

1.实验名称及目的

1.1.实验名称

平台直接输出目标视觉伺服控制无人机跟踪算法实验

1.2.实验目的

通过平台直接输出的目标结果进行输入，通过视觉伺服控制无人机飞行，进行高机动跟随。

1.3.关键知识点

目标跟随算法输入时目标，在RflySim平台里，目标可以是目标检测算法的输出，也可以目标跟踪算法的输出，还可以是平台直接输出的目标，如果仅仅调试目标跟踪算法，选择平台直接输出的目标结果。该例程则是采用平台直接输出的目标结果进行输入，通过视觉伺服控制无人机飞行，对于高机动跟随，需要参考无人机动力学相关知识，平台资料-控制篇有详细说明。

本实验主要是实现通过Python接口[VisionCaptureApi.py](#)（见RflySimAPIs\RflySimSDK\vision目录）获取RflySim3D图像并实时更新相机参数（姿态、位置、FOV等），通过平台直接输出的目标结果进行输入，通过视觉伺服控制无人机飞行，进行高机动跟随。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 cv2.rectangle(img, p1, p2, (255, 0, 0)) \#画出一个长方形, 在img图像里, 左上角右下角的坐标分别
12
13 cv2.imshow('Img'+str(i),vis.Img[i]) \# 显示图片i图像
14
15 tracker = cv2.TrackerTLD_create() \# 使用 OpenCV 中的 TLD跟踪器创建一个跟踪对象tracker
```

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] : 相机分布。
```

3) 飞机控制指令

```
1 MavList = MavList+[PX4MavCtrl.PX4MavCtrler(1+i)] \#初始化并建立i号飞机的MAVLink通信连接, 这
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 MavList[i].InitTrueDataLoop() \# 通过 30100 系列端口初始化来自 CopterSim 的 UDPTue 数据监
8
9 MavList[i].SendPosNED(0, 0, -10, 0) \# 对第i个飞机发送10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
```

4) UE控制

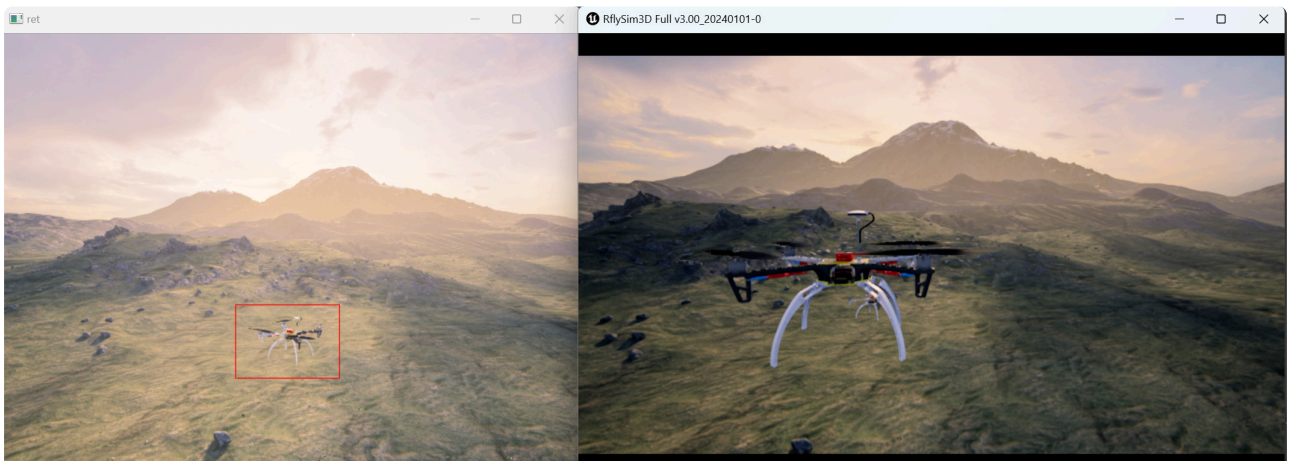
接口详细使用方法见：UE4CtrlAPI.py

```
1 ue = UE4CtrlAPI.UE4CtrlAPI() \# 创建UE控制实例
2
3 ue.sendUE4Cmd('r.setres 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.sendUE4Pos(110, 501, 0, [3, 0, -1], [0, 0, 0]) \#创建一个物体，if号为110，类型为501，速
```

5) 其余代码说明

```
1 def CtrlTargetDrone(mav: PX4MavCtrl.PX4MavCtrl, max_vel) \#用于控制无人机（目标无人机）
2
3 def GetFrame(target_drone: UE4CtrlAPI.CoptReqData, camera:UE4CtrlAPI.CameraData) \#用
4
5 def CtrlDrone(mav: PX4MavCtrl.PX4MavCtrl, camera: UE4CtrlAPI.CameraData,
6 rect_xywh: tuple) \#通过分析摄像头捕获的目标框信息，计算出控制指令，实现了对无人机的实时跟随控制，
7
8 target_th = threading.Thread(target=CtrlTargetDrone, args=(MavList[1], 1)) \#创建一个线程
9
10 target_th.start() \# 启动 target_th 线程，开始执行 CtrlTargetDrone函数中定义的任务
11
12 timeInterval = 1/20 \# 以20hz的频率进行控制
13
14 lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
15
16 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



3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\8.RflySimVision\2.AdvExps\e9_Object-Follow](#)

文件夹/文件名称	说明
target_follow.bat	启动仿真配置文件
target_follow.py	Python实验脚本
Config.json	视觉传感器配置文件
Python38Run.bat	Python程序运行脚本

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_fmuv6x_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。

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

5. 实验步骤

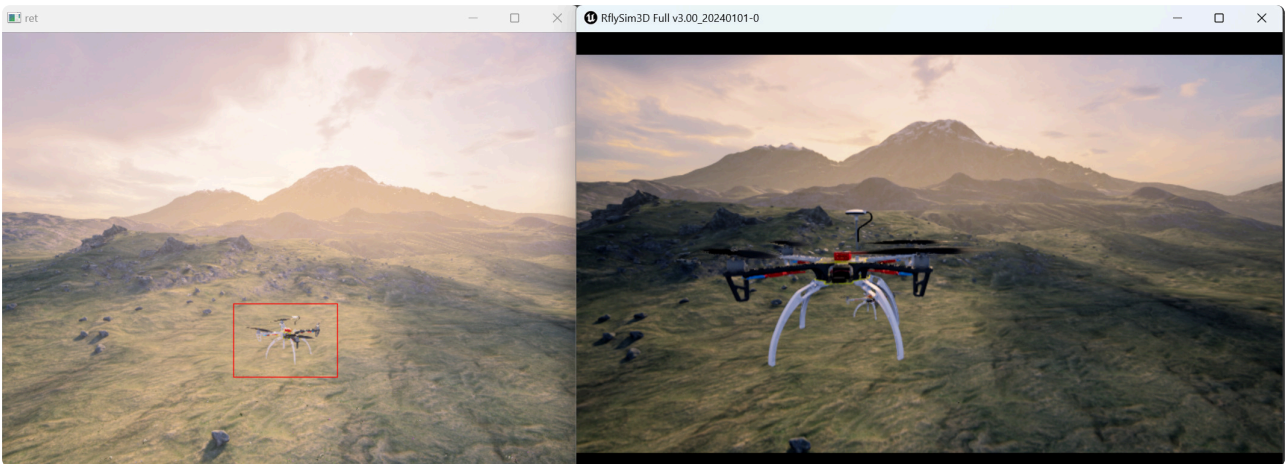
5.1. 必做实验：Windows取图控制

Step 1: 运行控制程序

1. 双击运行[target_follow.bat](#) 脚本。

2. 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 `target_follow.py` 可看到效果。

Step 2: 观察结果



```
center: [316, 308]
dis_err [ 4. -68.]
width_ 32.49694906138592
vx,vy,vz,yawrate 0.9749084718415776
copter_id: 1
center: [317, 307]
dis_err [ 3. -67.]
width_ 33.261498867084185
vx,vy,vz,yawrate 0.9978449660125255
copter_id: 1
center: [317, 305]
dis_err [ 3. -65.]
width_ 34.43109381351252
vx,vy,vz,yawrate 1.0329328144053755
copter_id: 1
center: [315, 305]
dis_err [ 5. -65.]
width_ 34.20915438496394
vx,vy,vz,yawrate 1.0262746315489182
copter_id: 1
center: [315, 305]
dis_err [ 5. -65.]
width_ 34.55905564605763
vx,vy,vz,yawrate 1.0367716693817288
copter_id: 1
```

Step 3: 结束仿真

在下图“`target_follow.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 [target_follow.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [target_follow.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 [target_follow.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。

```

VisionCapAPIDemo.py ×
> RflySimAPIs > 8.RflySimVision > 0.ApiExps > 1-UsageAPI > 0.VisionSensorAPI > 1.Came
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，；
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