

1. 实验名称及目的

1.1 实验名称

三个位置跟踪控制器接口的仿真实验

1.2 实验目的

在视觉控制时，我们常常需要在控制飞机飞往指定目标位置的同时，控制飞机的前飞速度，达到好的跟踪效果。

注意：本实验只支持Windows下Python环境运行。不支持在WSL以及虚拟机下运行。

1.3 关键知识点

本实验主要是实现通过Python接口PX4MavCtrlV4.py（见RflySimAPIs\RflySimSDK\ctrl目录）通过ue.sendUE4Cmd创建物体，然后通过mav.SendPosNED控制机体飞行到指定位置。关键代码解析如下：

1) 视觉接口使用

```
1 vis = VisionCaptureApi.VisionCaptureApi() \# 创建一个视觉传感器实例
2
3 vis.jsonLoad() \# 加载Config.json中的传感器配置文件
4
5 isSuss = vis.sendReqToUE4() \# 向RflySim3D发送取图请求
6
7 vis.startImgCap() \# 开启取图
8
9 vis.hasData[i] \# 图片i数据是否更新
10
11 vis.Img[i] \# 图片i数据（像素矩阵）
12
13 cv2.imshow('Img'+str(i),vis.Img[i]) \# 显示图片i图像
```

2) 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和1的两个视觉传感器
2 |
3 | "TypeID":1:传感器类型为RGB彩色图像
4 |
5 | "TargetCopter":1:相机绑定在1号飞机上
6 |
7 | "SendProtocol":[0,0,0,0,0,0,0,0]:传输模式为0共享内存机制,因此本例程只能运行在Windows环境下。
8 |
9 | "SensorPosXYZ":[0.3,0,0]:相机的分布

```

o

3) 飞机控制指令

```

1 | mav = PX4MavCtrl.PX4MavCtrl(1) \# 创建飞机控制实例
2 |
3 | mav.InitMavLoop() \# 初始化Mavlink监听程序,读取飞机数据
4 |
5 | mav.initOffboard() \# 进入Offboard模式
6 |
7 | mav.SendMavArm(True) \# 解锁飞控
8 |
9 | mav.SendPosNED(0, 0, -10, 0) \# 发送10米高的位置控制指令
10 |
11 | mav.SendCopterSpeed(3) \#调用函数来设定飞机最大速度为3m/s

```

4) 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 720x405w',0) \#发送指令,设置UE4窗口分辨率,注意本窗口仅限于显示,取图
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.sendUE4Pos(100,152,0,[3,0,-10],[0,0,0]) \#创建一个物体,id号为100,类型为152,速度为0,在
8 |
9 | ###

```

5) 其余代码说明

```
1 | timeInterval = 1/30.0 \# 以30hz的频率进行控制
2 |
3 | lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
4 |
5 | sleepTime = lastTime - time.time() \#计算休息时间，从而保持按照设定的频率执行代码
```

2.实验效果

通过不同模式控制器控制飞机速度，并始终指向目标。

3.文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\1-UsageAPI\6.ThreeCtrlModes](#)

文件夹/文件名称	说明
ThreeCtrlModesSITL.bat	软件在环一键仿真启动脚本
ThreeCtrlModesHITL.py	硬件在环一键仿真启动脚本
ThreeCtrlModes_PosCtrl.py	Python实验代码
ThreeCtrlModes_PosCtrlFRD.py	Python实验代码
ThreeCtrlModes_VelCtrlBody.py	Python实验代码
ThreeCtrlModes_VelCtrlEarth.py	Python实验代码
Config.json	视觉传感器配置文件
Python38Run.bat	Python环境启动脚本

4. 运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；VS Code。

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://rflsim.com/doc/zh/1/Hardware.html>

4.2 硬件要求

笔记本/台式电脑① 1台；Pixhawk 6x 或 Pixhawk 6x mini② 1台。

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

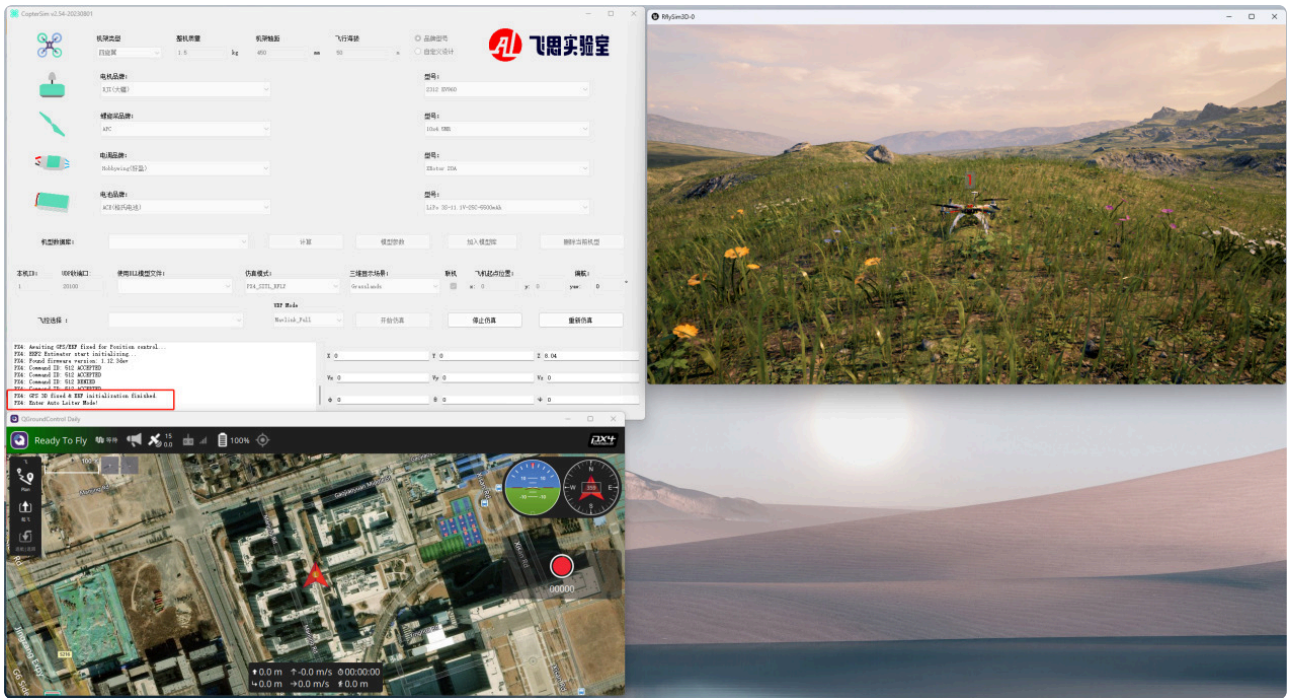
5. 实验步骤

5.1. 必做实验：Windows取图控制

软件在环实验步骤

Step 1: 开启仿真

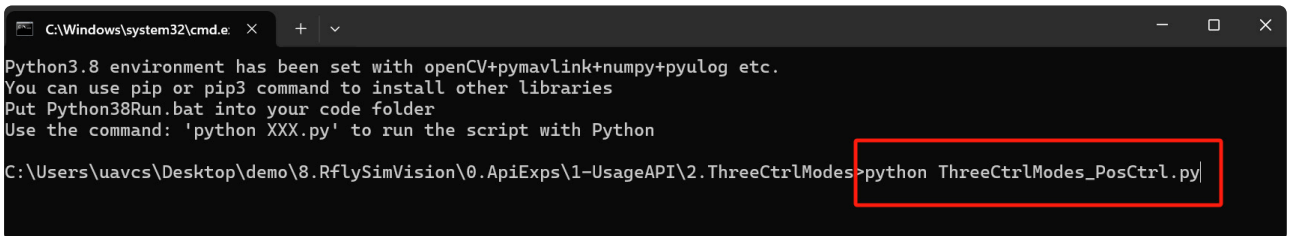
双击运行 `ThreeCtrlModesSITL.bat` 开启一个飞机的软件在环仿真。将会启动1个QGC地面站，1个CopterSim软件且其软件下侧日志栏必须打印出GPS 3D fixed & EKF initialization finished字样代表初始化完成，并且RflySim3D软件内有1架无人机。



Step 2: 运行控制程序

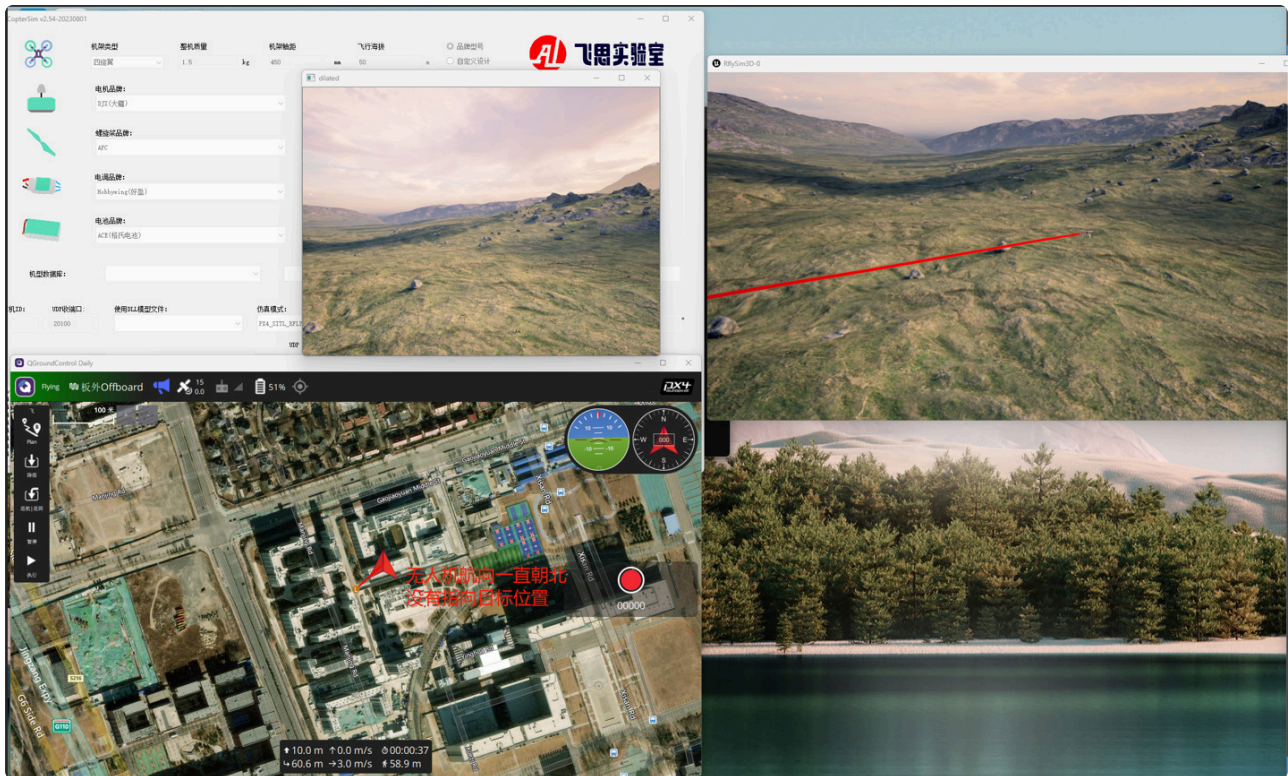
在文件夹下，双击 [Python38Run.bat](#)，打开集成好的python环境，在该环境下运行 [ThreeCtrlModes_PosCtrl.py](#) 文件，输入

```
python ThreeCtrlModes_PosCtrl.py
```



注：也可以使用VScode打开运行（如果安装好了VScode），执行如下：

用VScode打开到本实验路径文件夹，运行 [ThreeCtrlModes_PosCtrl.py](#) 文件，出现一个RGB相机窗口，并且可以看到飞机飞往指定位置[120,130,-10]，在QGC上可以看到速度设定值为3m/s，同时机头始终朝北和速度方向不一致。



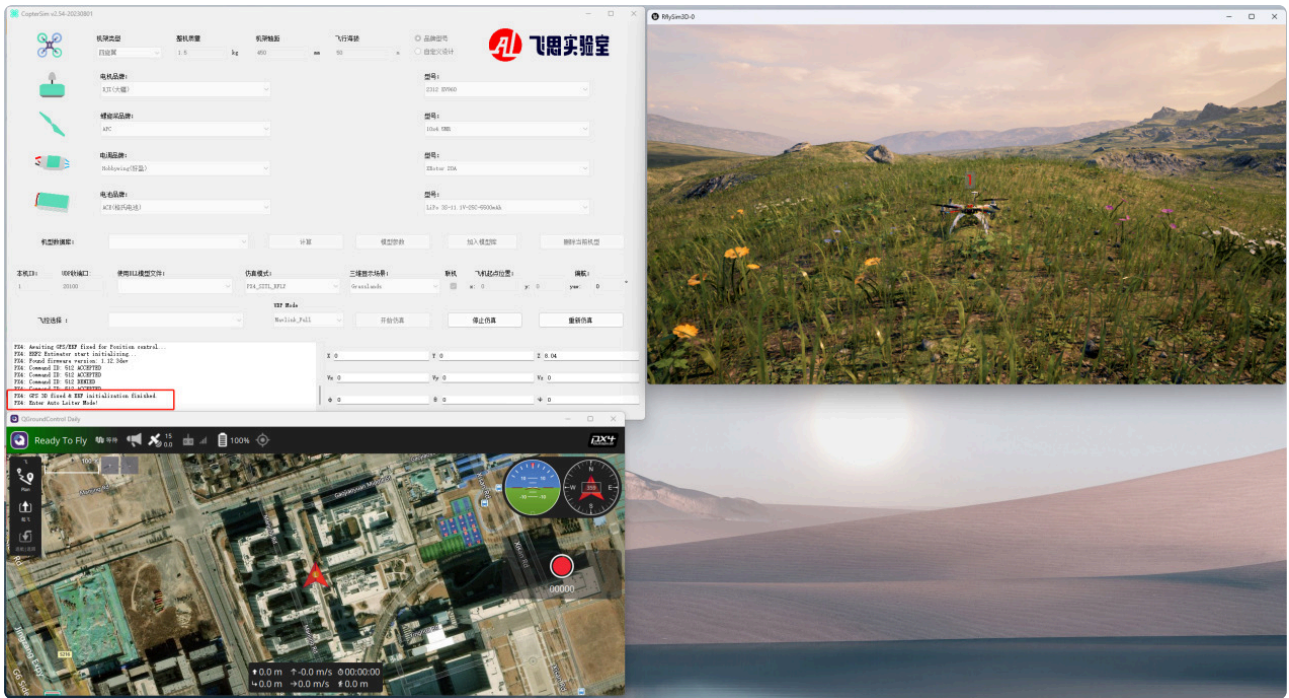
Step3: 观察结果

依次运行同目录下的 [ThreeCtrlModes_PosCtrlFRD.py](#)、[ThreeCtrlModes_VelCtrlBody.py](#)、[ThreeCtrlModes_VelCtrlEarth.py](#)，请自行阅读代码及注释，并确认实验效果。可以观察到该三个程序中分别设计了各自的控制器，对偏航角进行了控制，实时指向目标位置，相对于上步骤的程序对无人机速度、偏角控制得更精确。

硬件在环实验步骤

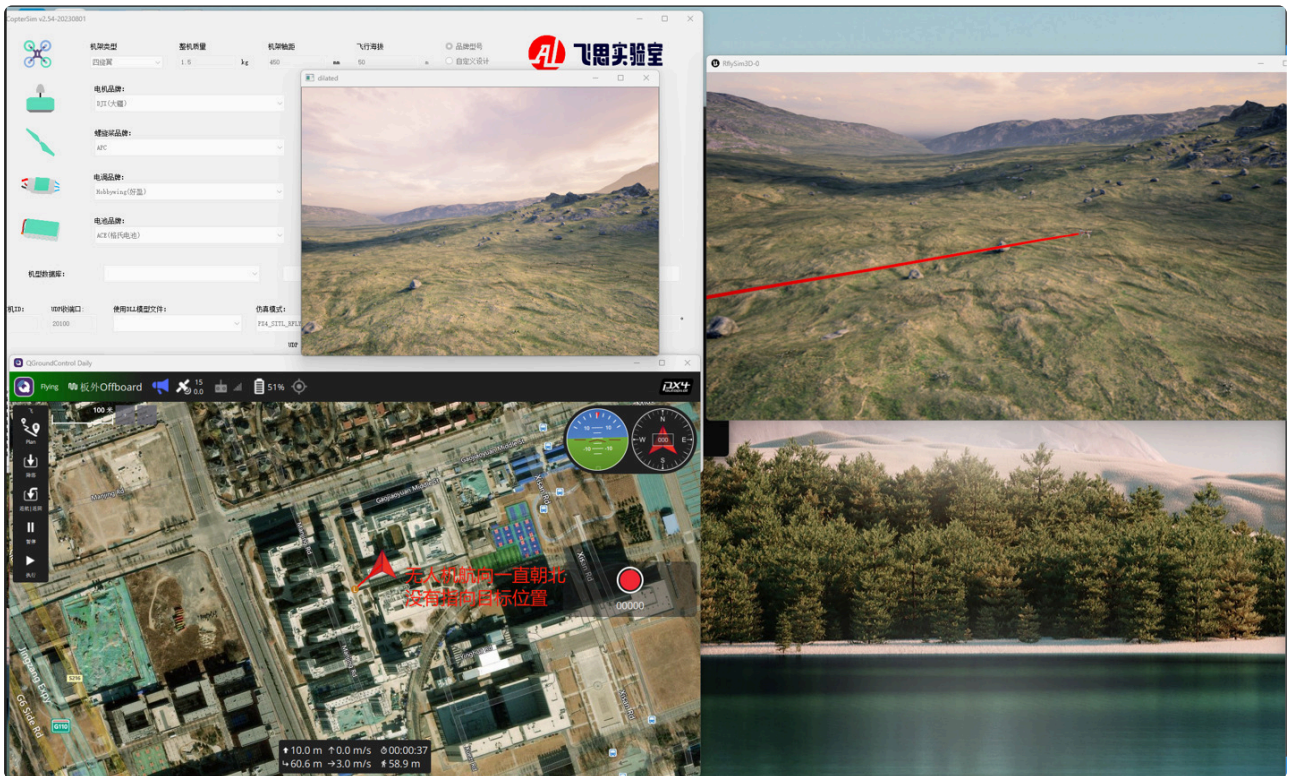
Step 1: 开启仿真

首先一定要确保飞控固件为平台标准固件，这是运行该硬件在环实验的前提条件，然后用USB数据线将Pixhawk6x飞控与电脑连接，以管理员方式运行 [ThreeCtrlModesHITL.bat](#)，输入端口号开启一个飞机的硬件在环仿真。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RflySim3D软件内有1架无人机。



Step 2: 运行控制程序

用VSCode打开到本实验路径文件夹，运行 `ThreeCtrlModes_PosCtrl.py` 文件，出现一个RGB相机窗口，并且可以看到飞机飞往指定位置[120,130,-10]，在QGC上可以看到速度设定值为3m/s，同时机头始终朝北和速度方向不一致。



Step3: 观察结果

依次运行同目录下的 [ThreeCtrlModes_PosCtrlFRD.py](#)、[ThreeCtrlModes_VelCtrlBody.py](#)、[ThreeCtrlModes_VelCtrlEarth.py](#)，请自行阅读代码及注释，并确认实验效果。可以观察到该三个程序中分别设计了各自的控制器，对偏航角进行了控制，实时指向目标位置，相对于上步骤的程序对无人机速度、偏角控制得更精确。

Step 4: 结束仿真

在下图“[ThreeCtrlModesSITL.bat](#)”脚本开启的命令提示符CMD窗口中，按下回车键（任意键）就能快速关闭CopterSim、QGC、RflySim3D等所有程序。

5.2. 选作实验（VS Code调试运行）

准备工作

- 先确保已经按 [RflySimAPIs\1.RflySimIntro\2.AdvExps\3.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。
- 其他步骤与上文相同，在Step2运行ThreeCtrlModesSITL.py时，可使用VS Code（或Pycharm等工具）来打开ThreeCtrlModesSITL.py文件，并阅读代码，修改代码，调试执行等。

扩展实验

- 请自行使用VS Code阅读ThreeCtrlModesSITL.py源码，通过程序跳转，了解每条代码的执行原理；再通过调试工具，验证每条指令的执行效果。

```
> RflySimAPIs > 8.RflySimVision > 0.ApiExps > 1-UsageAPI > 0.VisionSensorAPI > 1.CameraAPI
8   ue = UE4CtrlAPI.UE4CtrlAPI()
9
10  #Create a new MAVLink communication instance, UDP sending
11  mav = PX4MavCtrl.PX4MavCtrl(1)
12
13  # The IP should be specified by the other computer
14  vis = VisionCaptureApi.VisionCaptureApi()
15
16  # Send command to UE4 Window 1 to change resolution
17  ue.sendUE4Cmd('r.setres 1280x720w',0) # 设置UE4窗口分辨率, 同时也会设置UE4最大刷新频率, 同时也
18  ue.sendUE4Cmd('t.MaxFPS 30',0) # 设置UE4最大刷新频率, 同时也
19  time.sleep(2)
20
21  # VisionCaptureApi 中的配置函数
22  vis.jsonLoad() # 加载Config.json中的传感器配置文件
--
```

- 请尝试修改代码, 实现飞机位置改变、相机姿态角改变、相机参数改变等功能。

6.参考资料

无

7.常见问题

Q1: 无

A1: 无