

## 1、实验名称及目的

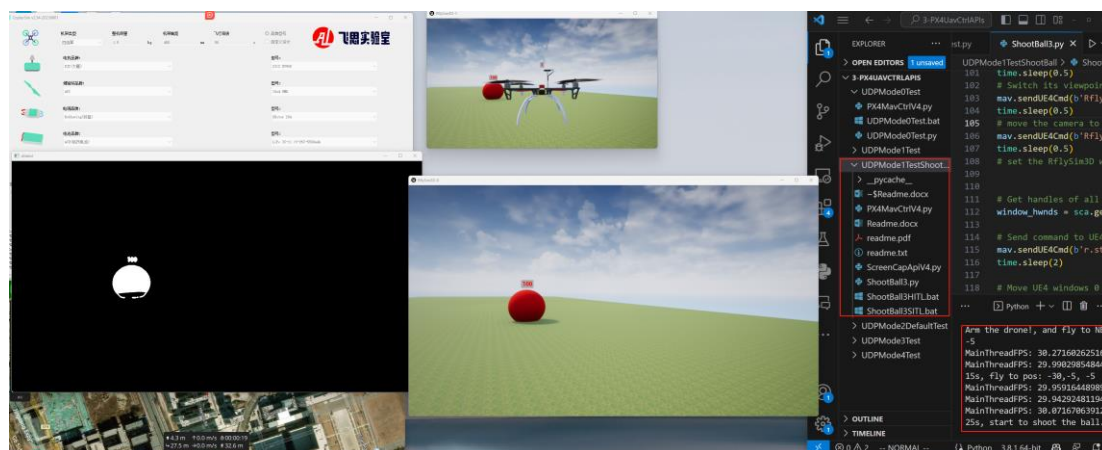
**视觉控制撞击小球实验：**通过调用平台接口进行对 RflySim3D 软件内图像的捕获，并利用 opencv 进行图像处理，并进行控制指令解算，控制无人机运动。

## 2、实验原理

通过平台图像提取文件 ScreenCapApiV4 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。

## 3、实验效果

通过利用 RflySim 平台接口，运行 python 程序提取 RflySim3D 场景图像，并利用 open cv 对提取的图像进行处理，并解算出控制指令让无人机撞向红色小球。



## 4、文件目录

文件夹/文件名称	说明
ShootBall3SITL.bat	启动仿真配置文件
ShootBall3.py	功能主程序文件

## 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 <sup>①</sup>	1
2	RflySim 平台免费版及以上	Pixhawk 6C 飞控 <sup>②</sup>	1
3	Visual Studio Code	MicroUSB 线	1

①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

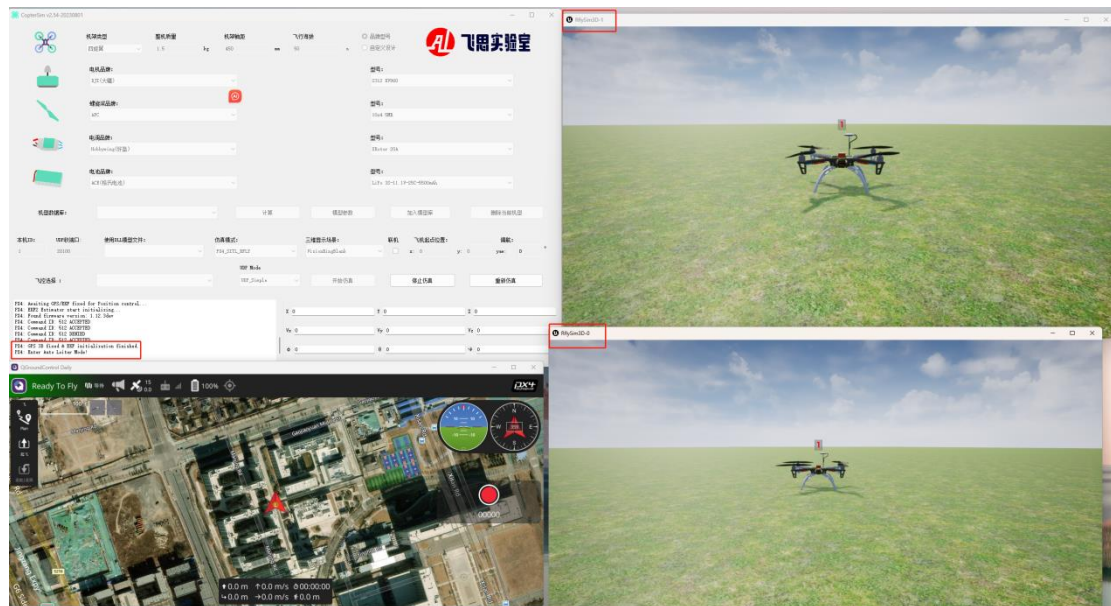
②：须保证平台安装时的编译命令为：px4\_fmu-v6c\_default，固件版本为：1.13.3。其他配套飞控请见：<http://doc.rflysim.com/hardware.html>

## 6、实验步骤

软件在环步骤：

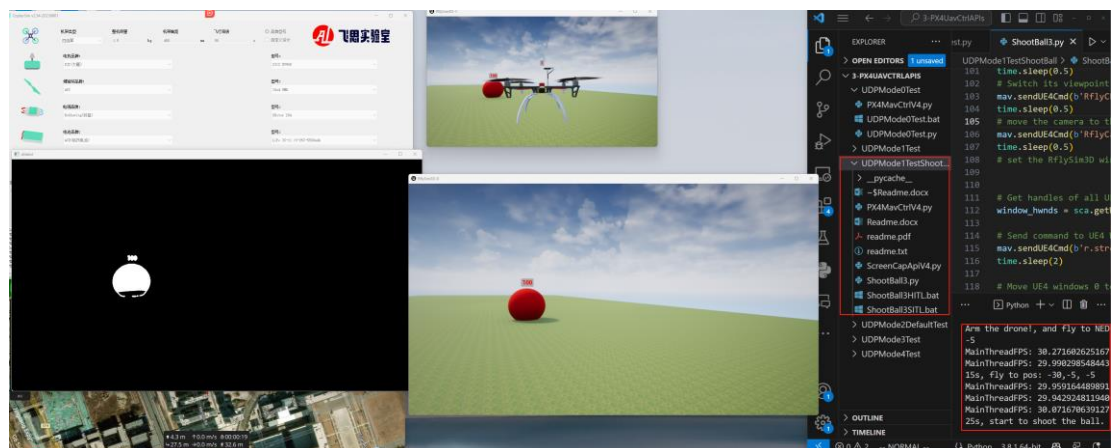
### Step 1:

启动 ShootBall3SITL.bat 脚本将会启动 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，并且有 2 个 RflySim3D 软件各有 1 架无人机。如下图所示：



### Step 2:

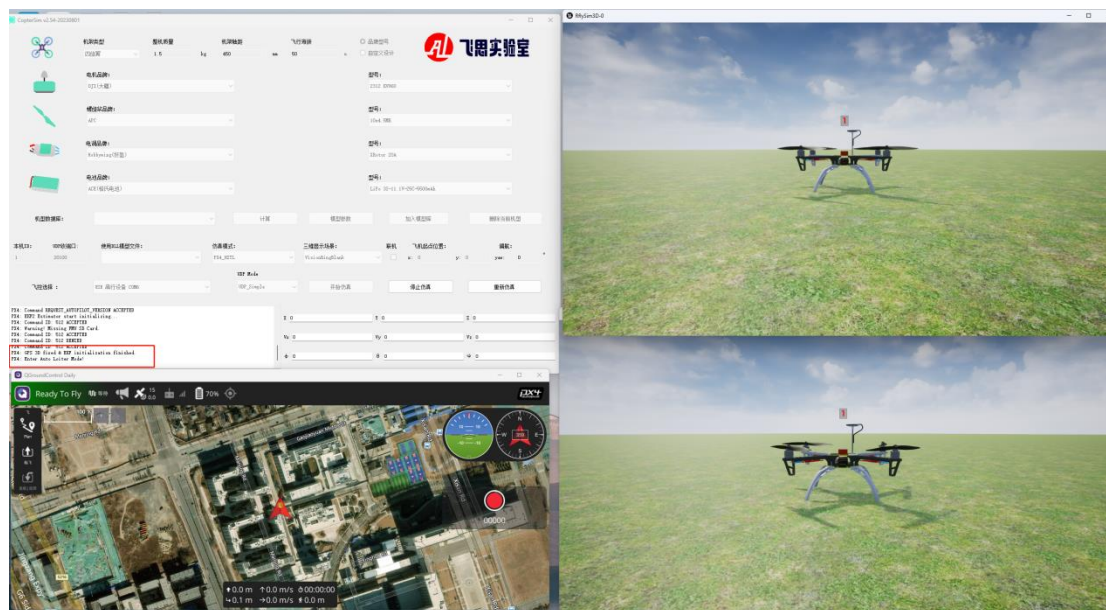
用 VSCode 打开到本实验路径文件夹，打开 ShootBall3.py，并运行程序，一个是前置摄像头，一个是上帝视角观测。在图像提取，opencv 进行图像处理，控制指令解算后，可看到无人机向红球撞击。并且按 T 键开启或关闭飞机轨迹记录功能，T+数字\*开启/更改轨迹粗细为\*号。然后在 VSCode 终端上就会出现无人机的仿真状态数据。



硬件在环步骤：

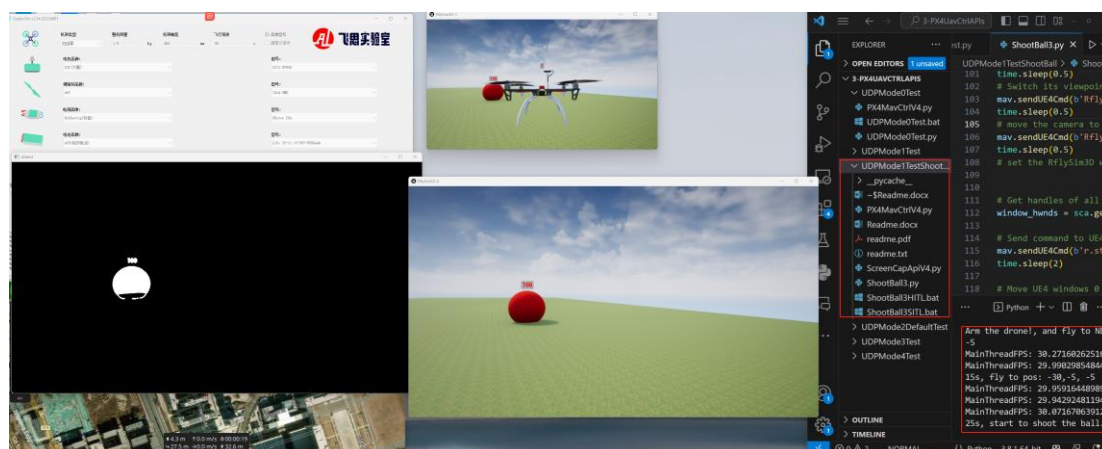
## Step 1:

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 数据线将飞控与电脑连接，以管理员方式运行 ShootBall3HITL.bat，输入端口号开启一个飞机的硬件在环仿真。脚本将会启动 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，并且有 2 个 RflySim3D 软件各有 1 架无人机。如下图所示：



## Step 2:

用 VScode 打开到本实验路径文件夹，打开 ShootBall3.py，并运行程序，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。并且按 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。然后在 VScode 终端上就会出现无人机的仿真状态数据。



## 7、参考文献

[1]. PX4MavCtrlV4.py 是通过 Mavlink 到 Pixhawk(和通过 UDP 到 UE4 的通信 API)

---

[2]. ScreenCapApiV4.py 是屏幕捕获 API。值"isNewUE=False"将启用 RflySim3D 屏幕捕获的旧 API，其速度更快，但不兼容 UE4.23+;值 isNewUE=True"将为 RflySim3D 屏幕捕获启用新的 API，其速度略慢，但兼容所有 UE4 版本。

## 8、常见问题

Q1: 无

A1: 无