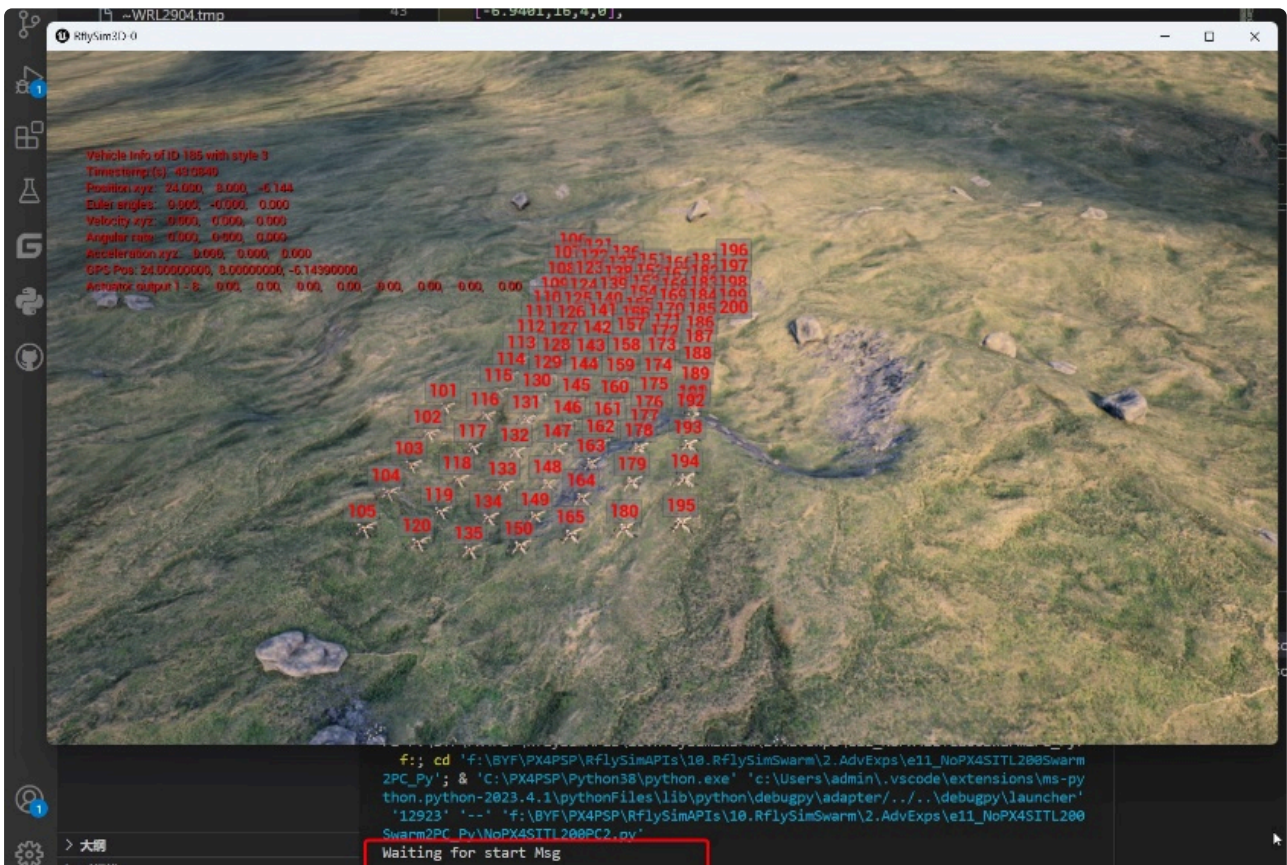


图 7 VS code调试

在Step 4电脑A启动仿真时，在电脑A打开VS

code本实验路径文件夹，运行NoPX4SITL200PC1.py文件，会在RflySim3D中看到生成的

100架飞机，编号是从1~100，同时启动仿真。



图 8 启动仿真

6. 参考资料

7. 下面以单个飞机为例，介绍控制流程：

```

mav = PX4MavCtrl.PX4MavCtrl(20100) # 创建一号飞机实例
mav.initPointMassModel(-8.086,[0,0,0]) # 初始化质点模型循环
print((mav.uavPosNED,mav.truePosNED, # 打印数据
mav.SendPosNED(0, 0, -1.7, 0) # 发送目标位置
mav.SendMavArm(True) # 解锁飞控，飞机起飞
time.sleep(5) # 代码暂停5s，飞机到达起飞点并悬停
mav.SendVelNED(0, 0, 1, 0) # 发送向下速度，飞机降落
mav.EndPointMassModel() # 退出质点模型循环

```

2. 代码解析如下：（与4飞机例子-

```

*\PX4PSP\RflySimAPIs\10.RflySimSwarm\1.BasicExps\e9_NoPX4SITL4Swarm
-区别部分):

```

```

MavList=MavList+[PX4MavCtrl.PX4MavCtrler(20100+ii*2)] #建立飞机实例矩阵

InitPosList=[ *****] # 配置飞机初值矩阵

MavList[i].initPointMassModel(InitPosList[i][0],InitPosList[i][1:4])
#通过矩阵初始化

Error2UE4Map = Error2UE4Map+[***] #
计算每个飞机起飞坐标系与UE4地图坐标系差值

MavList[0].sendUE4Cmd(b 'RflyChangeViewKeyCmd S') # RflySim3D显示飞机数字标
号

MavList[0].sendUE4Cmd(b 'RflyChangeViewKeyCmd T') # RflySim3D显示飞机轨迹

MavList[i].SendPosNED(0, 0, -10, 0) # 飞机各自起飞到10m高（以起飞点为坐标系）

targetPosE=np.array([-0,0,-15]) # 设置默认高度15m，所有飞机汇集到本坐标

targetPosE=np.array([10*math.sin(t/2+math.pi/2)-10,10*math.sin(t/2.0),-15]) #
生成圆形轨迹

targetPosE=targetPosE+Error2UE4Map[j] # 将圆形轨迹映射到各飞机起飞坐标系

mav.SendPosNED(targetPosE[0],targetPosE[1],targetPosE[2],0) #发送圆形轨迹

MavList[i].EndPointMassModel() # 各飞机退出仿真循环

```

3. 关键代码解析

200个飞机的初始位置和地形高度生成：见

*\PX4PSP\RflySimAPIs\10.RflySimSwarm\0.ApiExps\5.GetTerrainAPI例程，关键点重复两次100飞机位置生成代码，配置“VehicleNum=100;TOTOAL_COPTER=200;”，针对电脑1设置“START_INDEX=1;”，针对电脑2设置“START_INDEX=101;”，运行GenSwarmPos2PC200.m可以得到两电脑的InitPosList列表

NoPX4SITL200PC1.py设置“START_INDEX=1”和NoPX4SITL200PC2.py设置“START_INDEX=101”，同时初始化脚本使用广播地址“255.255.255.255”

MavList=MavList+[PX4MavCtrl.PX4MavCtrler(20100+ii*2,'255.255.255.255')]

targetPosE不经过Error2UE4Map[j]的校准，每个飞机画圆按照自己起飞坐标系来
NoPX4SITL200PC2.py中“MavList[0].waitForStartMsg()”，启用程序阻塞，等到接收到来自局域网的开始仿真消息后，才会继续执行后续的控制算法

NoPX4SITL200PC1.py中“MavList[0].sendStartMsg()”，发送开始仿真消息，局域网内所有等待的程序收到本消息后，会自动开始后续程序运行。

7.常见问题

Q1：无

A1：无