

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\1.BasicExps\](#)
[e2_NoPX4SITL4Swarm\](#)

表 1 文件目录

文件夹/文件名称	说明
----------	----

NoPX4SITL4Swarm.bat	启动仿真配置文件
NoPX4SITL4Swarm.py	实现功能主文件
Python38Run.bat	Python 环境启动脚本
Readme.pdf	用户指南

4. 运行环境

表 2 运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Win 10/Win11 系统	笔记本/台式电脑 ^①	1
2	RflySim 工具链		1
3	Visual Studio Code	注：选做	1

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: 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 x + v
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>
```

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。

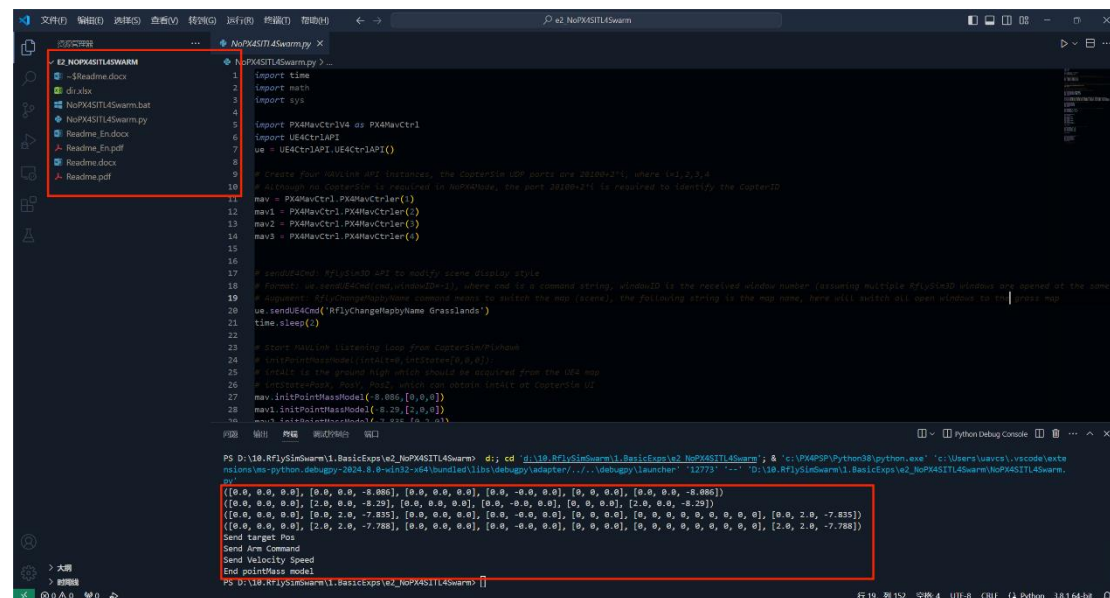


图 5 VS code 运行示例

6. 参考资料

[1] 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。

[2] 本实验与*PX4PSP\RflySimAPIs\10.RflySimSwarm\1.BasicExps\7_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: ***