

# 1. 实验名称及目的

## 1.1. 实验名称

轻量级无人机模型视觉穿环实验

## 1.2. 实验目的

基于质点模型的穿环实验例程。

## 1.3. 关键知识点

在Python中开发了一个基于质点的无人机控制模型，能够提供软硬件在环仿真相近的无人机动态效果，但是极大降低对电脑性能的占用和提升飞机平稳性。

首先，通过`mav.initPointMassModel()`创建了一架质点无人机模型，其可使用参数如下

```
1 | intAlt : 初始高度  
2 |  
3 | intState : 初始 X (米), Y (米), Yaw (角度)
```

开启取图从而获得从仿真图像，通过定义的函数找到相应环所在位置，并计算穿环位置，将穿环位置发送给飞机从而使得飞机向穿环位置飞行，完成穿环。

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），实现基于质点模型的穿环实验例程。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_bgr", img_bgr) \# 显示图片img_bgr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的视觉传感器
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":[1,0,0,0,0,0,0] : 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":[ 0.3,0,0] : RGB相机分布。
```

## 3) 飞机控制指令

```
1 mav = PX4MavCtrl.PX4MavCtrler(1) \# 创建飞机控制实例
2
3 mav.initPointMassModel() \# 初始化并启动无人机控制的点质量模型
4
5 mav.SendCopterSpeed(2) \# 飞机的速度为2m/s
6
7 mav.SendVelFRD(vx, vy, vz, yawrate) \#当飞行器向上飞行时，将飞行器目标速度 (m/s) 发送到车身前
8
9 mav.SendMavArm(True) \# 解锁飞控
10
11 mav.SendPosNED(0, 0, -10, 0) \# 发送10米高的位置控制指令
```

## 4) 实时修改相机参数（姿态位置等）

```
1 vs = vis.VisSensor[0] \#获取第0号相机基本参数\# 修改其中的可变部分，只修改需要改变的部分即可
2
3 vs.TargetCopter=1 \#修改视角绑定的飞机ID
4
5 vs.TargetMountType=0 \# 修改视角绑定类型，固连飞机还是地面
6
7 vs.CameraFOV=90 \# 修改视角的视场角（焦距），可以模拟对焦相机
8
9 vs.SensorPosXYZ=[0.3, -0.15, 0] \# 修改相机的位置，可以调整相机初始位置
10
11 vs.SensorAngEular=[0, 0, 0] \# 修改相机的姿态，可以模拟云台转动
12
13 vis.sendUpdateUEImage(vs) \# 发送更新数据
```

注意：相机宽高、相机类型等数据没法实时修改。

## 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) 其余代码说明

```
1 def angle_cos(p0, p1, p2) \# 计算三个点形成的向量之间的夹角的余弦值
2
3 def diagonal_check(p) \# 计算一个四边形的两条对角线的长度差异
4
5 def saturationYawRate(yaw_rate) \# 限制偏航速率在 -20.0 和 20.0 之间
6
7 def taskChange(pos_x) \# 根据当前位置 pos_x 决定当前任务
8
9 def sat(inPwm, thres=1) \# 将输入的 PWM 信号限制在 [-1, 1] 范围内
10
11 def objectDetect(task) \# 根据当前任务检测图像中的目标
12
13 def squareDetect(img_bgr, img_edge) \#检测图像中的方形, 并返回其中心和对角线长度
14
15 def circleDetect(img_bgr, img_edge, img_b) \#使用霍夫变换检测图像中的圆形, 并返回其中心和半径
16
17 def approachObjective() \# 控制无人机接近目标的全过程函数
18
19 timeInterval = 1/30.0 \# 以30hz的频率进行控制
20
21 lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
22
23 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

轻量级无人机质点模型飞行控制效果与软/硬件在环相近, 但更平稳。

## 3. 文件目录

例程目录:

[\[安装目录\]\RflySimAPIs\8.RflySimVision\1.BasicExps\1-VisionCtrlDemos\e1\\_CrossRingNoPX4](#)

文件夹/文件名称	说明
<a href="#">CrossRingNoPX4.bat</a>	一键仿真启动脚本
<a href="#">CrossRingNoPX4.py</a>	Python实验代码
Config.json	视觉传感器配置文件

文件夹/文件名称	说明
<a href="#">Python38Run.bat</a>	Python环境启动脚本

## 4. 运行环境

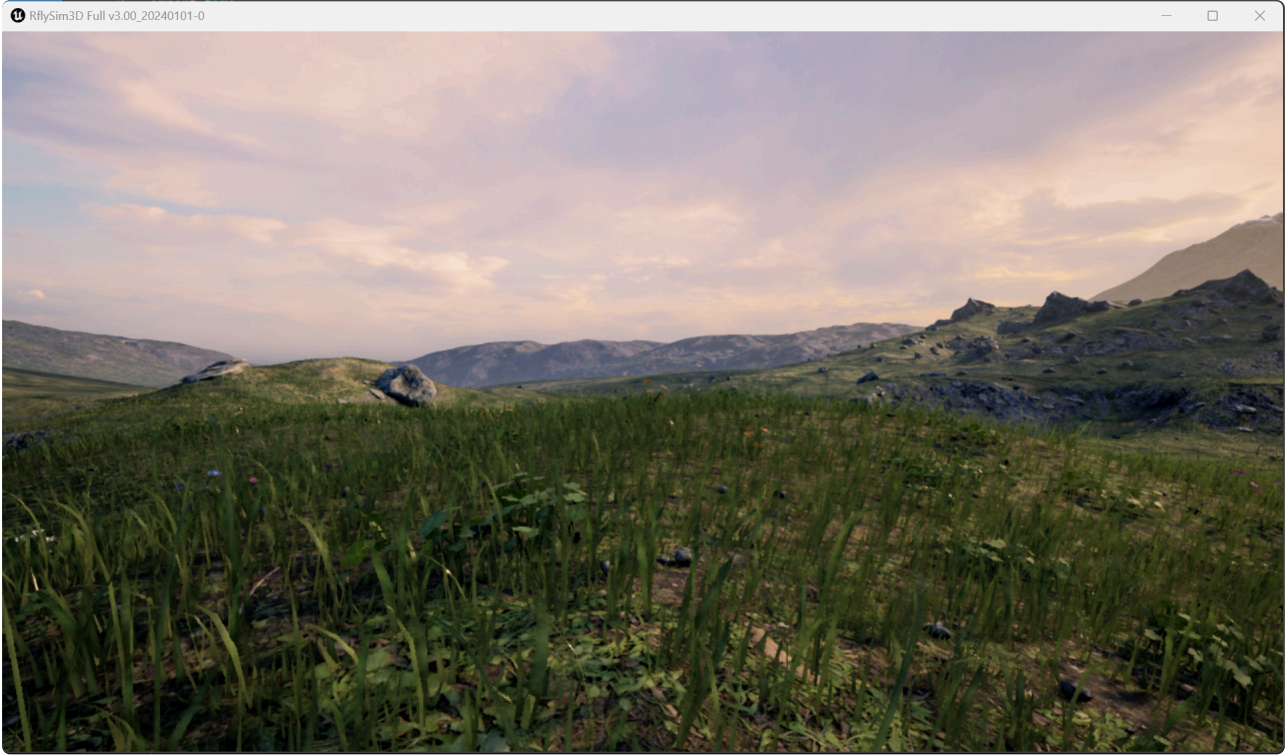
序号	软件要求	硬件要求	
		名称	数量 (个)
1	Windows 10及以上版本	笔记本/ 台式电脑①	1
2	RflySim工具链		
3	VS Code		可选
**： **推荐配置请见： <a href="https://rflysim.com">https://rflysim.com</a>			

## 5. 实验步骤

### 5.1. 必做实验：Windows取图控制

#### Step 1：开启仿真

双击运行 [CrossRingNoPX4.bat](#)。可以看到打开一个RflySim3D窗口，没有其他程序打开。



## Step 2: 运行控制程序

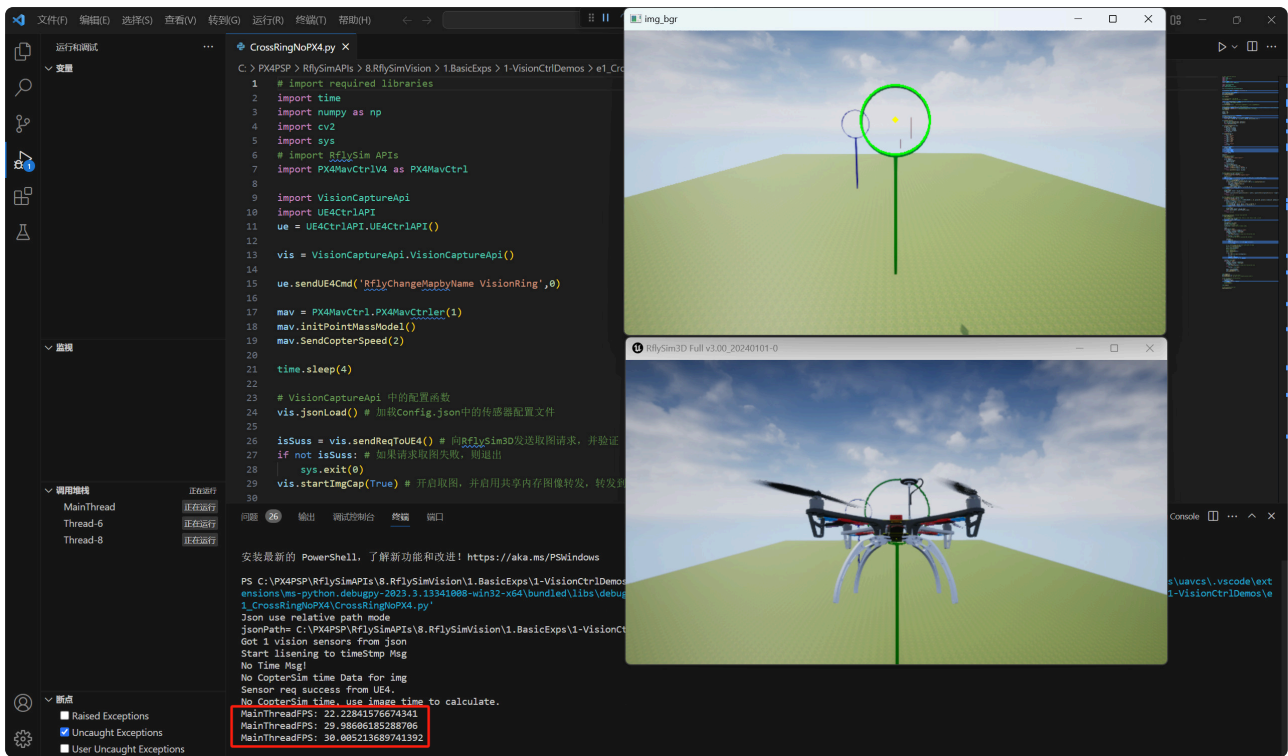
在文件夹下，双击 `Python38Run.bat`，打开集成好的python环境，在该环境下运行 “`CrossRingNoPX4.py`”，输入 `python CrossRingNoPX4.py` 可以看到场景切换到草地穿环场景，生成了一个多旋翼飞机，起飞后依次穿越三个环。

```
C:\Windows\system32\cmd.e  x  +  v
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

C:\Users\uavcs\Desktop\demo\8.RflySimVision\1.BasicExps\1-VisionCtrlDemos\e1_CrossRingNoPX4>python CrossRingNoPX4.py|
```

## Step 3: 观察结果

可以看到场景切换到草地穿环场景，生成了一个多旋翼飞机，起飞后依次穿越三个环。



## Step 4: 结束仿真

在下图“CrossRingNoPX4.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\e3.PythonConfig\Readme.pdf](#) 步骤，正确配置VSCode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 [CrossRingNoPX4.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [CrossRingNoPX4.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 [CrossRingNoPX4.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.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, 同时
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