

# 1. 实验名称及目的

## 1.1 实验名称

4机质点模型集群实验。

## 1.2 实验目的

本实验旨在通过使用高精度的6DOF模型（CopterSim）与真实飞控系统（PX4）进行软/硬件在环仿真闭环，以提高模型的可信度，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平台实现四旋翼飞机的四架质点模型，并验证其在起飞、悬停几秒钟后下降的动态行为。

## 1.3 关键知识点

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



图 1 实验效果

## 3. 文件目录

例程目录：

[[安装目录](#)]\RflySimAPIs\10.RflySimSwarm\0.ApiExps\e2.PyRflySwarmAPIExps\2.QuadGMAPIExp

e2\_NoPX4SITL4Swarm\](file:///C:\Users\uavcs\Desktop\1.SwarmLogGet)

表 1 文件目录

文件夹/文件名称	说明
<a href="#">NoPX4SITL4Swarm.bat</a>	启动仿真配置文件
<a href="#">NoPX4SITL4Swarm.py</a>	实现功能主文件
<a href="#">Python38Run.bat</a>	Python环境启动脚本
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 1台。

①：推荐配置请见：<https://rflysim.com/>

1. \*\*： \*\*推荐配置请见：<https://rflysim.com/doc/zh/1/InstallLearn.html>

## 5. 实验步骤

### 5.1 必做实验

#### Step 1: 开启一个软件在环仿真并等待初始化完毕

双击运行 `NoPX4SITL4Swarm.bat` 启动仿真脚本，会自动开启RflySim3D软件。



图 2 RflySim3D等待界面

## Step 2: 启动仿真

在文件夹下，双击 [Python38Run.bat](#)，打开集成好的python环境，在该环境下运行 [NoPX4SITL4Swarm.py](#) 文件，输入

python

[NoPX4SITL4Swarm.py](#)，接着按回车即可启动仿真。仿真开始后，即可看到RflySim3D中出现4架无人机，4架无人机会先起飞然后降落，同时Python38Run中会出现无人机的仿真状态数据。

```
C:\windows\system32\cmd.exe
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

D:\10.RflySimSwarm\1.BasicExps\e2_NoPX4SITL4Swarm>python NoPX4SITL4Swarm.py
[[0.0, 0.0, 0.0], [0.0, 0.0, -8.086], [0.0, 0.0, 0.0], [0.0, -0.0, 0.0], [0, 0, 0.0], [0.0, 0.0, -8.086]]
[[0.0, 0.0, 0.0], [2.0, 0.0, -8.29], [0.0, 0.0, 0.0], [0.0, -0.0, 0.0], [0, 0, 0.0], [2.0, 0.0, -8.29]]
[[0.0, 0.0, 0.0], [0.0, 2.0, -7.835], [0.0, 0.0, 0.0], [0.0, -0.0, 0.0], [0, 0, 0.0], [0, 0, 0, 0, 0, 0, 0, 0], [0.0,
2.0, -7.835]]
[[0.0, 0.0, 0.0], [2.0, 2.0, -7.788], [0.0, 0.0, 0.0], [0.0, -0.0, 0.0], [0, 0, 0.0], [0, 0, 0, 0, 0, 0, 0, 0], [2.0,
2.0, -7.788]]
Send target Pos
Send Arm Command
Send Velocity Speed
End pointMass model

D:\10.RflySimSwarm\1.BasicExps\e2_NoPX4SITL4Swarm>
```

图 3 Python38Run运行示例

## Step 3: 观察实验效果



图 4 RflySim3D展示效果

## 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Step2启动仿真时，用VS code打开到本实验路径文件夹，运行

[NoPX4SITL4Swarm.py](#)

文件，启动仿真。仿真开始后，即可看到RflySim3D中出现4架无人机，4架无人机会先起飞然后降落，同时VS

code终端中会出现无人机的仿真状态数据。

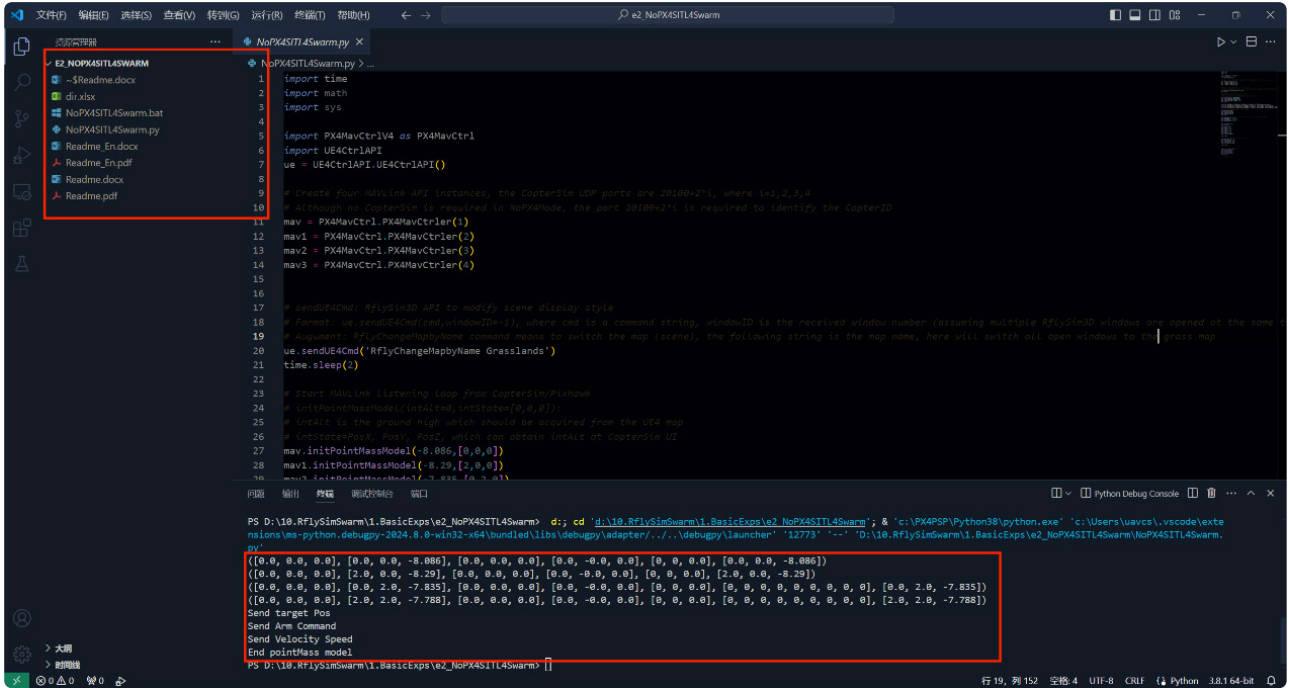


图 5 VS code运行示例

## 6.参考资料

7. 由于本例程没有bat脚本和CopterSim来配置飞机的初始位置和地形高度，需要手动在初始化脚本中设置，首先需要选定四个飞机的初始位置；本例程采用和bat脚本一样的矩形分布，即X和Y的位置点为(0,0)、(2,0)、(0,2)、(2,2)，将上述坐标输入CopterSim中，可以获取地形高度，然后将其输入到Python脚本的initPointMassModel()函数中。

8. 本实验与

\*\PX4PSP\RflySimAPIs\10.RflySimSwarm\1.BasicExps\e7\_MAVLinkFull4Swarm实验的区别在于以下几点：

- 增加切换RflySim3D地图的代码mav.sendUE4Cmd(b'RflyChangeMapbyName Grasslands')
- 去掉InitMavLoop()和initOffboard()初始化代码，使用质点模型初始化代码（包含设置地形高度、xy位置和偏航）initPointMassModel(intAlt=0,intState=[0,0,0])
- 其余状态获取、速度和位置指定发送函数保持相同

3. 下面以单个飞机为例，介绍控制流程：

4. mav = PX4MavCtrl.PX4MavCtrl(20100) # 创建一号飞机实例

5. mav.initPointMassModel(-8.086,[0,0,0]) # 初始化质点模型循环

6. print((mav.uavPosNED,mav.truePosNED, # 打印数据

7. mav.SendPosNED(0, 0, -1.7, 0) # 发送目标位置
8. mav.SendMavArm(True) # 解锁飞控，飞机起飞
9. time.sleep(5) # 代码暂停5s，飞机到达起飞点并悬停
10. mav.SendVelNED(0, 0, 1, 0) # 发送向下速度，飞机降落
11. mav.EndPointMassModel() # 退出质点模型循环

## 7. 常见问题

Q1: \*\*\*

A1: \*\*\*