

基于视觉接口(VisionCaptureApi)控制 撞击小球实验

1. 实验目的

通过调用平台接口进行对RflySim3D软件内图像的捕获，并利用opencv进行图像处理，并进行控制指令解算，控制无人机运动。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链^[1]。
- 硬件要求：笔记本/台式电脑1台^[2]。

3. 实验地址

例程目录：

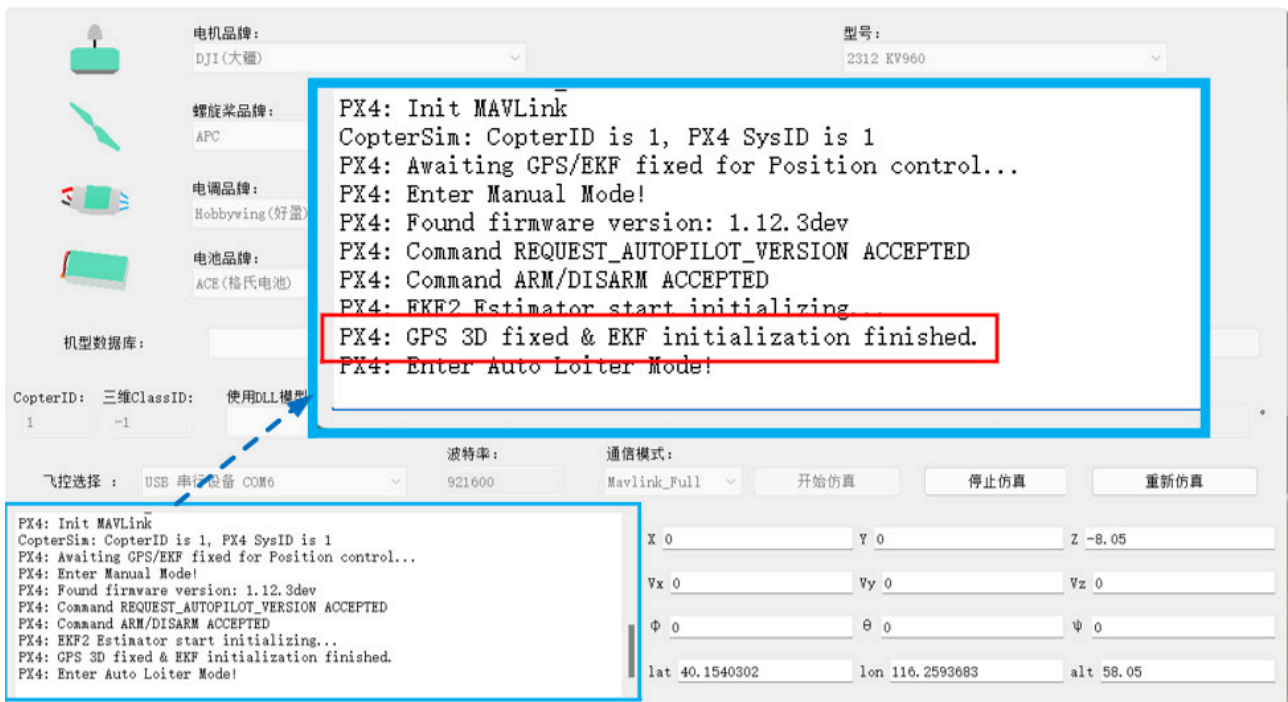
[\[安装目录\]\RflySimAPIs\6.RflySimExtCtrl\0.ApiExps\e9_UDPMode1TestShootBall](#)

- [ShootBall3SITL.bat](#)：软件在环仿真启动脚本
- [ShootBall3HITL.bat](#)：硬件在环仿真启动脚本
- [ShootBall3.py](#)：基于视觉识别的撞击小球主程序
- [Config.json](#)：VisionCaptureApi传感器配置文件
- [Python38Run.bat](#)：Python3.8环境启动脚本

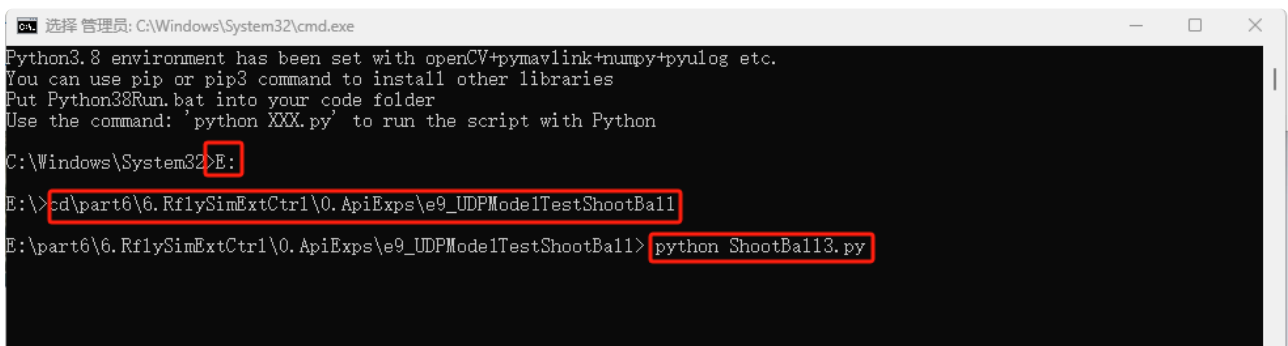
4. 实验内容或步骤

4.1 步骤1：软件在环

双击运行 `ShootBall3SITL.bat` 文件，等待仿真环境初始化完成。脚本将会启动 1 个 QGC 地面站，1 个 CopterSim、1 个 RflySim3D 软件，等待CopterSim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。如下图所示：



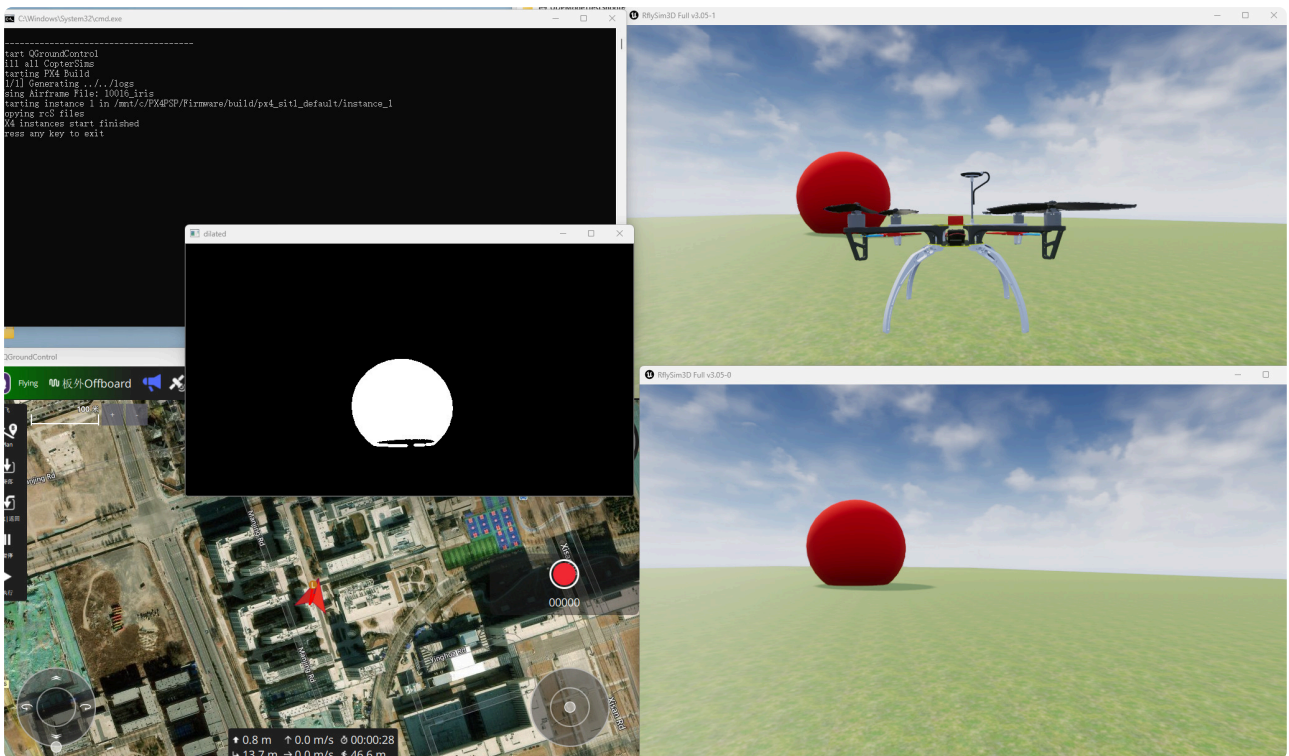
双击运行 `Python38Run.bat`，打开集成好的python环境，在该环境下运行 `ShootBall3.py` 文件即可，即输入如下命令：`python ShootBall3.py`



注：也可使用VS Code来调试运行。

运行程序，一个是前置摄像头，一个是上帝视角观测。在图像提取，opencv进行图像处理，控制指令解算后，可看到无人机向红球撞击，然后在cmd终端上就会出现无人机的仿真

状态数据。注意：在RflySim3D窗口按T键开启或关闭飞机轨迹记录功能，T+数字*开启/更改轨迹粗细为*号。



在下图"ShootBall3.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。

4.2 步骤2：硬件在环

本实验在进行之前需要将飞控内部固件进行还原并确保可正常进行硬件在环仿真，具体还原方式可见例

程：`[RflySim安装路径]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e2.FCUIntro`。按照如下图所示，进行硬件链接并接入电脑。



双击运行 `ShootBall3HITL.bat` 文件，在弹出得对话框中输入飞控COM号如：4。

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

-----
Please input the Pixhawk COM port list for HIL
Use ',' as the separator if more than one Pixhawk
E.g., input 3 for COM3 of Pixhawk on the computer
Input 3,6,7 for COM3, COM6 and COM7 of Pixhawks
-----
All COM ports on this computer are:

COM3: Intel(R) Active Management Technology - SOL (unavailable or busy)
COM4: USB 串行设备 * (Pixhawk with SysID=1)

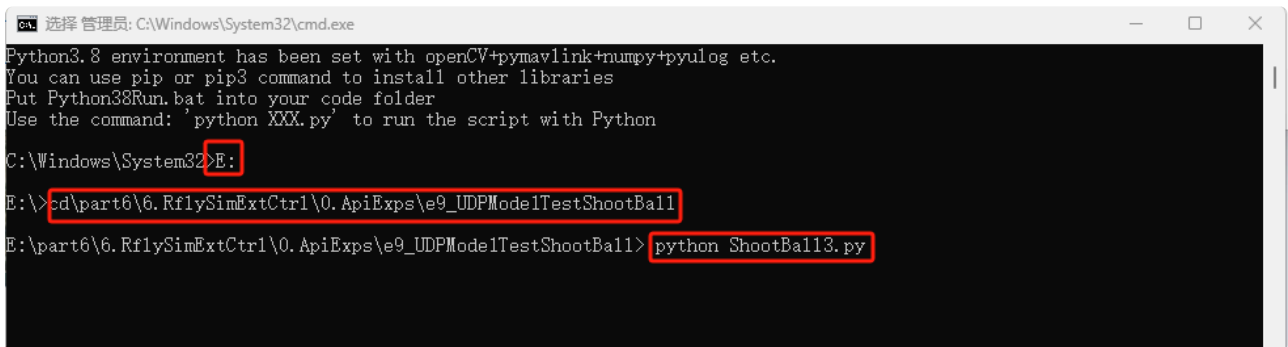
-----
Recommended COM list input is: 4
-----

My COM list for HITL simulation is: _
```

等待仿真环境初始化完成。脚本将会启动 1 个 QGC 地面站，1 个 CopterSim、1 个 RflySim3D 软件，等待CopterSim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。如下图所示：

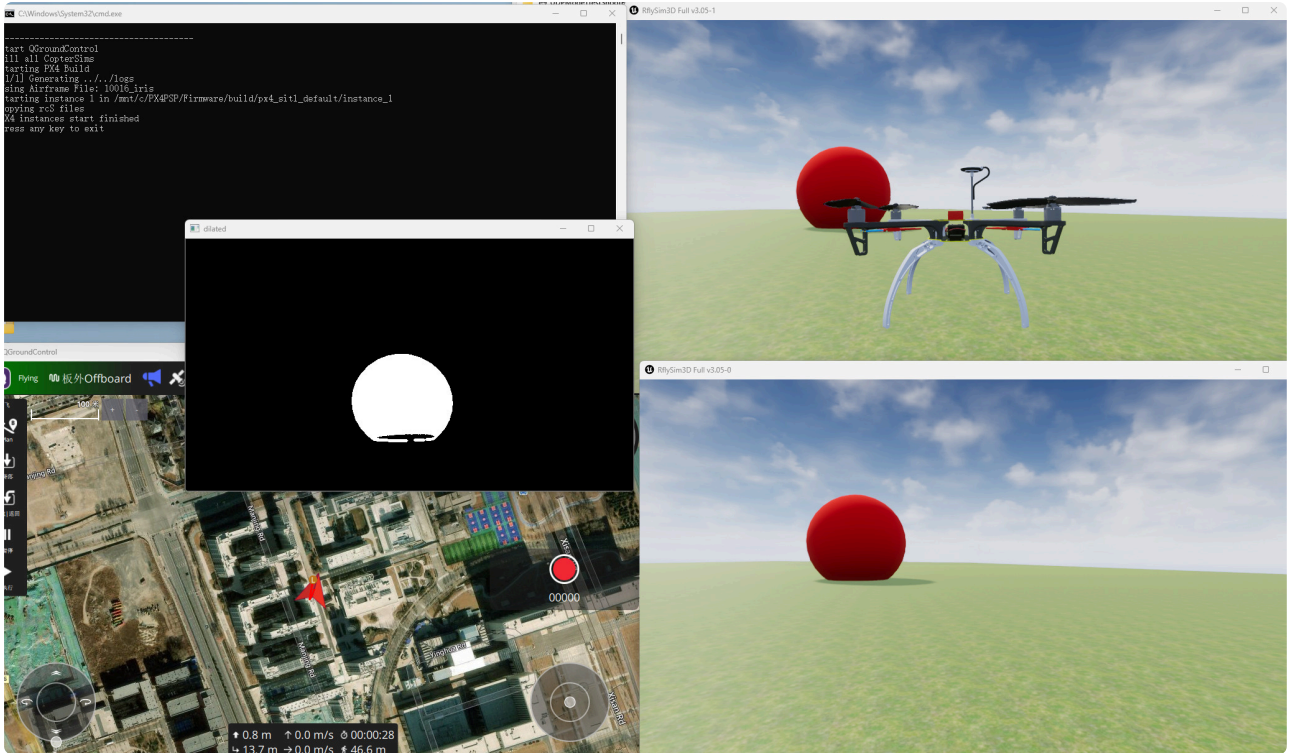


双击运行 `Python38Run.bat`，打开集成好的python环境，在该环境下运行 `ShootBall3.py` 文件即可，即输入如下命令：



运行程序，一个是前置摄像头，一个是上帝视角观测。在图像提取，opencv进行图像处理，控制指令解算后，可看到无人机向红球撞击，然后在cmd终端上就会出现无人机的仿真状态数据。

注意：在RflySim3D窗口按T键开启或关闭飞机轨迹记录功能，T+数字*开启/更改轨迹粗细为*号。



在下图"ShootBall3.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. 关键知识点

通过平台图像提取文件VisionCaptureApi中的相关接口进行对RflySim中的图像提取，然后在函数calc_centroid(img_bgr)中通过opencv中进行对图像识别出质心及半径，并计算出相应的位置结果。然后在函数controller(p_i)中根据位置结果计算出无人机的控制指令控制无人机运动。关键代码解析如下：

关键知识点1：飞行控制指令

```

1 | mav = PX4MavCtrl.PX4MavCtrl(1) # 创建飞机控制实例
2 | mav.InitMavLoop() # 初始化MavLink监听程序，读取飞机数据
3 | mav.initOffboard() # 进入Offboard模式
4 | mav.SendMavArm(True) # 解锁飞控
5 | mav.SendPosNED(0, 0, -5, 0) # 发送5米高的位置控制指令
6 | mav.SendVelFRD(ctrl[0], ctrl[1], ctrl[2], ctrl[3]) # 发送速度指令

```

关键知识点2：CV视觉

```

1 | th = cv2.inRange(img, low_range, high_range) # 创建一个二值图像
2 | dilated = cv2.dilate(th, cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3)),
3 | iterations=2) # 膨胀操作, 用于去除噪声并填充小孔
4 | cv2.imshow("dilated", dilated) # 在窗口中显示图像
5 | cv2.waitKey(1) # 等待键盘事件
   | M = cv2.moments(dilated, binaryImage=True) # 计算图像的矩, 并用于计算质心

```

关键知识点3: 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 | 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, 取图分辨率在json中配置, 本窗口设置越小, 资源需求越少。
3 |
4 | ue.sendUE4Cmd('t.MaxFPS 30', 0) # 设置UE4最大刷新频率, 同时也是取图频率
   | ue.sendUE4Pos(100, 152, 0, [3, 0, -2], [0, 0, 0]) # 设置红色球的初始位置和角度

```

关键知识点4: VisionCaptureApi视觉取图

```

1 | vis = VisionCaptureApi.VisionCaptureApi() # 创建视觉取图实例
2 | vis.jsonLoad() # 加载Config.json中的传感器配置文件
3 | isSuss = vis.sendReqToUE4() # 向RflySim3D发送取图请求, 并验证
4 | vis.startImgCap() # 开启共享内存取图
5 | img_bgr = vis.Img[0] # 获取第0号传感器的图像 (需先判断vis.hasData[0]为True)

```

6.参考资料

1. PX4MavCtrlV4.py是通过Mavlink到Pixhawk(和通过UDP到UE4的通信API)
2. VisionCaptureApi.py是视觉取图API, 能够从RflySim3D请求并获取图片, 详见第8章。
3. [RflySim官方文档](#)

7.常见问题

Q1: 无

A1: 无

1. <https://rflysim.com/> ↩

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