

1. 实验名称及目的

1.1 实验名称

四旋翼分布式轨迹跟踪实验 (2Win or

4Win): 该实验基于领导跟随法的四旋翼无人机轨迹跟踪实验 (仅限完整版及以上版本)

1.2 实验目的

本章实验的主要目的是介绍DistSim远程控制节点的功能，以使用户在进行单人操作远程多机节点时提高效率。

该实验基于领导跟随法的四旋翼无人机编队，领导跟随法是一种常见的编队控制方法，其基本思想是选取一架无人机作为领导者，其他无人机作为跟随者，通过控制跟随者的位置和速度，使整个编队保持一定的队形和运动状态。在这种方法中，领导者无人机的位置和速度由任务需求确定并通过UDP组网发送给各个跟随者无人机，而跟随者无人机的控制则通过接收前一架无人机的信息并计算得到。

本实验中1号领导者旋翼机接收到起飞指令后先垂直起飞，2、3、4号跟随者旋翼机接收到前列旋翼机的起飞数据后，依次间隔3s起飞。1号领导者旋翼机垂直起飞至10m后实时检测所有跟随者旋翼机数据，并等待其他跟随者旋翼机垂直起飞至同等高度后，开始进入路线规划。

1.3 关键知识点

实验中主要通过Python接口PX4MavCtrlV4、UE4CtrlAPI、NetSimAPIV4等，控制飞行器飞行。具体实现思路如下：

下述介绍代码说明。

UAV1Ctrl.py代码简介：

本段代码的主要功能是获取特定 ID 的 CopterSim 的计算机 IP 地址，并配置该计算机将模拟数据重定向到本机，以便进行进一步的控制和监测。这可以简化网络配置的过程，优化无人机模拟的操作。创建1号飞行器通信mav、ue，与

```
CopterSim、RflySim3D相连copterSimId
```

```
= 1;
```

```
ue = UE4CtrlAPI.UE4CtrlAPI()
```

```
#
```

```
创建一个CopterSim状态获取实例，并监听2s钟，获取局域网内所有CopterSim所在电脑的IP
```

```
req = ReqCopterSim.ReqCopterSim()
```

```
## 获取目标电脑IP，并且配置CopterSim回传数据到本电脑
```

```
# 获取到指定CopterID的CopterSim所在电脑的IP
```

```
#
```

```
注：Windows下运行本函数获取TargetIP=127.0.0.1，如果在其他电脑上运行，获取CopterID对应电脑的IP
```

```
TargetIP = req.getSimIpID(copterSimId)
```

```
# 请求目标CopterSim将数据返回到本电脑
```

```
# 通过本接口，可以不用再去bat脚本里面填写IP地址了
```

```
req.sendReSimIP(copterSimId)
```

本段代码的整体功能是初始化无人机控制器和网络模拟接口。mav

实例用于控制无人机的飞行，而 net

实例则用于管理与模拟环境之间的网络通信。这为后续的无人机控制和数据交互提供了基础；

```
mav = PX4MavCtrl.PX4MavCtrlr(copterSimId,TargetIP)
```

```
net = NetSimAPIV4.NetSimAPI(mav)
```

本函数的主要作用是每隔 1

秒打印一次引导无人机和跟随者无人机的状态信息。通过这种方式，可以实时监控无人机的状态和行为，便于调试和分析飞行情况；

```
def func():
```

```
# 每隔1s打印指定飞机的数据
```

```
while True:
```

```
if copterSimId != 1:
```

```
print('跟随者状态信息: ',net.UavData[0].CopterID,net.UavData[0].uavAngEular,net.UavData[0].uavVelNED,net.UavData[0].uavAngEular,net.UavData[0].uavPosNED,net.UavData[0].uavGlobalPos)
```

```
print('领航者状态信息:',mav.CopterID,mav.uavAngEular,mav.uavVelNED,mav.uavAngEular,mav.uavPosNED,mav.uavGlobalPos)
```

```
time.sleep(1)
```

本段代码的整体功能是初始化与 PX4 无人机的 MAVLink 通信，并将本机的数据根据指定的 IP 和端口转发给特定的无人机。这种设置允许在多架无人机之间进行同步和数据交换；

```
#mav.InitMavLoop(UDPMODE), where UDPMODE=0,1,2,3,4
```

```
# Use UDP_Simple Mode to control PX4
```

```
# In this mode, this script will send struct inHILCMDData to CopterSim
```

```
# Then CopterSim convert it to MAVLinkOffboard message to PX4
```

```
# In this mode, PX4 send MAVLink data to CopterSim, which convert Struct data to this script
```

```
# Obviously, UDP Simple is faster (Little data, low delay) than UDP Simple.
```

```
# 这里使用MAVLink_Full模式来传输数据
```

```
mav.InitMavLoop()
```

```
# 将本机数据转发给'224.0.0.10' 下的 60002,60003,60004系列端口
```

```
# 这几个端口目前会被#2 #3 #4号飞机订阅，因此实际上是发给了#2 #3 #4号飞机
```

```
# 如果是组网仿真的话，先要发给组网程序的指定IP和端口，再转发到各飞机监听端口
```

```
#net.enNetForward([60002,60003,60004],'224.0.0.10')
```

```
net.enNetForward([60000 + copterSimId +1],'224.0.0.10')
```

本段代码的主要功能是检查无人机的状态，确保其准备好飞行后，启用 Offboard 控制。通过这种方式，用户可以在确认无人机状态良好后，开始对其进行控制和指令发送；

```

print('Check if CopterSim 3D Fixed...')

while True:

if mav.isPX4EKF3DFixed:

print('CopterSim/PX4 3D Fixed, ready to fly.')

break

time.sleep(0.5)

time.sleep(1)

print(copterSimId)

print('开始起飞： Start Offboard Send!')

# 启用Offboard控制

mav.initOffboard()

time.sleep(1)

```

本行代码的主要作用是启动网络接收功能，使程序能够在计算机上监听从 IP 地址 224.0.0.10 发送到 60000 + copterSimId 端口的数据流。这对于实时接收无人机或者其他相关设备的信息至关重要，能够使程序获取并处理从网络传入的数据；

```
net.StartNetRec(60000 + copterSimId,'224.0.0.10')
```

本段代码根据无人机的 ID

分别处理引导无人机和跟随无人机的起飞过程。引导无人机直接解锁并起飞，而跟随无人机则等待接收引导无人机的数据后再进行相应的解锁和起飞操作。这种设计确保了多架无人机之间的协调和同步。如果为1号飞行器，则发送ue指令给RflySim3D进行飞行器数量显示和飞行器航迹显示，并解锁起飞，垂直起飞至十米后处于等待模式；对于其他 ID 的无人机，则需要等待接收到数据后再执行相同的起飞命令；

```

time.sleep(1)

if copterSimId == 1:

ue.sendUE4Cmd('RflyChangeViewKeyCmd S')

ue.sendUE4Cmd('RflyChangeViewKeyCmd T 4')

```

```
print('尝试重新解锁飞机')

mav.SendMavArm(True)

print('起飞到十米高')

mav.SendPosNED(0, 0, -10, 0)

else:

# 本例子期望接收领航者飞机的数据

while True:

if net.UavData[0].CopterID == copterSimId-1:

break

time.sleep(2)

print('开始执行航迹模式')

print('尝试重新解锁飞机')

mav.SendMavArm(True)

print('起飞到十米高')

mav.SendPosNED(0, 0, -10, 0)
```

本段代码的主要目的是启动一个新的线程，用于并行执行 func 函数的代码，而不影响主程序的其他操作。多线程可以用来执行耗时的任务而不冻结主界面或允许其他操作同时进行；

```
dance_thread = threading.Thread(target=func)

dance_thread.start()

time.sleep(0.5)
```

本段代码实现了一个基本的无人机控制逻辑。引导无人机在前 10 秒内向下飞行到指定高度，然后进入一个圆形轨迹飞行；而跟随无人机则根据领航无人机的位置动态调整自身速度，以保持跟随状态。整个过程通过简易定时器确保以 30Hz 的频率执行；

```
lastTime = time.time()
```

```

startTime = time.time()

timeInterval = 1/30.0 #here is 0.0333s (30Hz)

if copterSimId == 1:

    Error2UE4Map=[]

    Error2UE4Map =
    Error2UE4Map+[-np.array([mav.uavGlobalPos[0]-
    mav.uavPosNED[0],mav.uavGlobalPos[1]-mav.uavPosNED[1],mav.uavGlobalPos[2]-
    mav.uavPosNED[2]])]

    while True:

        lastTime = lastTime + timeInterval

        sleepTime = lastTime - time.time()

        if sleepTime > 0:

            time.sleep(sleepTime) # 未到期望时间，就休息

        else:

            lastTime = time.time()

            # 上面的程序创建了一个简易定时器，保证代码以30Hz频率执行循环

            # 在下面开发你的控制算法

            t=time.time()-startTime

            # target position in UE4 map global frame

            if t<10:

                targetPosE=np.array([-0,0,-20]) # fly to 15 meters high in 0 to 10s

            else: #fly circle after 10s

                targetPosE=np.array([10*math.sin(t/2+math.pi/2)-10,10*math.sin(t/2.0),-15])

            # target position in vehilce takeoff frame

            targetPosE=targetPosE+Error2UE4Map[0]

```

```
mav.SendPosNED(targetPosE[0],targetPosE[1],targetPosE[2],0)
```

```
else:
```

```
while True:
```

```
lastTime = lastTime + timeInterval
```

```
sleepTime = lastTime - time.time()
```

```
if sleepTime > 0:
```

```
time.sleep(sleepTime) # 未到期望时间，就休息
```

```
else:
```

```
lastTime = time.time()
```

```
# 上面的程序创建了一个简易定时器，保证代码以30Hz频率执行循环
```

```
mav.SendVelNED(net.UavData[0].uavGlobalPos[0]-mav.uavGlobalPos[0]
```

```
,net.UavData[0].uavGlobalPos[1]-
```

```
mav.uavGlobalPos[1],net.UavData[0].uavGlobalPos[2]-mav.uavGlobalPos[2],
```

```
0) 1号飞行器解锁具备起飞状态，飞行器进入外部模式（offboard）；
```

UavCmdTest.py代码简介：

该命令的目的是在新的命令提示符窗口中运行一个名为 UavPythonRunALL.bat 的批处理文件。这个批处理文件可能用于启动与无人机模拟或控制相关的程序或服务。

```
# 需要发送指令 UavCmdTest.Py
```

```
cmdResultData = "start cmd /k start
```

```
C:\\PX4PSP\\RflySimAPIs\\10.RflySimSwarm\\3.CustExps\\e2.PyDISCtrlUAVsSim\\e4_
```

```
FourUAVSwarmDistSim\\UavPythonRunALL.bat"
```

```
# 需要发送指令 UavCmdMuliTest.Py
```

```
cmdResultData = 'start cmd /k start C:\\PX4PSP\\Python38\\python.exe
```

```
C:\\PX4PSP\\RflySimAPIs\\10.RflySimSwarm\\3.CustExps\\e2.PyDISCtrlUAVsSim\\e4_
```

```
FourUAVSwarmDistSim\\'
```

```
cmdResultData1 = "start cmd /k start
```

```
C:\\PX4PSP\\RflySimAPIs\\10.RflySimSwarm\\3.CustExps\\e2.PyDISCtrlUAVsSim\\e4_
```

```
FourUAVSwarmDistSim\\SITLRun4MavlinkFull.bat"
```

此代码的主要功能是扫描特定网络并列出现所有发现的主机的信息。每个主机的特点包括：

序号：主机在列表中的位置，易于识别。

主机ID：唯一标识每个主机的ID。

系统类型：指示主机操作系统是Linux还是Windows。

主机名称：主机的名称。

```
#获取组网列表
```

```
scan_Networking_list = distSimCtrlAPI().scan_udp();
```

```
for i in range(len(scan_Networking_list)):
```

```
if scan_Networking_list[i].WinOrLinux != 0:
```

```
print(f"序号：{i + 1}, 主机ID：{scan_Networking_list[i].NodeID}, 系统类型：Linux  
, 主机名称：{scan_Networking_list[i].hostname}")
```

```
else:
```

```
print(f"序号：{i + 1},主机ID：{scan_Networking_list[i].NodeID}, 系统类型：Win ,  
主机名称：{scan_Networking_list[i].hostname}")
```

本段代码是一个简单而有效的用户输入验证程序，用于：

从用户获取输入。

将输入转换为适合程序可以处理的形式。

验证输入的有效性，避免超出列表范围的错误。

处理输入过程中的异常情况，提升代码的健壮性和用户体验。

```
# 用户输入序号
```

```
try:
```

```
index =
```

```
int(input("请选择运行RflySim平台的主机，输入上述检索序号（从1开始，例如1, 2, 3,  
...）：")) - 1 # 转换为0-index
```

```
# 检查索引是否有效
```

```
if 0 <= index < len(scan_Networking_list):
```

```
matched_id = scan_Networking_list[index].NodeID
```

```
print(f"序号 {index + 1} 对应的ID是: {matched_id}")
```

```
else:
```

```
print("序号超出范围，请输入一个有效的序号。")
```

```
except ValueError:
```

```
print("请输入一个有效的整数序号")
```

本行代码发送测试指令给当前局域网指定的计算机。在 RflySim 平台启动后，通过循环执行一系列 UAV 控制脚本。它确保在进行这些操作之前，软件已经准备就绪。

```
# 等待用户输入
```

```
input("等待RflySim平台正常启动后，回车执行下一步操作")
```

```
for i in range(len(scan_Networking_list)):
```

```
distSimCtrlAPI().execute_bat_script(cmdResultData +  
"\\UAV"+str(i+1)+" Ctrl.py",scan_Networking_list[i].NodeID)
```

2. 实验效果

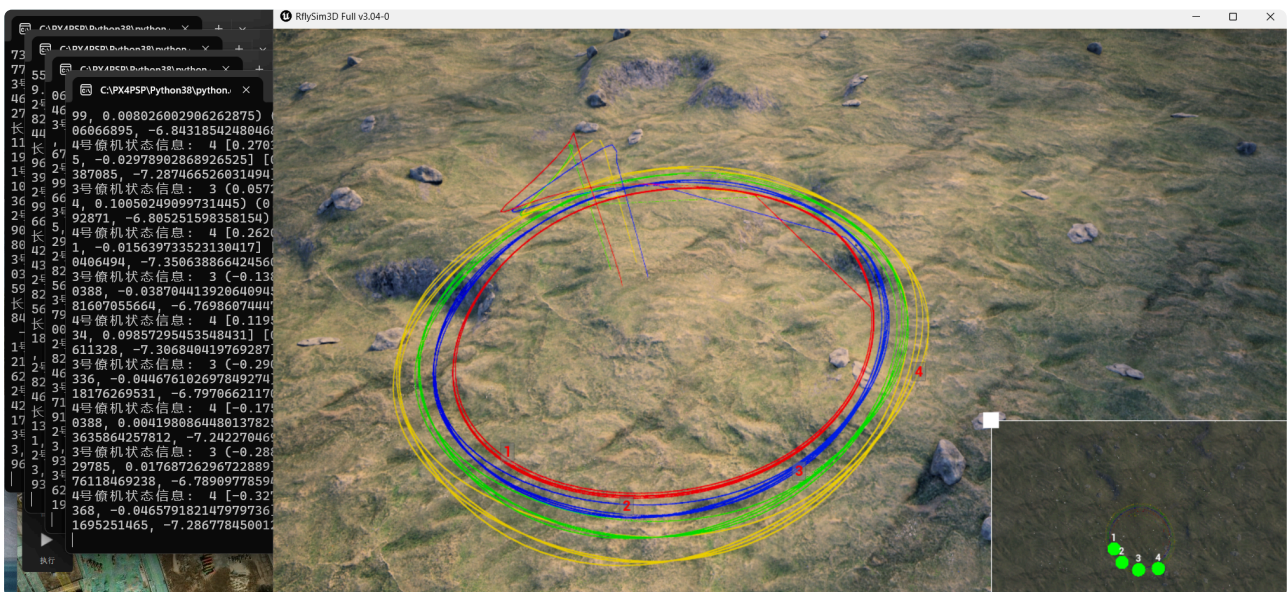


图 1 实验效果

3. 文件目录

文件夹/文件名称	说明
Python38Run.bat	Python 环境启动脚本
SITLRun4MavlinkFull.bat	软件在环仿真脚本
UavCmdTest.py	远程启动指令脚本
UavCmdMMultiTest.py	远程启动指令脚本
UAV1Ctrl.py	领航者控制程序
UAV2Ctrl.py	跟随者1号控制程序
UAV3Ctrl.py	跟随者2号控制程序
UAV4Ctrl.py	跟随者3号控制程序
UavPythonRunALL.bat	启动控制程序脚本
Readme.pdf	用户指南

4. 运行环境

4.1 软件要求

Windows 10及以上版本；RflySim平台完整版及以上；MATLAB R2022B。

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

台式电脑/笔记本 1台。

①：推荐配置请见：<https://rflysim.com/>

5.实验步骤

DistSim两机实验

5.1.1实验连接方式

图 2 硬件连接方式

5.1.2实验操作

主机①运行：

以管理员身份运行SITLRun4MavlinkFull.bat，启动仿真脚本，等待旋翼机飞行器初始化完成，即CopterSim界面返回语句“GPS 3D fixed & EKF initialization finished”和“Enter Auto Loiter Mode”即可。



图 3 CopterSim等待界面



图 5 DistSim软件界面

主机②运行:

主机②运行桌面RflySimTools中的DistSim。

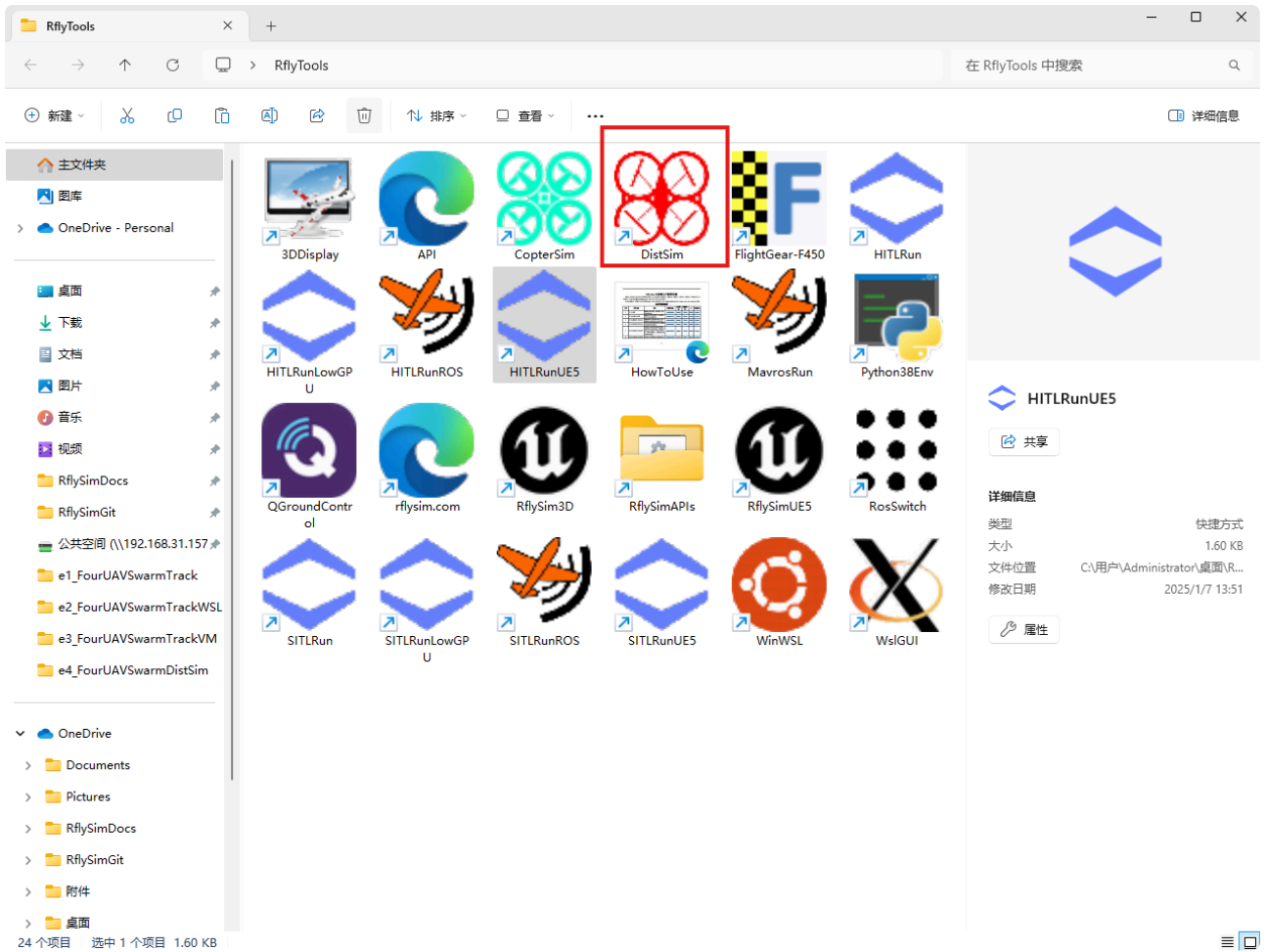


图 6 RflySimTools工具链功能快捷键



图 7 主机②distSim界面

DistSim启动后可在软件中看到主机①的ip端口信息。主机①的DistSim可以看到主机②的ip端口信息；



图 8 主机①DistSim显示界面

5.1.3 远端发送指令

在主机②打开与主机①同目录下的实验例程：安装目录
\\PX4PSP\RflySimAPIs\10.RflySimSwarm\3.CustExps\e2.PyDISCtrlUAVsSim\e4_FourUAVSwarmDistSim。

主机②通过Visual Studio [Code运行UavCmdTest.py](#)。根据提示输入检索出的序号，回车。

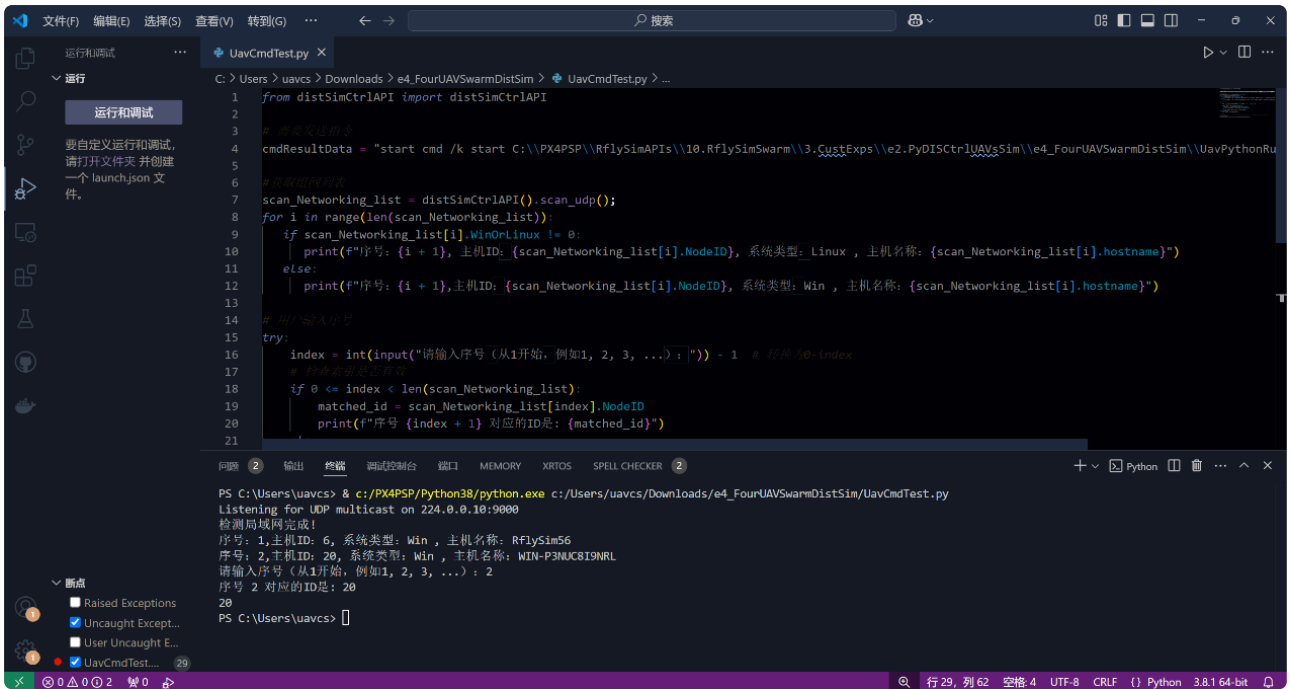


图 9 Visual Studio Code执行自动化脚本文件

可看到主机①打开执行窗口，自动执行UavPythonRunALL.bat文件，如图 10所示。

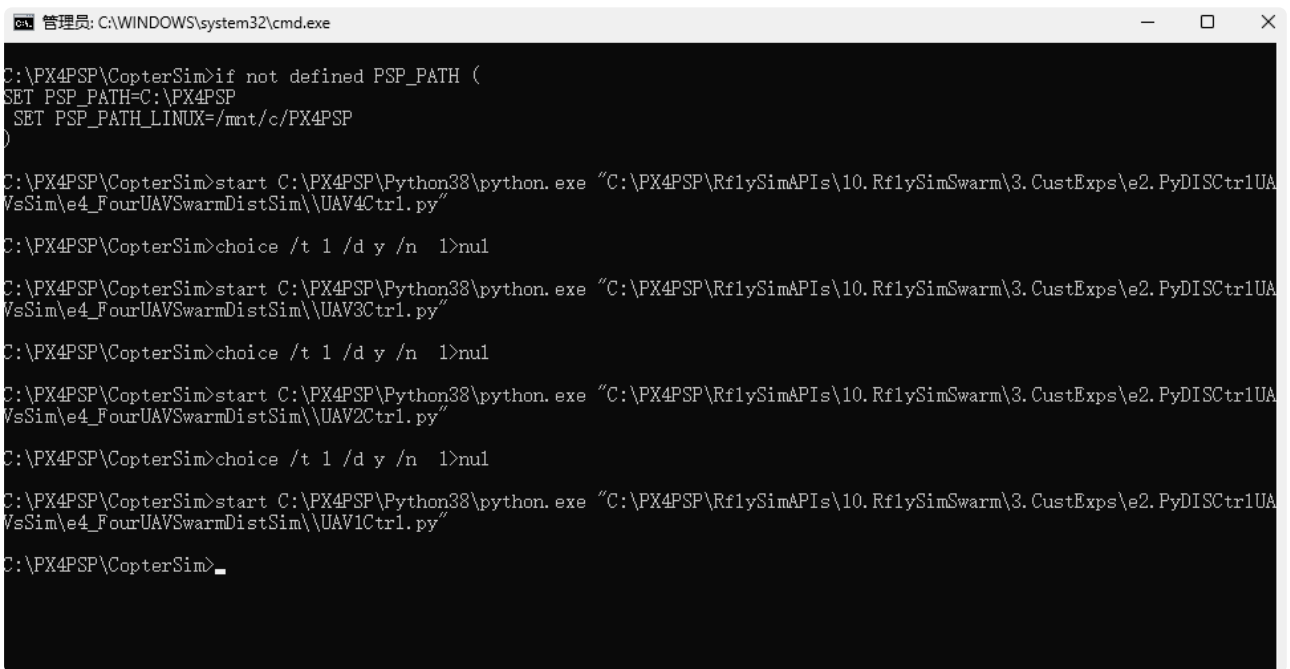


图 10 主机①执行UavCmdTest.py脚本

```
18 C:\PX4\Python38\python
38 C:\PX4\Python38\python
27 C:\PX4\Python38\python
85 C:\PX4\Python38\python
84 4号值机状态信息: 4 [-0.0016961246728897095, -0.0064117396250367165, 0.0020983577705919743] [-0.029784422367811203, 0.02
72 7715057134628296, -2.1420512199401855] [-0.0016961246728897095, -0.0064117396250367165, 0.0020983577705919743] [-0.01105
1 2250862121582, -0.008252652361989021, -1.138553261756897] [1.9876428767282492, 1.9769638587614338, -8.981552639382816]
79 3号值机状态信息: 3 (-0.0019485465018078685, -0.0017940407851710916, 0.00041039736242964864) (0.0069496408104896545, 0.0
03 014953295467421412, -2.7590744495391846) (-0.0019485465018078685, -0.0017940407851710916, 0.00041039736242964864) (-0.01
2 578819751739502, 0.012263753451406956, -6.191514492034912) (1.9829067658602568, 0.003743510212546264, -14.34051417824311
67 9)
77 4号值机状态信息: 4 [0.0016132843447849154, -0.0014429939910769463, 0.00017707339429762214] [0.007101243827491999, 0.000
5) 4734742396976799, -2.601773738861084] [0.0016132843447849154, -0.0014429939910769463, 0.00017707339429762214] [-0.003558
3 994038030505, -0.006153383292257786, -3.661099672317505] [1.9951361335523403, 1.979063127831165, -11.504099049943424]
94 3号值机状态信息: 3 (0.002276388928294182, -0.000515249150339514, 0.0004906977410428226) (0.008127948269248009, -0.02079
06 4784650206566, -1.5053926706314087) (0.002276388928294182, -0.000515249150339514, 0.0004906977410428226) (-0.00440258672
42 4609137, 0.0012685168767347932, -8.468419075012207) (1.9942923766530427, -0.007251726362125899, -16.617418761220414)
15 4号值机状态信息: 4 [-0.002369684400036931, -0.0006839717389084399, 0.0006120600155554712] [0.004448528867214918, -0.008
17 354198187589645, -2.760632038116455] [-0.002369684400036931, -0.0006839717389084399, 0.0006120600155554712] [0.006038515
00 828549862, 0.017057640478014946, -6.2575297355651855] [2.0047336434189207, 2.002274151601438, -14.100529113191104]
03 3号值机状态信息: 3 (0.0029352728743106127, -0.00023928451992105693, -0.0005052810884080827) (0.004697184078395367, -0.0
24 05094936117529869, -0.5980961322784424) (0.0029352728743106127, -0.00023928451992105693, -0.0005052810884080827) (-0.000
90 6673317402601242, -0.00631852587684989, -9.327807426452637) (1.9980276316373917, -0.014838769115710582, -17.476807112660
2 843)
15 4号值机状态信息: 4 [0.0012916279956698418, 0.0019182272953912616, 0.00018970767268911004] [0.0034569636918604374, -0.01
36 818380318582058, -1.474865198135376] [0.0012916279956698418, 0.0019182272953912616, 0.00018970767268911004] [0.025242194
31 533348083, 0.002337016398087144, -8.491278648376465] [2.023937322123719, 1.98755352752151, -16.334278026002384]
3 81 3号值机状态信息: 3 (0.003308999352157116, 3.859584830934182e-05, 0.0002689578686840832) (-0.004398096352815628, 0.00199
52 20426420867443, -0.32940712571144104) (0.003308999352157116, 3.859584830934182e-05, 0.0002689578686840832) (0.0083458507
13 42353367, -0.013040877878665924, -9.725537300109863) (2.007040814121005, -0.021561121117526616, -17.87453698631807)
75 4号值机状态信息: 4 [0.0018635233864188194, 0.00031158275669440627, 0.0003390918718650937] [-0.020194776356220245, -0.00
59 62578520737588406, -0.5882262587547302] [0.0018635233864188194, 0.00031158275669440627, 0.0003390918718650937] [0.026931
11 656524538994, -0.006103155668824911, -9.338369369506836] [2.02562678411491, 1.9791133554545979, -17.181368747132755]
```

图 11控制程序临近感知信息

5.1.4观察状态- RflySim3D

观察RflySim3D查看各个旋翼机飞行状态。

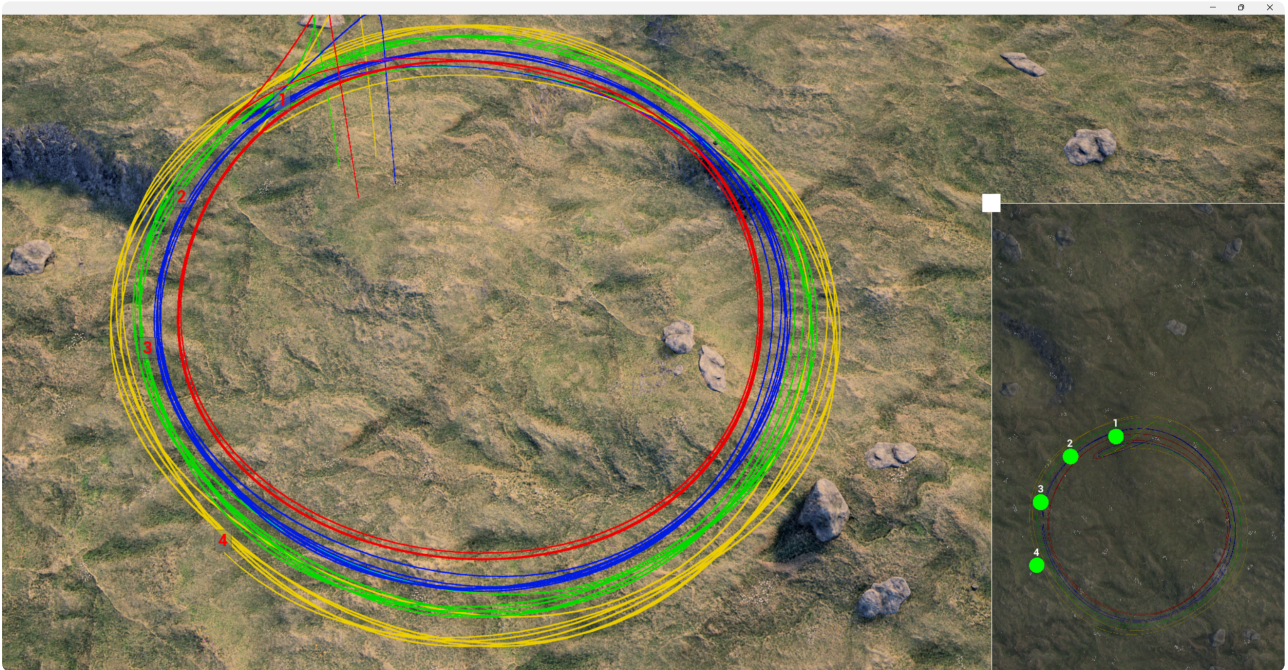


图 12 RflySim3D展示效果

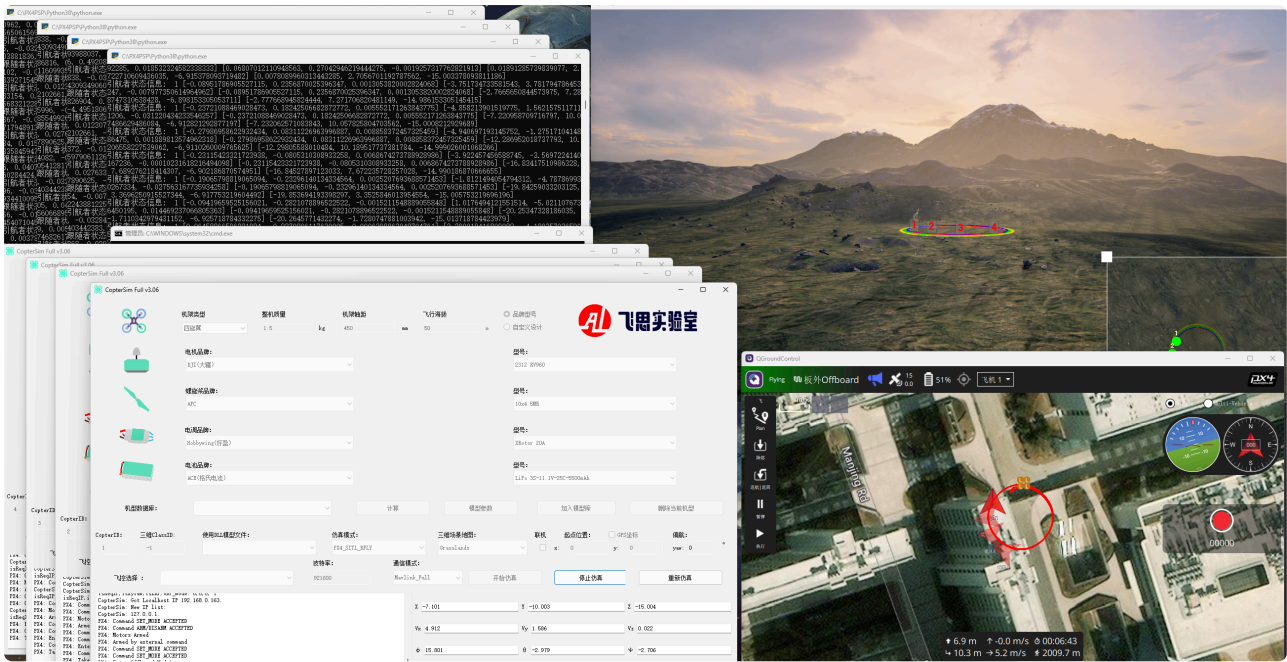


图 13 RflySim工具链实验阶段

5.1.5 结束实验- CMD

在启动的命令窗口按下回车键（任意键）就能快速关闭CopterSim、QGC、RflySim3D等所有程序。

DistSim多机实验

5.2.1 实验连接方式

图 14 硬件连接方式

5.2.2 实验操作

所有主机均运行桌面RflySimTools中的DistSim。

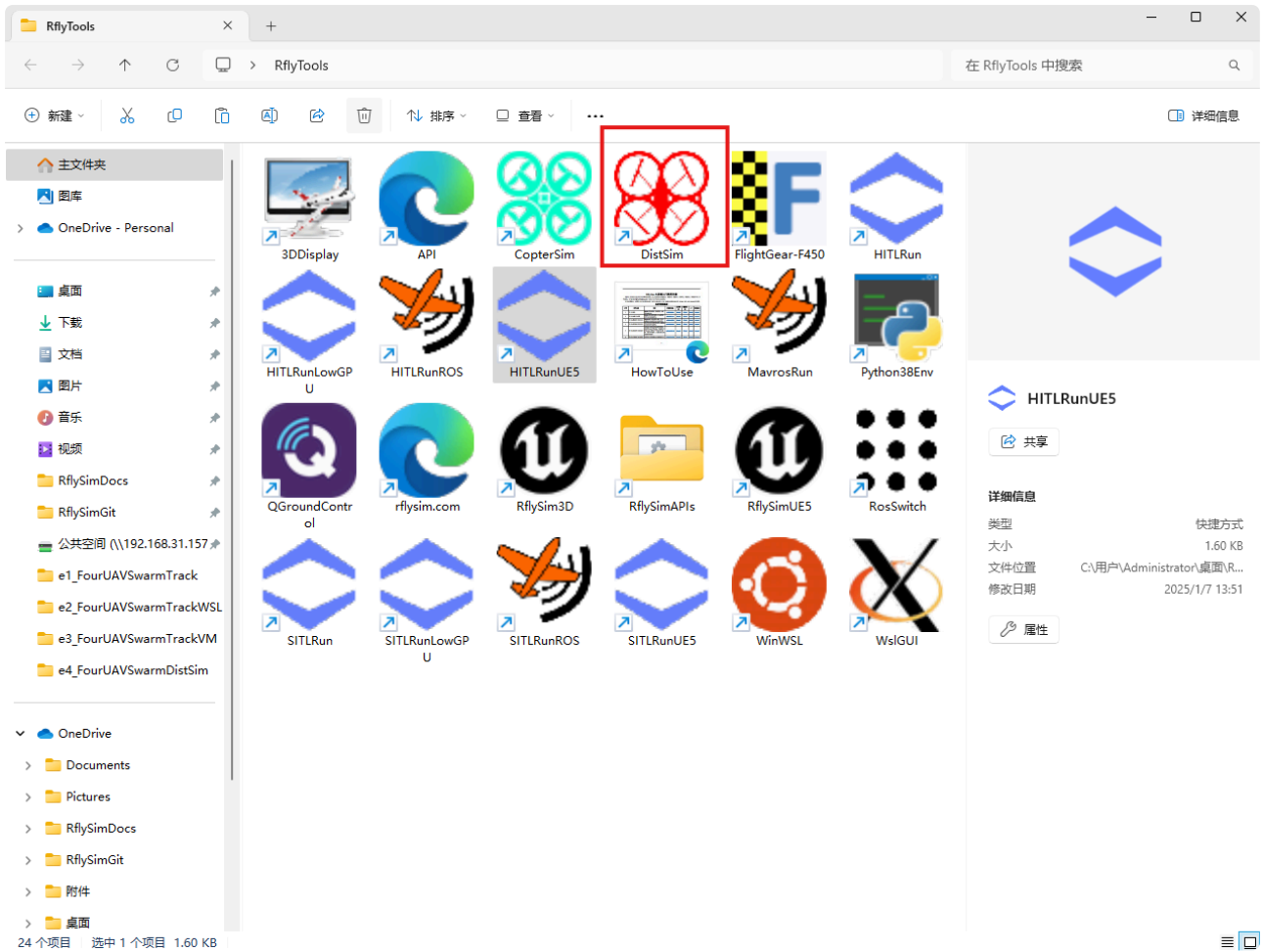


图 15 RflySimTools工具链功能快捷键



图 16 主机①DistSim显示界面示例

5.2.3 远端发送给指令

主机①通过Visual Studio

Code运行UavCmdMuliTest.py文件，发送给局域网内所有仿真测试主机。局域网内仿真测试主机DistSim接收到指令后，执行指令操作。

执行第一步，在检索到的局域网主机中，选择运行RflySim平台的仿真测试主机，本小节实验选择主机①运行RflySim平台。

```

UavCmdMultiTest.py x
UavCmdMultiTest.py >
cmdResultData1 = startCmd /k start C:\PX4SP\Python38\python.exe f:\10.RflySimSwarm\3.CustExps\e2.PyDISCtrlUAVsSim\e4_FourUAVSwarmDistSim\UavCmdMultiTest.py
6
7
8 (variable) scan_Networking_list: list
9
10 scan_Networking_list = distSimCtrlAPI().scan_udp();
11 for i in range(len(scan_Networking_list)):
12     if scan_Networking_list[i].WinOrLinux != 0:
13         print(f"序号: {i + 1}, 主机ID: {scan_Networking_list[i].NodeID}, 系统类型: Linux, 主机名称: {scan_Networking_list[i].hostname}")
14     else:
15         print(f"序号: {i + 1}, 主机ID: {scan_Networking_list[i].NodeID}, 系统类型: Win, 主机名称: {scan_Networking_list[i].hostname}")
16
17
18 # 用户输入序号
19
20 try:
21     index = int(input("请选择运行RflySim平台的主机，输入上述搜索序号(从1开始，例如1, 2, 3, ...) : ")) - 1 # 转换为0-index
22     # 检查索引是否有效
23     if 0 <= index < len(scan_Networking_list):
24         matched_id = scan_Networking_list[index].NodeID
25         print(f"序号 {index + 1} 对应的ID是: {matched_id}")
26     else:
27         print("序号超出范围，请输入一个有效的序号。")
28 except ValueError:
29     print("请输入一个有效的整数序号")
30
31 #发送指令
32 #startComputer_id_list = input("请输入上述局域网内指定的计算机发送测试指令 (按下Enter表示确认): ")
33 print(matched_id)
34 distSimCtrlAPI().execute_bat_script(cmdResultData1, matched_id)
35
36 # 等待用户输入
37 input("等待RflySim平台正常启动后，回车执行下一步操作")
38 for i in range(len(scan_Networking_list)):
39     distSimCtrlAPI().execute_bat_script(cmdResultData + "\\UAV"+str(i+1)+"Ctrl.py", scan_Networking_list[i].NodeID)
40 #distSimCtrlAPI().execute_bat_script(cmdResultData, 20)
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

图 17 UavCmdMultiTest.py示例代码

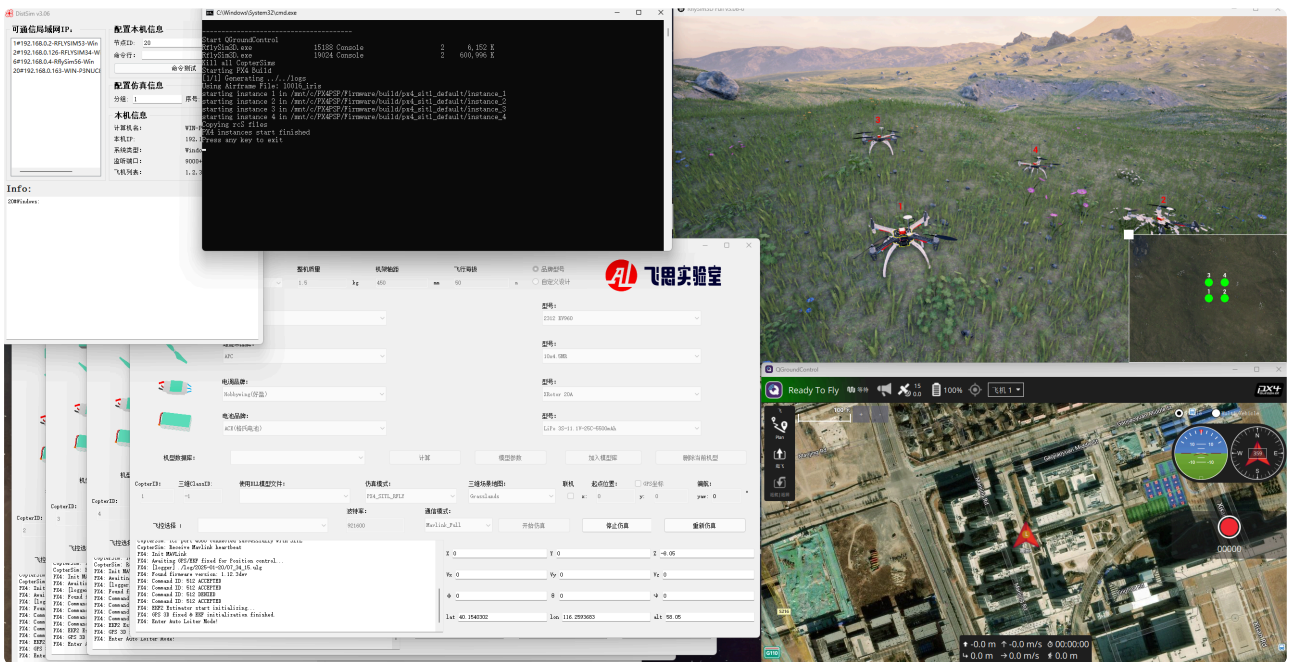


图 18 RflySim平台启动完毕

RflySim初始化完毕后，在Visual Studio Code回车执行下一步操作。

RflySim初始化完毕的标志是CopterSim打印出“PX4: GPS 3D fixed & EKF initialization finished.PX4: Enter Auto Loiter Mode!”语句。

```
C:\PX4PSP\Python38\python.exe
7, -0.0070323217660188675) (-0.3099934458732605, 0.16599492728710175, 0.3276301324367523) (-13.371360778308594, 8.560066
223144531, -7.177097797393799) (-13.382464379326224, 10.545283256837585, -15.007097488294132)
领航者状态信息: 3 [-0.2099379599094391, 0.21149024367332458, 0.2546854615211487] [-5.537547588348389, 1.558897852897644
-0.04378620907664299] [-0.2099379599094391, 0.21149024367332458, 0.2546854615211487] [-9.89383602142334, 11.2519197463
98926, -6.7333455085754395] [-7.917348749613094, 11.251919746199944, -14.976345200181612]
跟随者状态信息: 2 (-0.2532132863998413, 0.02765318937599659, 0.35499778389930725) (-3.661900758743286, -4.2716732025146
48, 0.02091354690492153) (-0.2532132863998413, 0.02765318937599659, 0.35499778389930725) (-17.880762100219727, 5.3528380
9398193, -7.150320053100586) (-17.891865700737355, 7.3380550730912475, -14.98031974400092)
领航者状态信息: 3 [-0.341341108083725, 0.15641936659812927, 0.2734486162662506] [-5.174440383911133, -1.541450733906860
4, -0.015013670548796654] [-0.341341108083725, 0.15641936659812927, 0.2734486162662506] [-15.321464538574219, 11.2683429
71801758, -6.761824131011963] [-13.344977266763973, 11.268342971602776, -15.004823822618135]
跟随者状态信息: 2 (-0.23358100652694702, -0.1656234860420227, 0.3774358332157135) (-1.5023468732833862, -5.254020690917
969, -0.022010399028658867) (-0.23358100652694702, -0.1656234860420227, 0.3774358332157135) (-20.579843521118164, 0.4921
9366908073425, -7.151285648345947) (-20.590947121635793, 2.4774107027737884, -14.98128533924628)
领航者状态信息: 3 [-0.28454601764678955, 0.02864721417427063, 0.2996863126754761] [-3.850041627883911, -4.5057969093322
75, 0.030765429139137268] [-0.28454601764678955, 0.02864721417427063, 0.2996863126754761] [-20.0446834564209, 7.97683763
5040283, -6.732192516326904] [-18.068196184610652, 7.9768376348413, -14.975192207933077]
跟随者状态信息: 2 (-0.18311567604541779, -0.29817062616348267, 0.3830360472202301) (1.397558569908142, -5.0767717361450
3818, -0.02033219486474991) [-0.18311567604541779, -0.29817062616348267, 0.3830360472202301] (-20.6485652923584, -4.787877
082824707, -7.160083770751953) (-20.659668892876027, -2.802660049131653, -14.990083461652286)
领航者状态信息: 3 [-0.2272370159626007, -0.16979822516441345, 0.32453545928001404] [-1.7774735689163208, -5.55880117416
3818, -0.02033219486474991] [-0.2272370159626007, -0.16979822516441345, 0.32453545928001404] [-22.998382568359375, 2.834
9913555145264, -6.727630138397217] [-21.02189529654913, 2.8343913553155433, -14.97062983000339]
跟随者状态信息: 2 (-0.05482125282287598, -0.2657754421234131, 0.3745996057987213) (4.057488441467285, -3.90899610519409
957, -0.024632876738905907) [-0.05482125282287598, -0.2657754421234131, 0.3745996057987213] (-17.745065689086914, -9.435042
881286621, -7.138570308685303) (-17.756169289604543, -7.449825347593567, -14.968569999585636)
领航者状态信息: 3 [-0.1851460486650467, -0.32885631918907166, 0.33360034227371216] [1.1506881713867188, -5.322850227355
957, -0.024632876738905907] [-0.1851460486650467, -0.32885631918907166, 0.33360034227371216] [-23.353008270263672, -2.58
67080688476562, -6.74900484085083] [-21.376520998453426, -2.5867080690466393, -14.992004532457003]
```

图 19 主机①执行示例

5.1.4 观察状态- RflySim3D

观察RflySim3D查看各个旋翼机飞行状态。

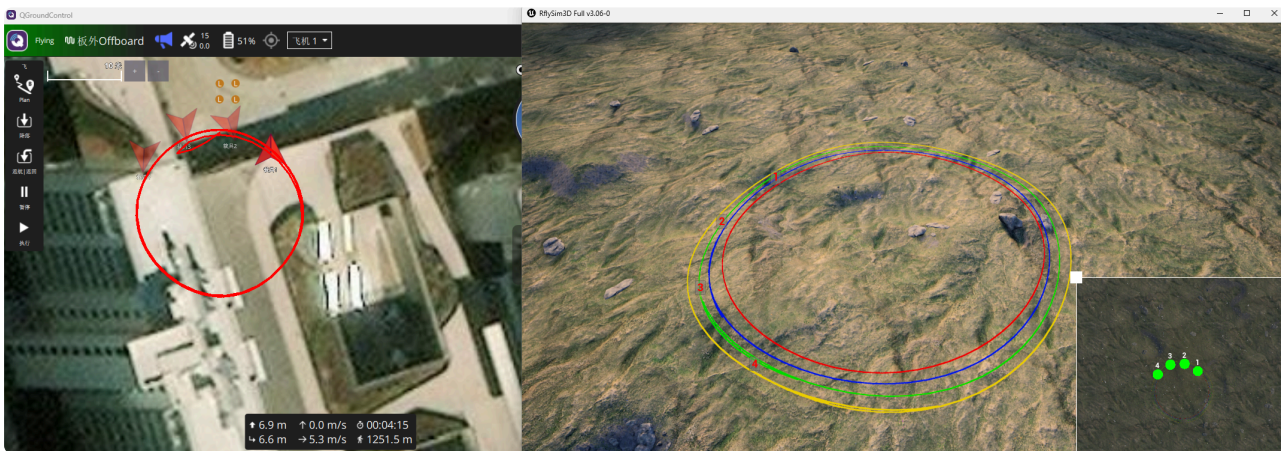


图 20 RflySim3D展示效果

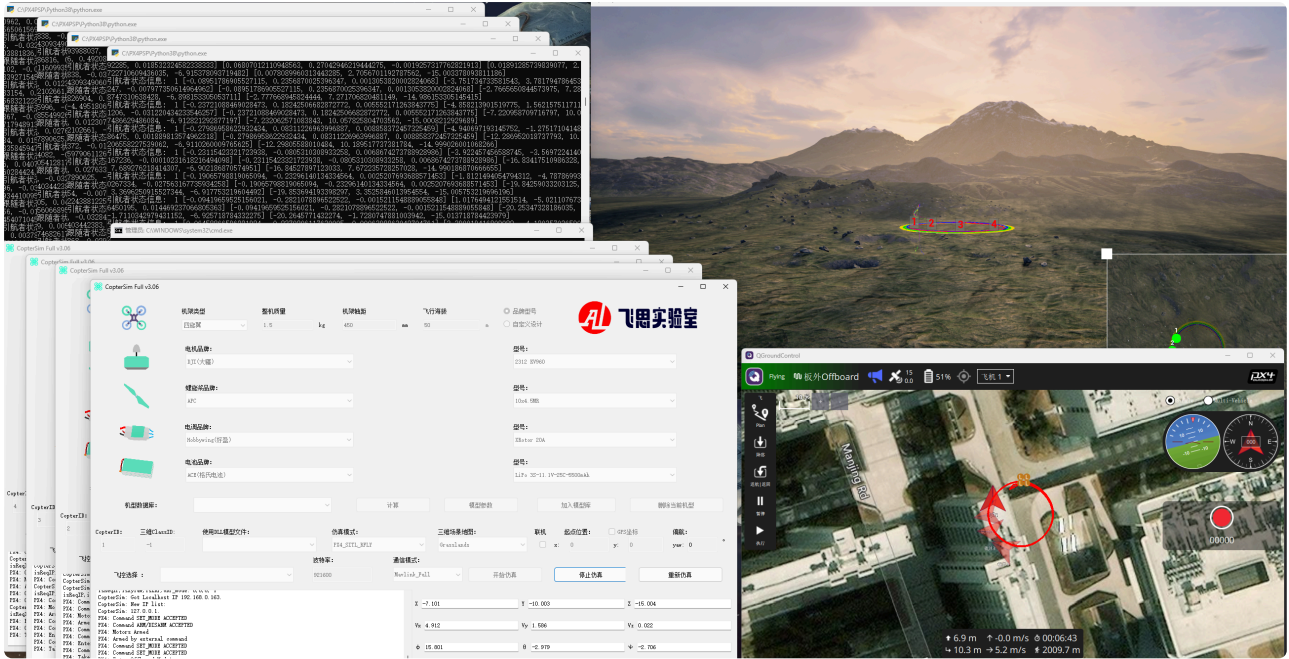


图 21 RflySim工具链实验阶段

6.参考资料

1. 无。

7.常见问题

Q1: ***

A1: ***