

1. 实验名称及目的

1.1 实验名称

Python Mavsdk控制实验

1.2 实验目的

展示用Python mavros进行飞机控制。

1.3 关键知识点

本实验主要是实现通过Python接口ReqCopterSim.py使用（见RflySimAPIs\RflySimSDK\ctrl目录）自动获取的IP，去建立远端电脑与本机RflySim3D（发送图片）与CopterSim（接收控制指令）的联机仿真。展示用Pythonmavsdk进行飞机控制。关键代码解析如下：

本例子和其他分布式例子的区别，主要在于使用mavsdk进行飞机控制。

关键知识点1：通过ReqCopterSim可以自动从局域网获取到仿真电脑的IP地址，从而自动建立连接，不再需要手动指定IP地址。不过，此种连接方式，可能在局域网中产生干扰（多台电脑同时打开多个CopterSim会产生误识别），不适合多个实验同时进行的场景。

1) ReqCopterSim接口使用（自动获取ip接口）

```
1 req = ReqCopterSim.ReqCopterSim() \# 获取局域网内所有CopterSim程序的电脑IP列表
2 TargetIP = req.getSimIpID(CopterID) \#自动获取CopterSim的CopterID号程序所在电脑的IP，作为目
3 req.sendReSimUdpMode(CopterID,UDP_mode=2) \#请求CopterSim将UDP模式转换为MAVLink_Full，便
4 req.sendReSimIP(CopterID) \# 请求mavlink数据到本电脑
```

2) 其余代码说明

```
1 | drone = System() \# 创建一个mavsdk实例
2 |
3 | await drone.connect(system_address="udp://:"+str(TargetPort)) \#等待连接上飞控, 这里指定在
4 |
5 | await drone.action.arm() \# 解锁飞控
6 |
7 | await drone.offboard.set_position_ned(PositionNedYaw(0.0, 0.0, 0.0, 0.0)) \#设置目标点
8 |
9 | await drone.offboard.start() \# 启动离板模式
10 |
11 | await drone.offboard.stop() \# 停止离板模式
```

2.实验效果



3.文件目录

例程目录:

[安装目录]\RflySimAPIs\8.RflySimVision\0.ApiExps\2-DistributedSimAPI\2.UavDistCtrl
\2.MavsdkApiTest

文件夹/文件名称	说明
MavsdkApiTest.bat	软件在环仿真一键脚本（MAVLinkFull模式）
offboard_position_ned.py	取图控制程序（支持Windows或Linux分布式运行）
Python38Run.bat	Windows下Python程序运行脚本
WinWSL.bat	WSL1/Ubuntu 20.04环境程序运行脚本
WslGUI.bat	WSL1/Ubuntu 20.04可视化界面脚本

4. 运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；Visual Studio Code；Linux（Ubuntu 20.04）；Linux（Ubuntu 20.04）。

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；WinWSL 1台；虚拟机/视觉盒子/其他板卡 可选台。

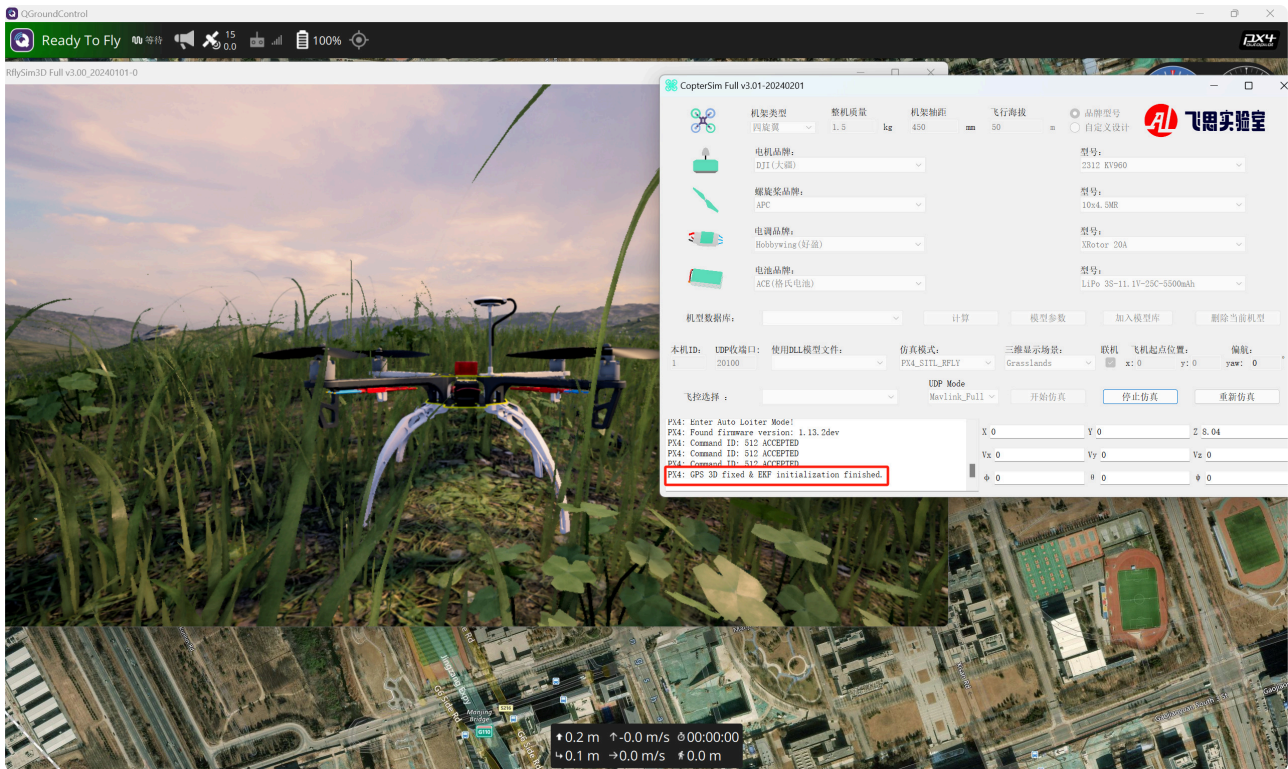
①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

5.1 必做实验：WinWSL控制

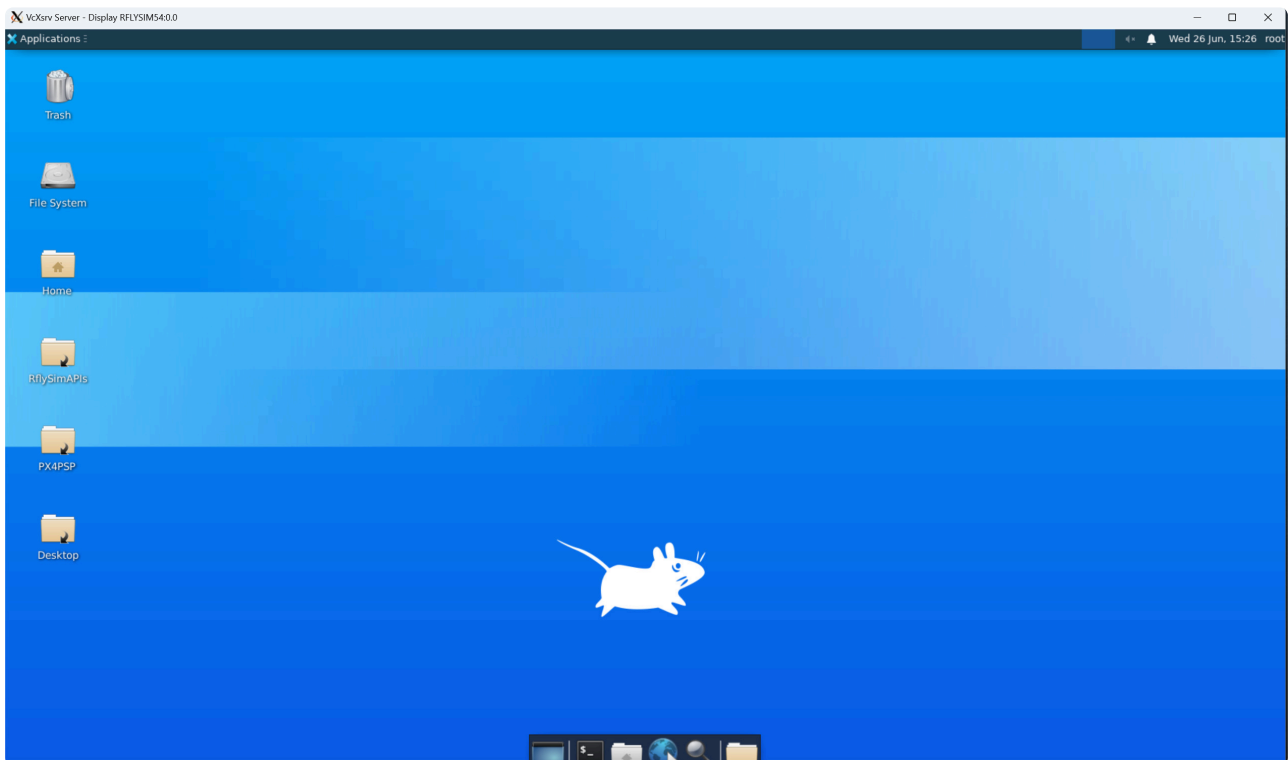
Step 1: 开启仿真

在Windows主机中，双击运行[MavsdkApiTest.bat](#)启动仿真界面。等待CopterSim界面中返回如下语句即可。



Step 2: 开启WSL可视化界面

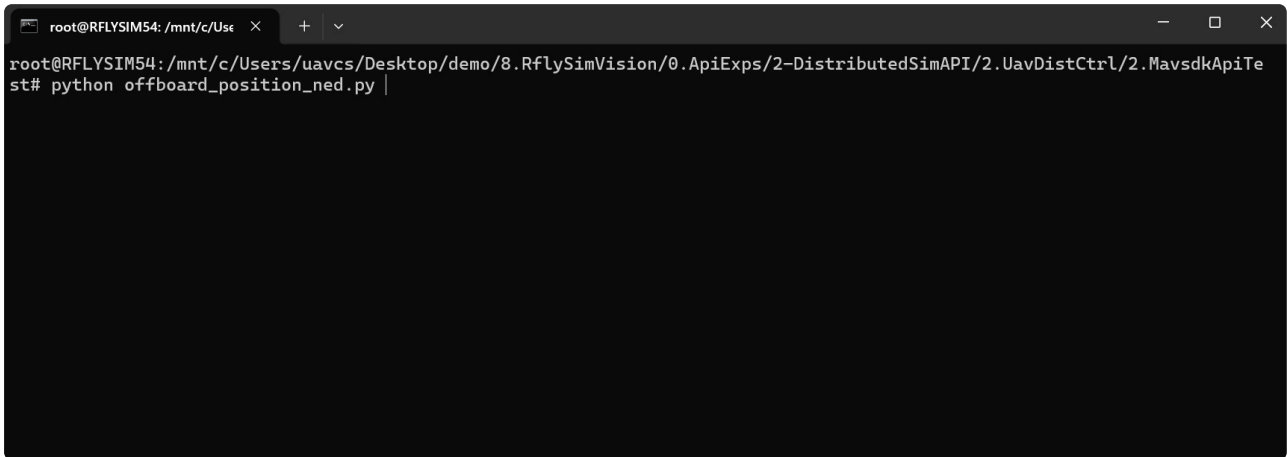
双击打开 [WslGUI.bat](#)，启动WSL可视化界面。（注：如果打开发现窗口白屏，没有桌面，则关了重开一两次。）



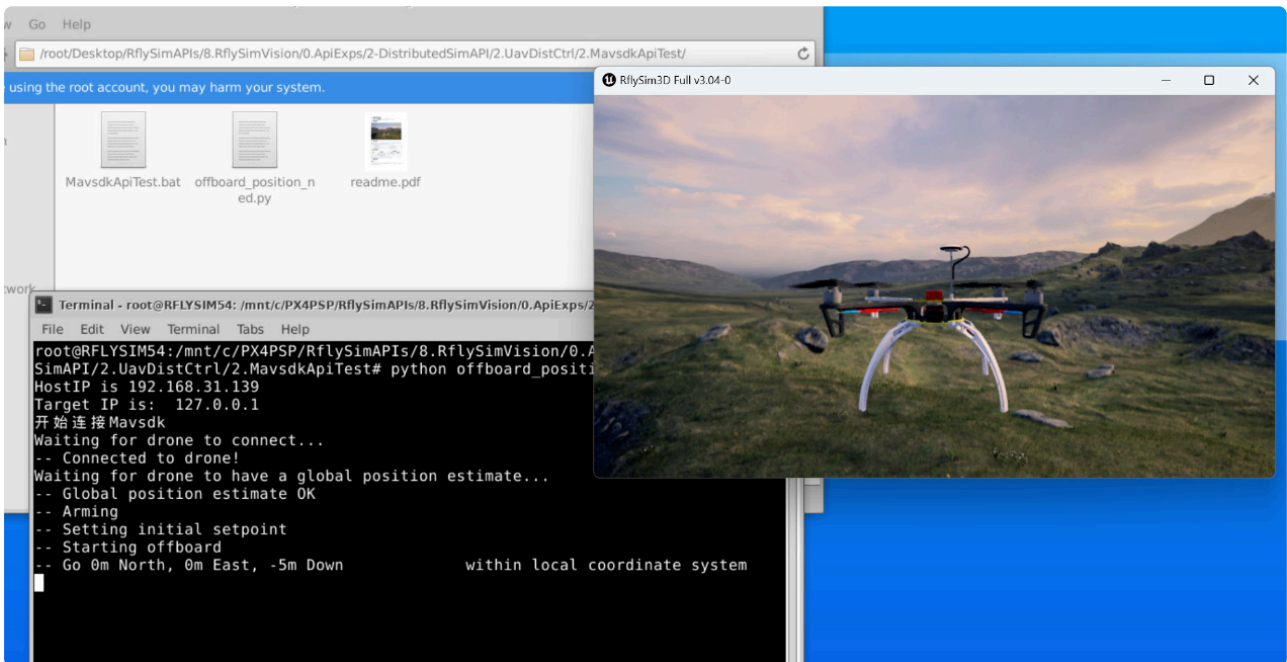
注意：参考 [\[安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e7_WslGUI\Intro.pdf](#)，了解WslGUI的功能与使用。

Step 3: 运行控制程序

双击打开 [WinWSL.bat](#)，运行命令 `python3 offboard_position_ned.py`，在WslGUI界面中可以看到生成如图所示三个窗口。



```
root@RFLYSIM54: /mnt/c/Users/...
root@RFLYSIM54: /mnt/c/Users/uavcs/Desktop/demo/8.RflySimVision/0.ApiExps/2-DistributedSimAPI/2.UavDistCtrl/2.MavsdApiTest# python offboard_position_ned.py
```



备注：可以参考

[\[安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e8.WslVsCode\Intro.pdf](#) 来使用VS Code开发并调试Ubuntu下python文件。

```
server_ue4.py X
> Users > uavcs > Desktop > demo > 8.RflySimVision > 0.ApiExps > 0.Preparation > 5.ManModifyIPRun > server_ue4.py > ...
1
2 # import required libraries
3 # pip3 install pymavlink pyserial
4
5 import cv2
6 import numpy as np
7 import time
8 import VisionCaptureApi
9 import PX4MavCtrlV4 as PX4MavCtrl
10 import math
11
12 StartCopterID = 1 # 初始飞机的ID号
13 TargetIP = "192.168.31.141"# 手动修改为电脑主机的IP
14 # 注意:如果是本电脑运行的话,那TargetIP是127.0.0.1的本地地址;如果是远程访问,则是192打头的局域网地址。
15 # 因此本程序能同时在本机运行,也能在其他电脑运行。
16 vis = VisionCaptureApi.VisionCaptureApi(TargetIP)
17
18 # VisionCaptureApi 中的配置函数
19 vis.jsonLoad()
20 isSuss = vis.sendReqToUE4(
21     0, TargetIP
22 )
23 vis.startImgCap() # 开启取图循环,执行本语句之后,已经可以通过vis.Img[i]读取到图片了
24 print('Start Image Receiver')
25 #vis.sendImuReqCopterSim(StartCopterID, TargetIP) # 发送请求,从目标飞机CopterSim读取IMU数据,回传地址为127.0.0.1,默认频率为200Hz
26 # 执行本语句之后,会自动开启数据监听,已经可以通过vis.imu读取到IMU数据了。
27
28 VehilceNum = 1
29 MavList=[]
30 # Create MAV instance
31 for i in range(VehilceNum):
32     CopterID=StartCopterID+i # 当前配置的飞机序号
33
34     time.sleep(1)
35     MavList = MavList+[PX4MavCtrl.PX4MavCtrl(CopterID,TargetIP)] # 初始化并建立i号飞机的MAVLink通信连接
36
```

Run Python File
Run Python File in Dedicated Terminal
Python 调试程序: 调试 Python 文件
Python 调试程序: 使用 launch.json 进行调试

5.2. 选作实验

准备工作:

虚拟机或NX的配置方法是相同的。

1) Ubuntu虚拟机环境下,进行分布式联机实验。先参考[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\0.Preparation\1.VMwareUbuntu\Readme.pdf](#)

,完成虚拟机的下载与配置。

2) 用第二台Ubuntu电脑或NX板卡,实现联机实验。其他Ubuntu电脑的配置,先看

[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\0.Preparation\2.GenenalUbuntuConfig\Readme.pdf](#)

; NX板卡的配置方法,先看

[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\0.Preparation\3.NXwithPX4Config\Readme.pdf](#)

o

扩展实验:

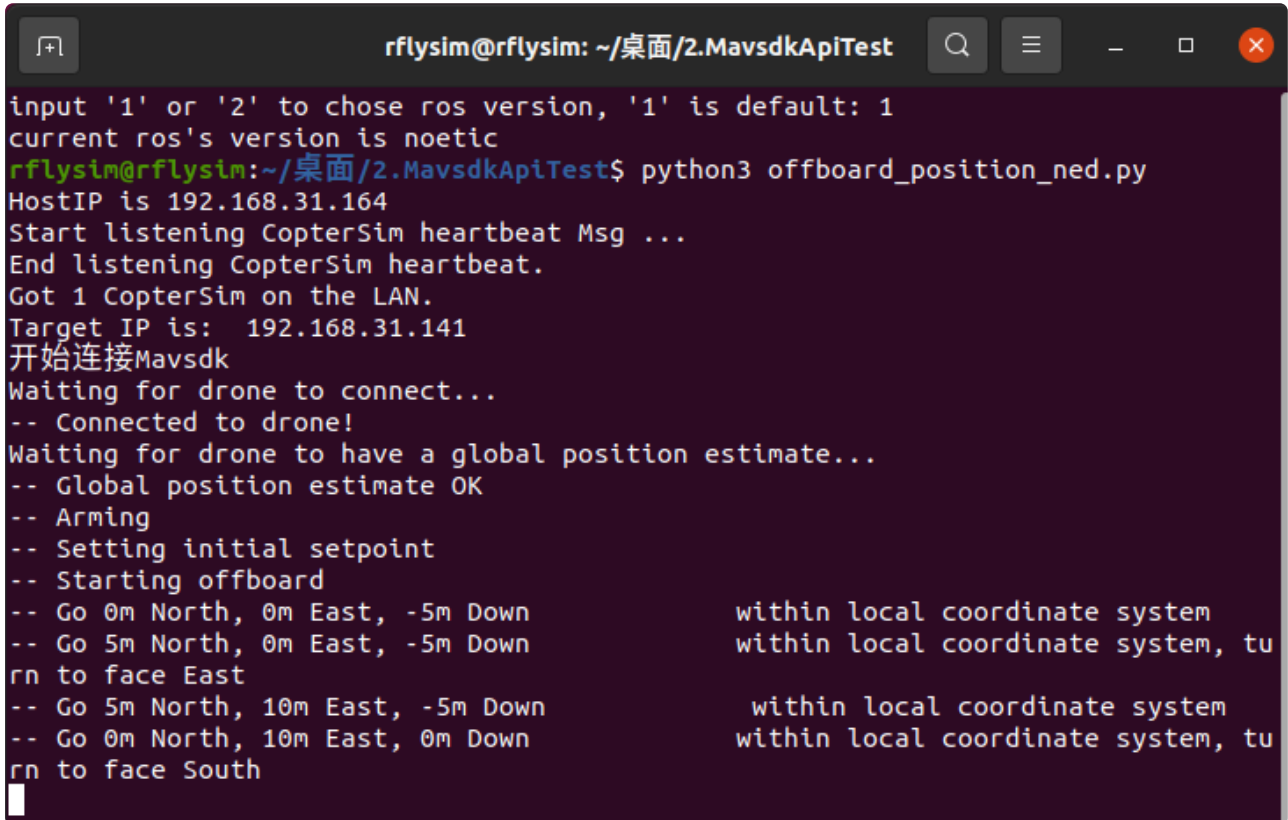
5.2.1在虚拟机/视觉板卡/另一台Ubuntu上接收图像实验

Step 1: 开启仿真

步骤1同上面的Step1步骤。

Step 2: 运行控制程序

将整个文件夹拷贝到Linux下，在虚拟机用命令 `python3 offboard_position_ned.py` 启动 `offboard_position_ned.py` 文件后，可以看到飞机在被控制。



```
rflsim@rflsim: ~/桌面/2.MavsdkApiTest
input '1' or '2' to chose ros version, '1' is default: 1
current ros's version is noetic
rflsim@rflsim:~/桌面/2.MavsdkApiTest$ python3 offboard_position_ned.py
HostIP is 192.168.31.164
Start listening CopterSim heartbeat Msg ...
End listening CopterSim heartbeat.
Got 1 CopterSim on the LAN.
Target IP is: 192.168.31.141
开始连接Mavsdk
Waiting for drone to connect...
-- Connected to drone!
Waiting for drone to have a global position estimate...
-- Global position estimate OK
-- Arming
-- Setting initial setpoint
-- Starting offboard
-- Go 0m North, 0m East, -5m Down          within local coordinate system
-- Go 5m North, 0m East, -5m Down        within local coordinate system, tu
rn to face East
-- Go 5m North, 10m East, -5m Down      within local coordinate system
-- Go 0m North, 10m East, 0m Down       within local coordinate system, tu
rn to face South
```



6.参考资料

无

7.常见问题

Q1: 无

A1: 无