

1. 实验名称及目的

1.1. 实验名称

激光雷达SLAM实验

1.2. 实验目的

在进行仿真时，获取载具运动数据以便后续处理。

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）并进行SLAM控制的联机仿真。关键代码解析如下：

本例子和其他分布式例子的区别，主要在于通过遍历各个目标点坐标进行对偏角、速度的实时控制，达到相应的路径规划控制。单线束激光雷达实时建图与定位。

关键知识点1：SendProtocol[0]决定了图像的传出模式。SendProtocol[0]=0：共享内存（仅限Windows下获取图像），1：UDP直传png压缩，2：UDP直传图片不压缩（只适用图片类传感器），3：UDP直传jpg压缩（只适用图片类传感器）。如果是激光雷达数据只有0或1（共享内存和UDP网络传输）。

关键知识点2：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) 视觉接口使用

```
1 vis.jsonLoad() \# 加载Config.json中的传感器配置文件
2
3 isSuss = vis.sendReqToUE4(0, TargetIP) \#向RflySim3D发送取图请求, 发给ip为TargetIP的地址
4
5 vis.startImgCap() \# 开启取图
6
7 vis.sendImuReqCopterSim(StartCopterID, TargetIP) \# 发送id号为StartCopterID的imu数据到ip
```

2) 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
3 TargetIP = req.getSimIpID(StartCopterID) \#自动获取CopterSim的StartCopterID号程序所在电脑IP
4
5 vis = VisionCaptureApi.VisionCaptureApi(TargetIP) \#创建一个视觉传感器实例, 这个实例对应的ip就是TargetIP
6
7 req.sendReSimIP(CopterID) \# 请求mavlink数据到本电脑
```

3) 相机数量和参数配置

其中，视觉传感器的初始状态由本文件夹下的Config.json决定，主要包含以下配置项：

```
1 "SeqID":0 : 使用自动更新ID的方式, 创建了SeqID为0、1和2的三个视觉传感器
2
3 "TypeID":3 : 传感器类型为深度图像
4
5 "TargetCopter":1 : 相机绑定在1号飞机上
6
7 "SendProtocol":[1,0,0,0,0,0,0,0] : 传输模式为1 : UDP网络传输模式 (图片使用jpeg压缩, 点云直传)。
8
9 "SensorPosXYZ":[0.3, -0.025, 0]和"SensorPosXYZ":[0.3, 0.025, 0] : 两个相机一左一右分布。
```

4) 飞机控制指令

```
1 MavList = MavList+[PX4MavCtrl.PX4MavCtrl(CopterID,TargetIP)] \#初始化并建立i号飞机的MA
2
3 MavList[i].InitMavLoop() \# 初始化Mavlink监听程序, 读取第i个飞机数据
4
5 MavList[i].initOffboard() \# 第i个飞机进入Offboard模式
6
7 Error2UE4Map =Error2UE4Map+[-np.array([mav.uavGlobalPos[0]-mav.uavPosNED[0],mav.uavGL
8
9 MavList[i].SendPosNED(0, 0, -1, 0) \# 对第i个飞机发送1米高的位置控制指令
10
11 mav=MavList[j] \# 从列表中取第i个实例作为当下处理的飞机实例
12
13 MavList[i].endOffboard() \# 对第i个飞机进入endoffboard模式
14
15 MavList[i].stopRun() \# 终止第i个飞机
```

5) UE控制

接口详细使用方法见: [UE4CtrlAPI.py](#)

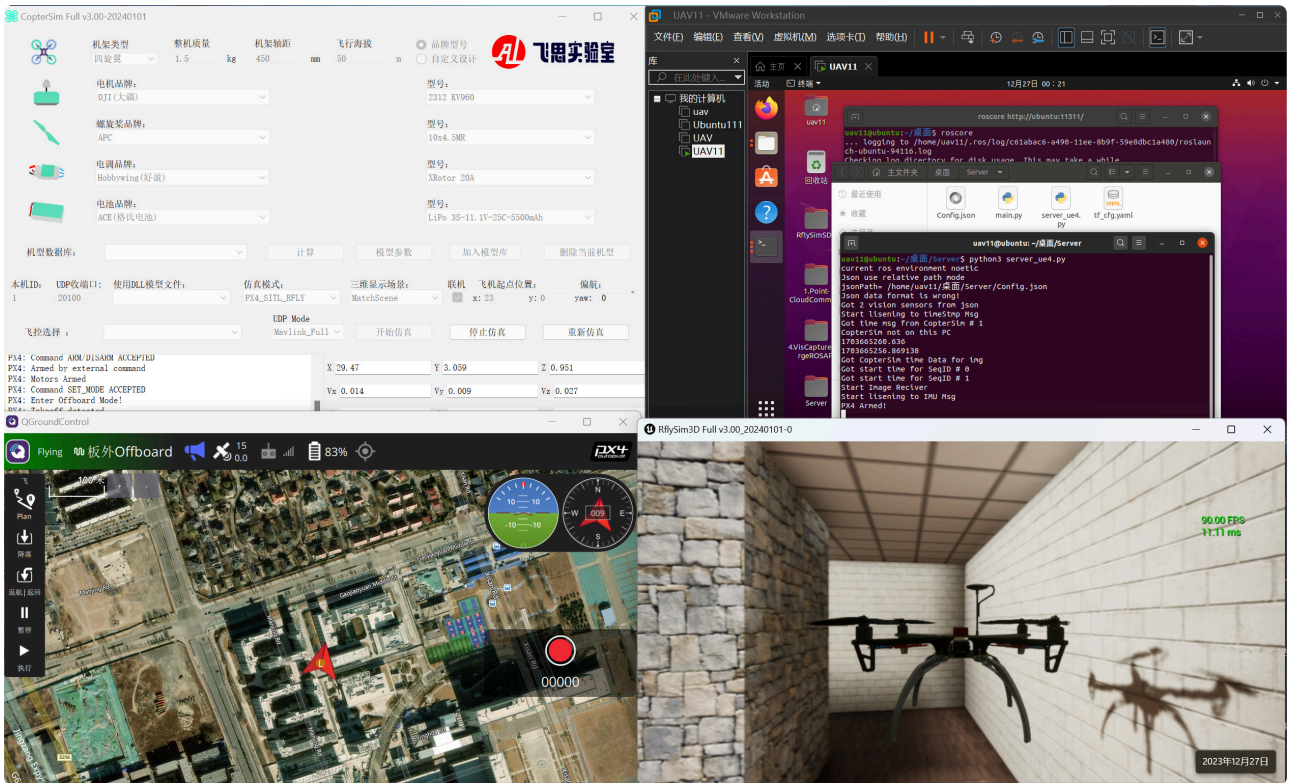
```
1 ue = UE4CtrlAPI.UE4CtrlAPI() \# 创建UE控制实例
2
3 ue.sendUE4Cmd('r.setres 1280x720w',0) \#发送指令, 设置UE4窗口分辨率, 注意本窗口仅限于显示, 取
4
5 ue.sendUE4Cmd('t.MaxFPS 30',0) \#发送指令, 设置UE4最大刷新频率30Hz, 同时也是取图频率
6
7 ue.sendUE4Cmd('stat fps',0) \# 设置RflySim3D-0显示当前帧率
```

6) 其余代码说明

```
1 RoomThreeAirPlan = nparry.tolist() \#nparry是一个NumPy数组。使用tolist()方法将其转换为一个F
2
3 RoomThreeAirPlan3D = np.insert(RoomThreeAirPlan, 2, -np.ones(len(RoomThreeAirPlan)), a
4
5 def direction(StartAngle, EndAngle) \# 计算旋转方向
6
7 def SearchPoint(speed, rate, PosE, AngEuler, mav, PointList) \#用于控制无人机按照预定的点位
8
9 class myThread(threading.Thread) \# 用于创建一个独立线程来运行 SearchPoint函数, 控制无人机路
```

2. 实验效果

本实验将看到无人机自动避障移动。



3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\8.RflySimVision\2.AdvExps\e6_LaserSLAMdemo](#)

文件夹/文件名称	说明
RflySimPlatform	实验运行文件夹
cartographer_ros.zip	激光 SLAM 开源算法压缩包
pointcloud_to_laserscan.zip	激光雷达点云数据转换成 laserScan 数据功能包
RflySim 激光SLAM的应用.pdf	RflySim 激光 SLAM 的应用说明文档
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。

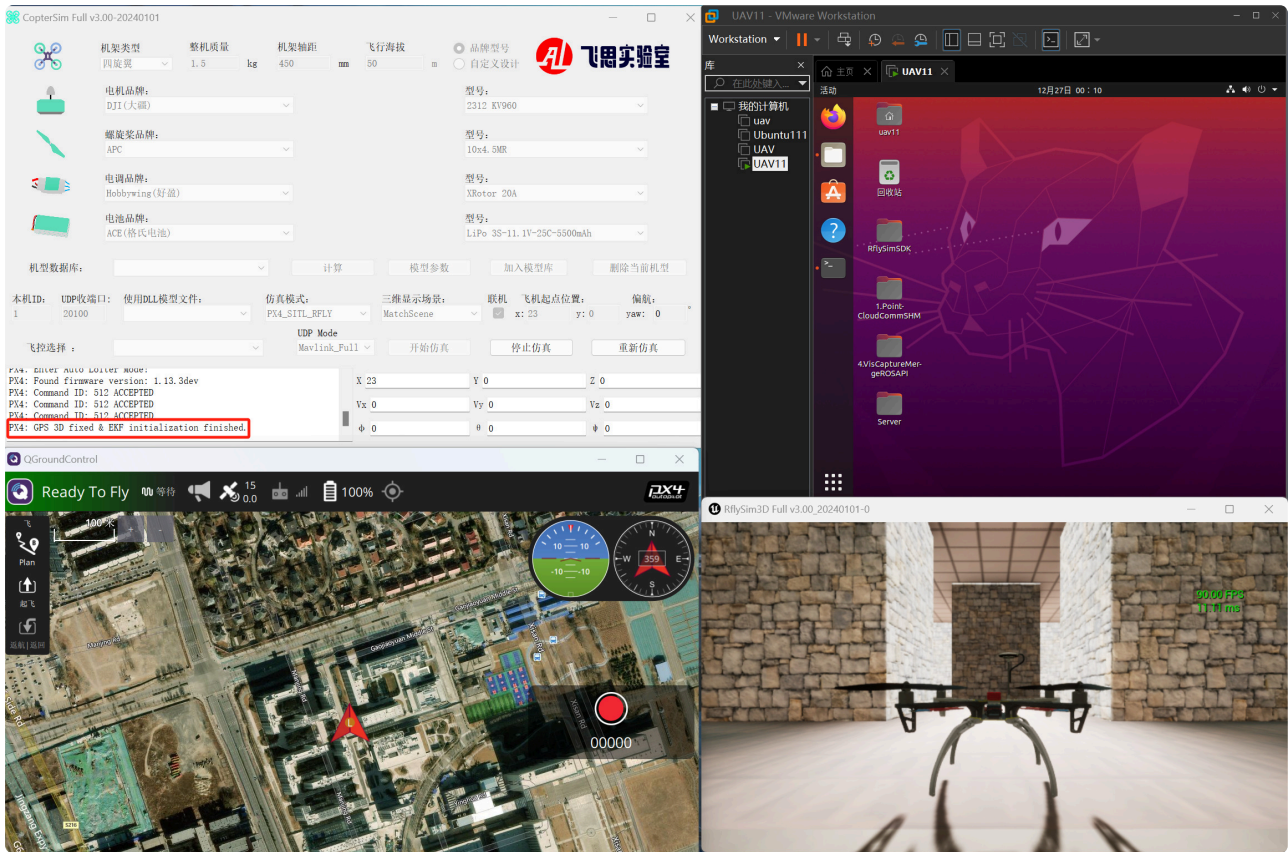
①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

5.1. 必做实验：WinsWSL控制

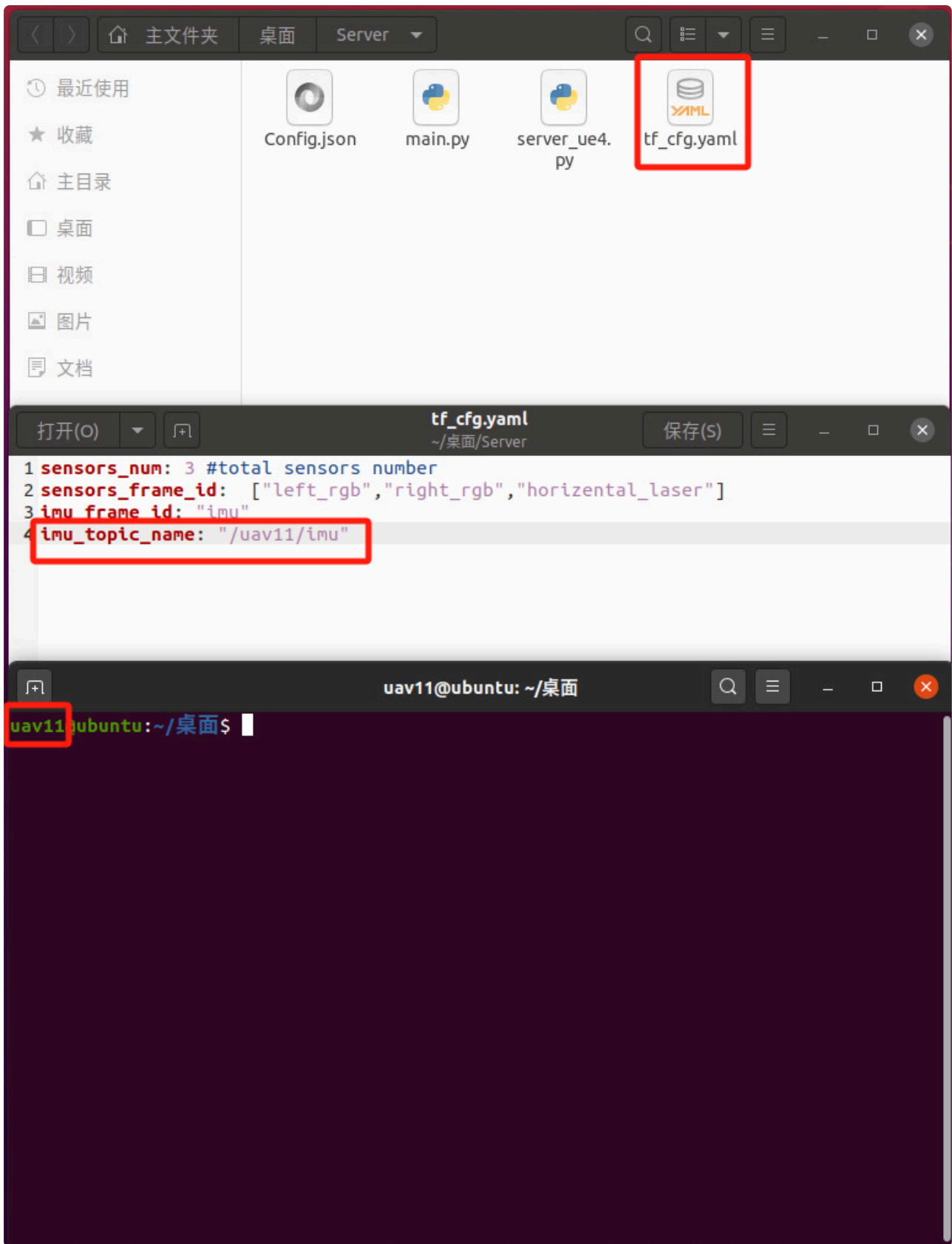
Step 1: 开启仿真

在 windows 下双击运行 Client 文件夹下的VisionCapAPIDemo.bat，将会启动1个QGC地面站，1个CopterSim软件且其软件下侧日志栏必须打印出GPS 3D fixed & EKF initialization finished字样代表初始化完成，并且有1个RflySim3D软件有1架无人机。并且启动一个已安装了ros的Ubuntu虚拟机。如下图所示：



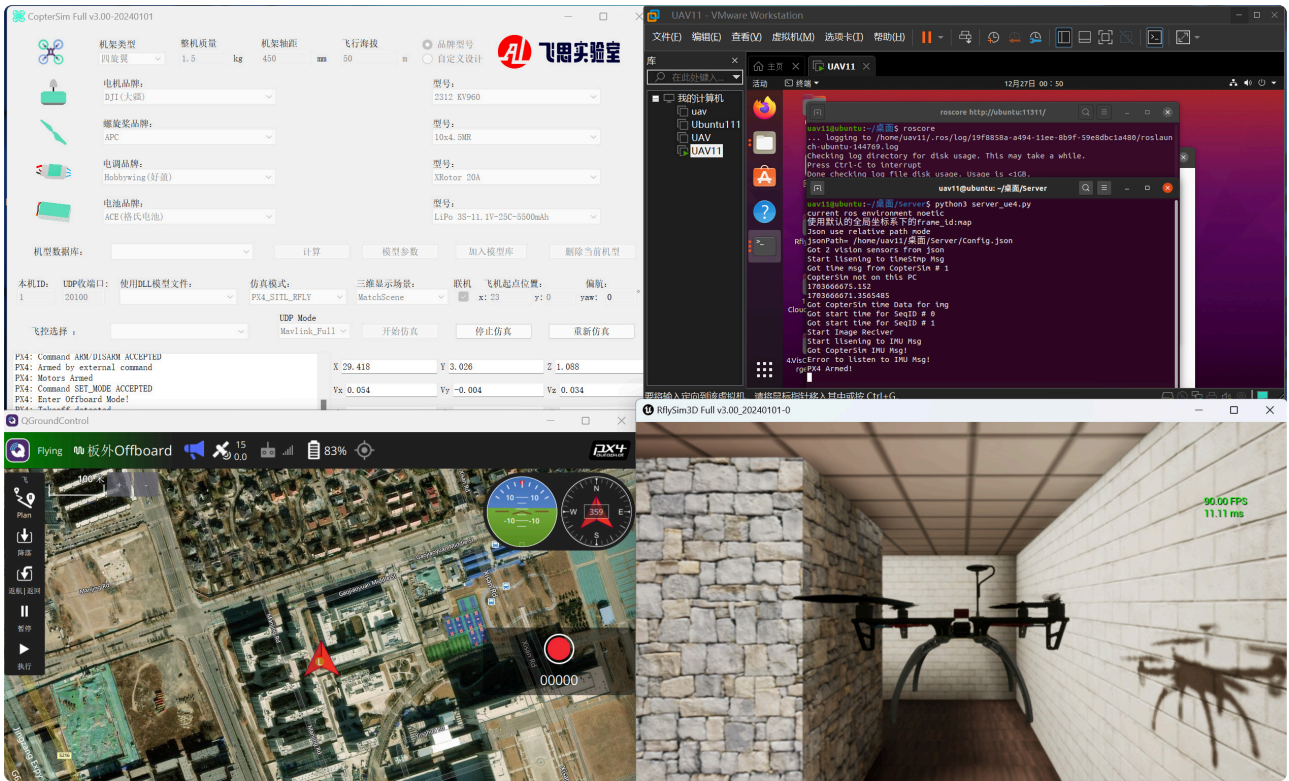
Step 2: 修改配置文件

进入RflySimPlatform\Server文件夹，打开tf_cfg.yaml，如下图所示增加imu_topic_name。



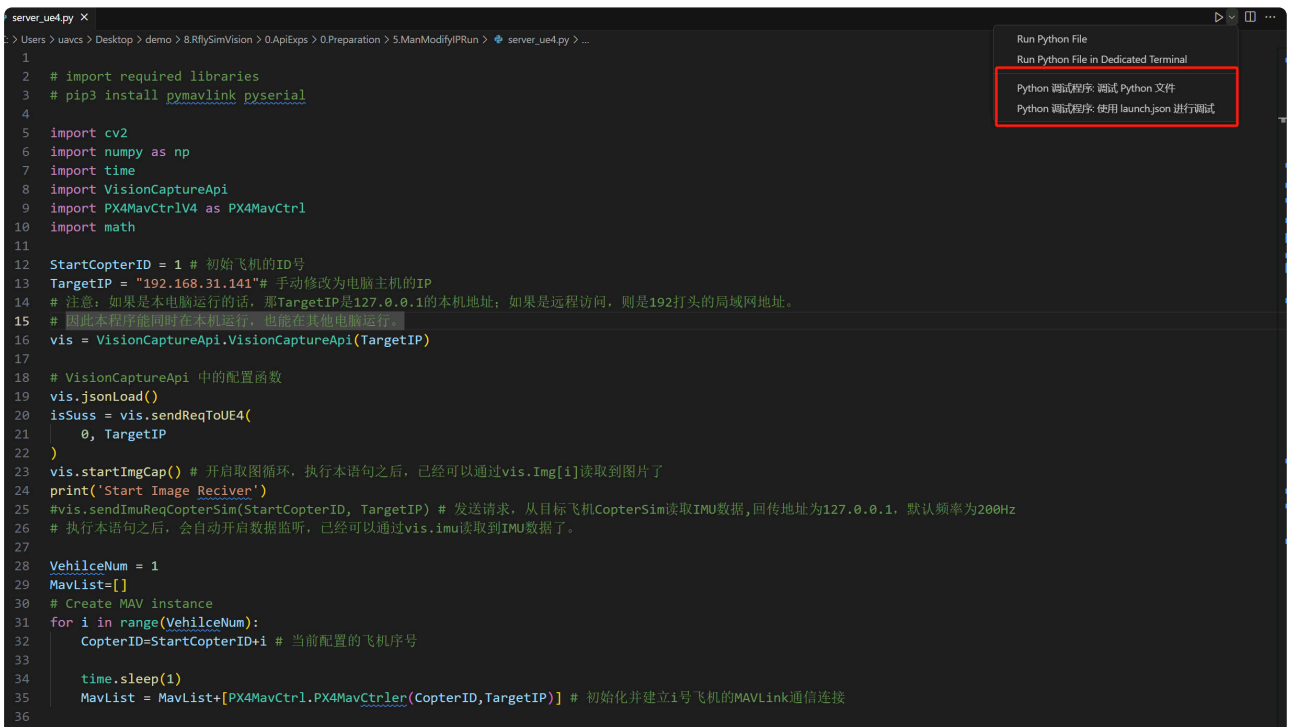
Step 3: 运行控制程序

进入 `\RflySimPlatform\Server` 文件夹双击打开 `WinWSL.bat`，打开一个终端输入 `roscore` 命令启动 `roscore`。双击打开 `WinWSL.bat`，运行命令 `python3 server_ue4.py`，将看到无人机自动避障移动。



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。



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.GeneralUbuntuConfig\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](#)

。

扩展实验：

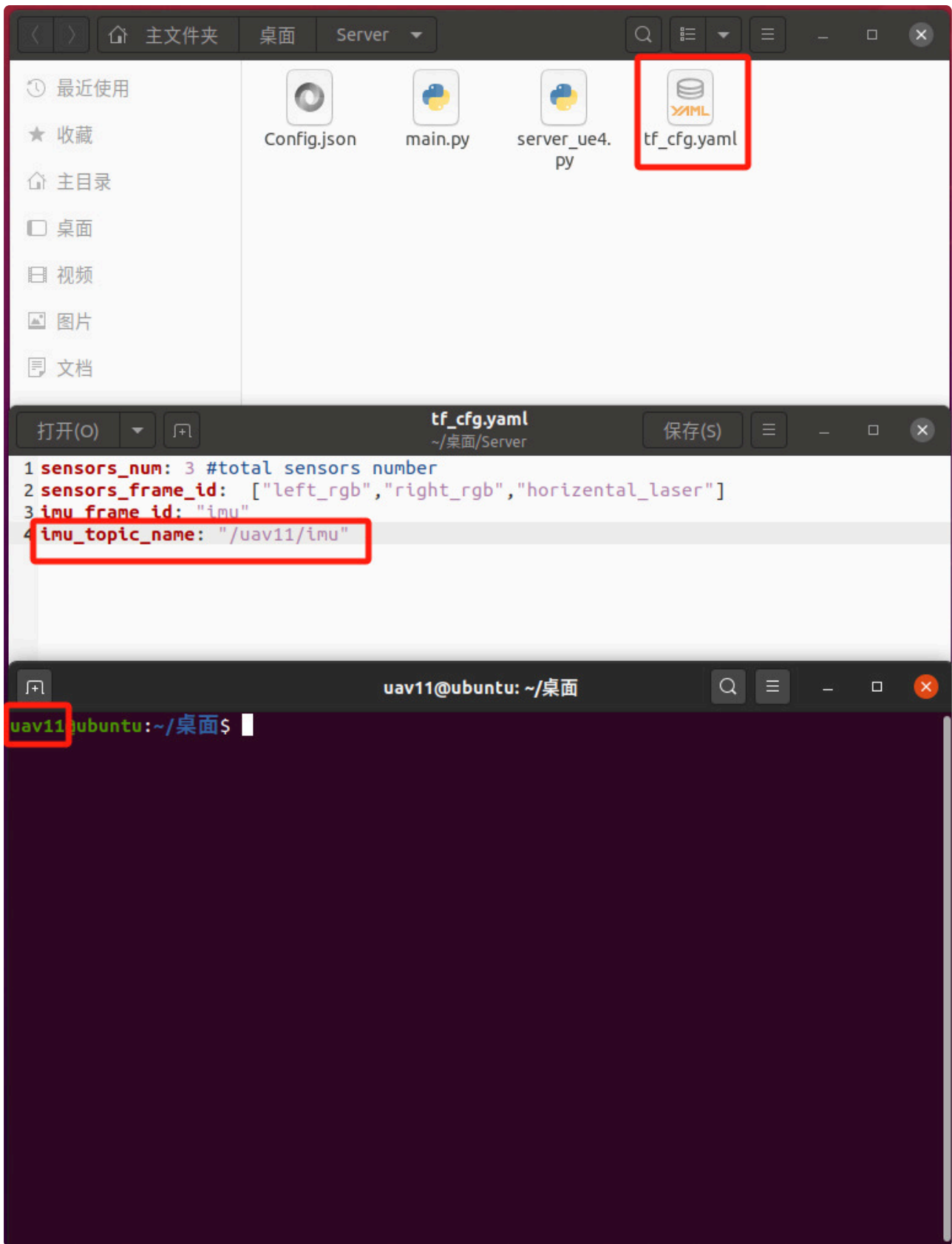
5.2.1在虚拟机/视觉板卡/另一台Ubuntu上接收图像实验

Step 1: 开启仿真

步骤1同6、实验步骤的Step1步骤。

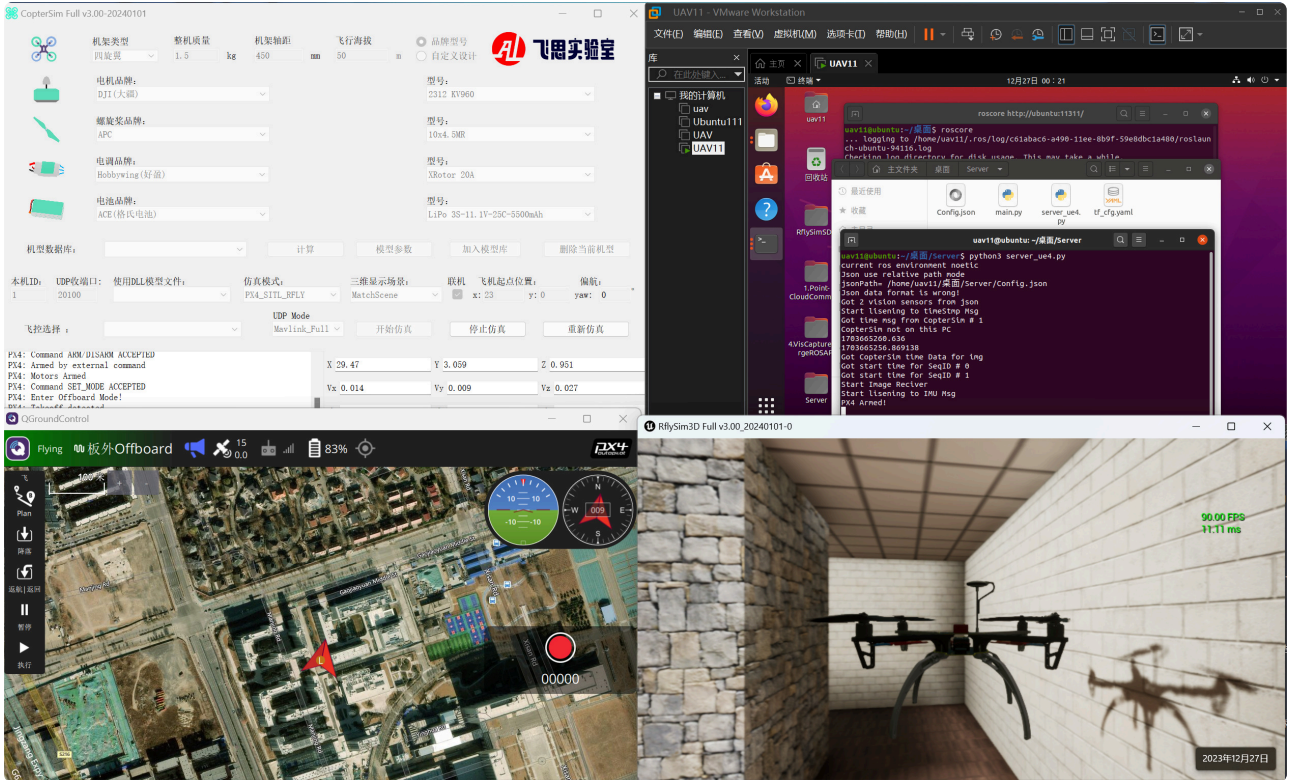
Step 2: 修改配置文件

拷贝 Server 文件夹到 ubuntu系统。打开tf_cfg.yaml，如下图所示增加 imu_topic_name。



Step 3: 运行控制程序

打开一个终端输入roscore命令启动roscore，重启一个在Server文件夹路径的终端下，运行命令 `python3 server_ue4.py`。将看到无人机自动避障移动。



6.参考资料

无

7.常见问题

Q1: 无

A1: 无