
1、实验名称及目的

三个位置跟踪控制器接口的仿真实验：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。

2、实验原理

首先进行 Config.json 文件进行相机传感器的配置，配置参数如下解释：

“SeqID”代表第几个传感器。此处 0 表示第 1 个传感器（免费版只支持 2 个图）。

“TypeID”代表传感器类型 ID，1:RGB 图（免费版只支持 RGB 图），2:深度图，3:灰度图。

“TargetCopter”传感器装载的目标飞机的 ID，可改变。

“TargetMountType”代表坐标类型，0：固定飞机上（相对几何中心），1：固定飞机上（相对底部中心），2：固定地面上（监控）也可变。

“DataWidth”为数据或图像宽度此处为 640，“DataHeight”为数据或图像高度此处为 480。

“DataCheckFreq”检查数据更新频率此处为 30HZ。

“SendProtocol[8]”为传输方式与地址，SendProtocol[0]取值 0：共享内存（免费版只支持共享内存），1：UDP 直传 png 压缩，2：UDP 直传图片不压缩，3：UDP 直传 jpg 压缩；SendProtocol[1-4]：IP 地址；SendProtocol[5]端口号。

“CameraFOV”为相机视场角（仅限视觉类传感器），单位度也可改变。

“SensorPosXYZ[3]”为传感器安装位置，单位米也可改变。

“SensorAngEular[3]”为传感器安装角度，单位度°也可改变。

然后通过 ue.sendUE4Cmd 创建物体，然后通过 mav.SendPosNED 控制机体飞行到指定位置

ue.sendUE4Cmd 是向 RflySim3D 发送一个“命令”，其中 cmd 是一个字节字符串，表示发送给 RflySim3D 的命令，当发送的‘命令’不同时，将会产生不同的效果

1. RflyShowTextTime(String txt, float time)\\ 让 UE 显示 txt，持续 time 秒
2. RflyShowText(String txt)\\ 让 UE 显示 txt，持续 5 秒
3. RflyChangeMapbyID(int id)\\ 根据地图的 ID 切换 RflySim3D 场景地图
4. RflyChangeMapbyName(String txt)\\ 根据地图名切换 RflySim3D 场景地图
5. RflyChangeViewKeyCmd(String key, int num) \\ 与在 RflySim3D 中按一个 key + num 效果一致
6. RflyCameraPosAngAdd(float x, float y, float z,float roll,float pitch,float yaw) \\ 给摄像机的位置与角度增加一个偏移值
7. RflyCameraPosAng(float x, float y, float z, float roll, float pitch, float yaw) \\ 设置摄像机的位置与角度(UE 的世界坐标)
8. RflyCameraFovDegrees(float degrees) \\ 设置摄像机的视域体 FOV 角度
9. RflyChange3DModel(int CopterID, int veTypes=0) \\ 修改一个无人机的模型样式
10. RflyChangeVehicleSize(int CopterID, float size=0) \\修改一个无人机的缩放大小

11. RflyMoveVehiclePosAng(int CopterID, int isFitGround, float x, float y, float z, float roll, float pitch, float yaw) \\ 给无人机的位置与角度设置一个偏移值, isFitGround 设置无人机是否适应地面
12. RflySetVehiclePosAng(int CopterID, int isFitGround, float x, float y, float z, float roll, float pitch, float yaw) \\ 设置无人机的位置与角度
13. RflyScanTerrainH(float xLeftBottom(m), float yLeftBottom(m), float xRightTop(m), float yRightTop(m), float scanHeight(m), float scanInterval(m)) \\ 扫描地形, 生成一个 png 的高度图与 txt, CopterSim 程序会需要它才知道 UE 有哪些地形、以及它们的高程
14. RflyCesiumOriPos(double lat, double lon, double Alt) \\ 根据经纬度修改 Cesium 的原点位置
15. RflyClearCapture() \\ 清空抓取的图像
16. RflySetActuatorPWMs(int CopterID, float pwm1, float pwm2, float pwm3, float pwm4, float pwm5, float pwm6, float pwm7, float pwm8); \\ 传入 8 个值, 并触发目标无人机的蓝图的接口函数
17. RflySetActuatorPWMsExt(int CopterID, float pwm9, float pwm10, float pwm11, float pwm12, float pwm13, float pwm14, float pwm15, float pwm16, float pwm17, float pwm18, float pwm19, float pwm20, float pwm21, float pwm22, float pwm23, float pwm24); \\ 传入 16 个值, 并触发目标无人机的蓝图接口函数。该函数需要完整版才能有作用
18. RflyReqVehicleData(FString isEnabled); 如果 'isEnabled' 不为 0, 则 RflySim3D 会开始发送所有 Copter 的数据 (就是前面介绍的 reqVeCrashData)。
19. RflySetPosScale(float scale); \\ 全局位置的缩放
20. RflyLoad3DFile(FString FileName); \\ 加载并执行路径下的 TXT 脚本文件
21. RflyReqObjData(int opFlag, FString objName, FString colorStr); \\ 请求获取三维场景中物体的数据
22. RflySetIDLabel(int CopterID, FString Text, FString colorStr, float size); \\ 设置一个 Copter 的头顶 ID 显示内容 (默认显示 CopterID)
23. RflySetMsgLabel(int CopterID, FString Text, FString colorStr, float size, float time, int flag); \\ 设置一个 Copter 头顶的 Message 显示的内容
24. RflyDelVehicles(FString CopterIDList); \\ 删除一些 Copter (逗号是分隔符)
25. RflyDisableVeMove(FString CopterIDList, int disable); \\ 拒接接收指定 ID 的 Copter 的信息 (逗号是分隔符)
26. 除此之外, 还有一些 UE 内置的命令可以使用, 例如 'stat fps' 可以显示当前帧率, 't.Maxfps 60' 可以设置最大帧率为 60。

3、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。

4、文件目录

文件夹/文件名称	说明
ThreeCtrlModesSITL.bat	软件在环仿真实验脚本
ThreeCtrlModesHITL.py	硬件在环仿真实验脚本

ThreeCtrlModes_PosCtrl.py	Python 实验脚本
ThreeCtrlModes_PosCtrlFRD.py	Python 实验脚本
ThreeCtrlModes_VelCtrlBody.py	Python 实验脚本
ThreeCtrlModes_VelCtrlEarth.py	Python 实验脚本
Config.json	视觉传感器配置文件

5、运行环境

序号	软件要求	硬件要求	
		名称	数量(个)
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台免费版及以上	卓翼 H7 飞控 ^②	1
3	Visual Studio Code	MicroUSB 线	1

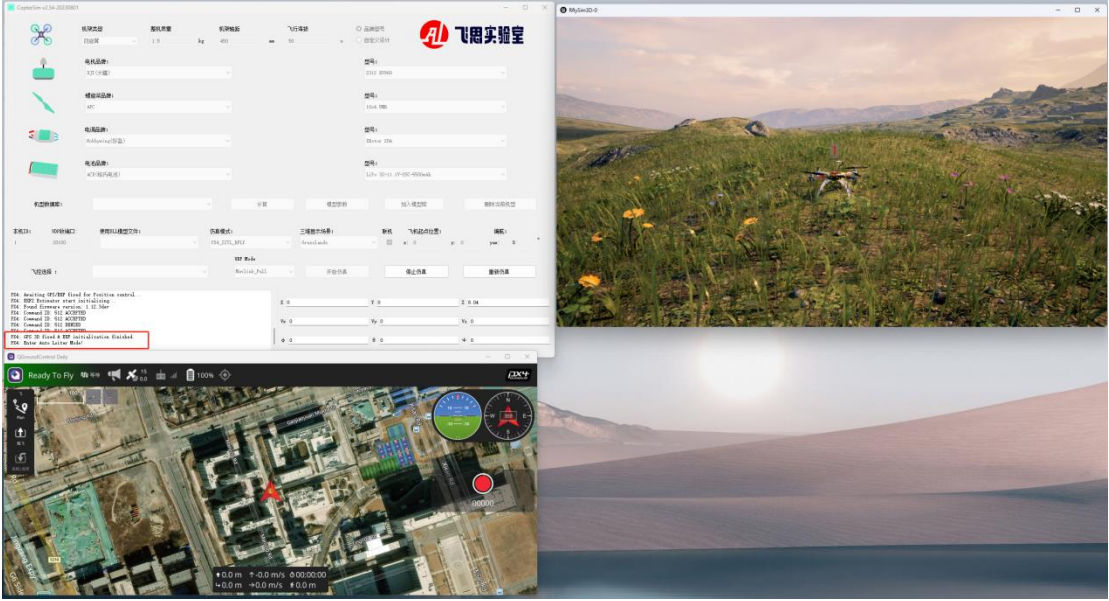
- ①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>
- ②：须保证平台安装时的编译命令为：droneyee_zyfc-h7_default，固件版本为：1.12.1。其他配套飞控请见：<http://doc.rflysim.com/hardware.html>

6、运行步骤

软件在环步骤：

Step 1:

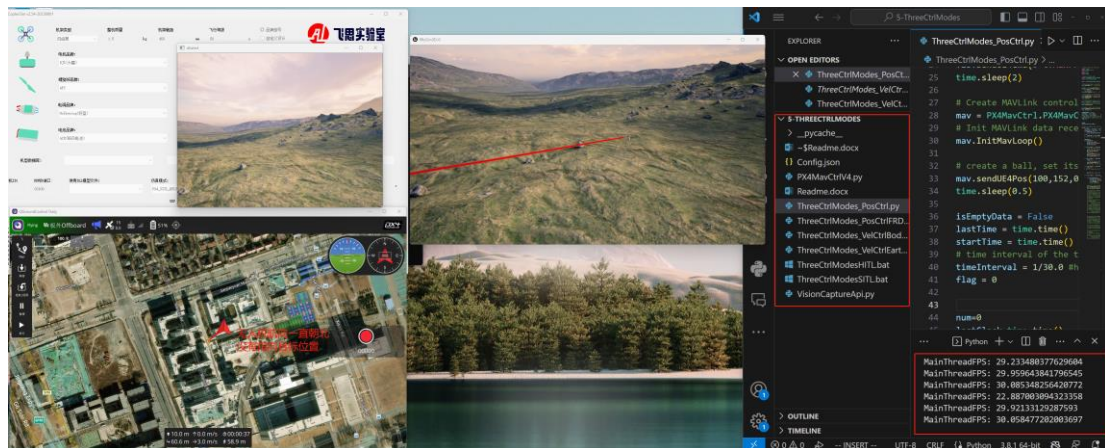
以管理员方式运行 ThreeCtrlModesSITL.bat 开启一个飞机的软件在环仿真。将会启动 1 个 QGC 地面站，1 个 CopterSim 软件且其软件下侧日志栏必须打印出 GPS 3D fixed & EKF initialization finished 字样代表初始化完成，并且 RflySim3D 软件内有 1 架无人机。



Step 2:

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。



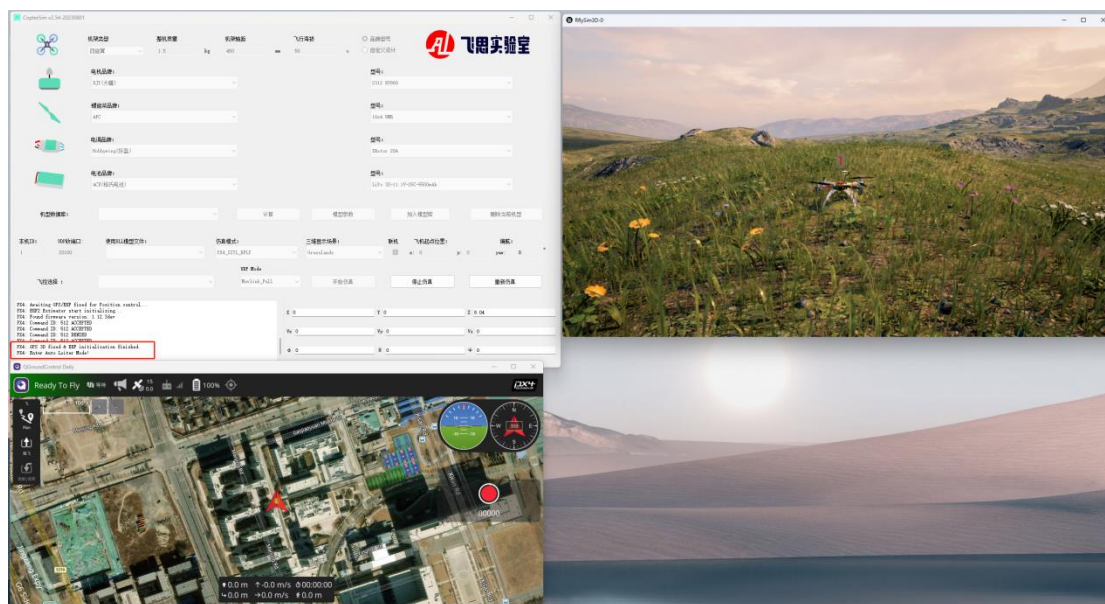
Step 3:

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：

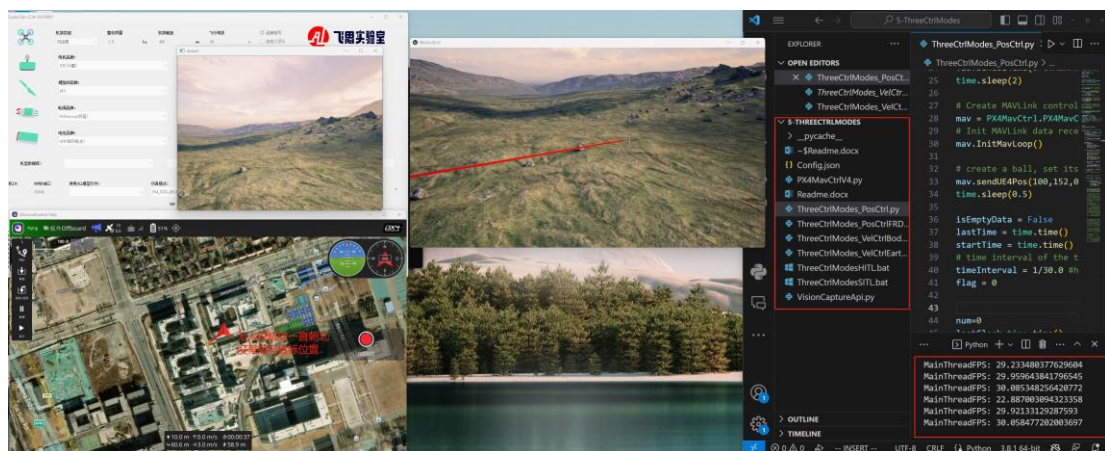
Step 1:

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 数据线将卓翼 H7 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:

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。



Step 3:

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:

在下图“`VisionCapAPIDemo.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。

```
C:\WINDOWS\system32\cmd.exe

-----
Start QGroundControl
Kill all CopterSims
Starting PX4 Build
[1/1] Generating ../../logs
killing running instances
starting instance 1 in /mnt/c/PX4PSPFull/Firmware/build/px4_sitl_default/instance_1
PX4 instances start finished
Press any key to exit
```

按下回车键，快速关闭所有仿真窗口

Step 5:

在下图 VScode 中，点击“终止终端”，可以彻底退出脚本运行。



7、参考文献

[1]. 无

8、常见问题

Q1: 无

A1: 无