

# 1. 实验名称及目的

## 1.1 实验名称

ReqCopterSim.py接口库使用说明

## 1.2 实验目的

以平台进行仿真时，用户不仅可以通过SITL/HITLRun.bat来设置模型初始化相关信息，还可以通过Python的方式进行设置。平台ReqCopterSim.py支持通过Python的方式对CopterSim相关初始化参数进行设置，参数包括：

- 1) 是否联机
- 2) 指定CopterSim 回传数据IP
- 3) 北东地坐标系下的x、y、z
- 4) 姿态，滚转、俯仰、偏航
- 5) 仿真模式
- 6) DLL模型
- 7) DLL序号
- 8) 三维场景
- 9) 三维场景序号

通过该例程熟悉ReqCopterSim.py接口库的使用。

## 1.3 关键知识点

无

## 2. 实验效果

运行 [MulticopterNoCtrl\\_SITLRun.bat](#) 后 [testReqCopterSim.py](#)，会看到以下效果：

- 1、当vscode终端提示“发送联机模式重设命令”时，表明已完成CopterSim消息获取设置，在本例程中设置获取数据的CopterSim ID为1。
- 2、当vscode终端提示“发送UDP仿真模式重设命令”时，在CopterSim界面的“UDP mode”栏可以看到仿真通讯模式从UDP\_Full重设成Mavlink\_full。
- 3、当vscode终端提示“发送XYyaw初始值重设命令”时，可以分别在CopterSim和RflySim3D中看到变化，在CopterSim右下角会看到x为1，y为2，偏航为30°，在RflySim3D中能观察到飞机位置和姿态发生了变化。
- 4、当vscode终端提示“发送XyzRPYaw初始值重设命令”时，可以分别在CopterSim和RflySim3D中看到变化，在RflySim3D中可以看到飞机从地面变化到了空中，并且自由落体直到落在地面上，偏航角为45°。
- 5、当vscode终端提示“发送DLL模型重设命令”时，可以分别在CopterSim和RflySim3D中看到变化，在CopterSim界面的“使用DLL模型文件”一栏中切换为了“HexarotorModelCTRL”，在RflySim3D中能看到显示模型从四旋翼切换为了六旋翼。
- 6、当vscode终端提示“发送map地图重设命令”时，可以分别在CopterSim和RflySim3D中看到变化，在CopterSim界面“三维显示场景”一栏中切换为了VisionRing，同时可以在RflySim3D中场景改为了VisionRing。
- 7、以上修改依次完成后，可以进行正常的初始化与操作，让飞控重新fixed后依然可以进行offboard控制。

## 3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\2.RflySimUsage\0.ApiExps\e3\\_ReqCopterSim](#)

文件夹/文件名称	说明
MulticopterNoCtrl.dll	四旋翼模型dll动态链接库
<a href="#">testReqCopterSim.py</a>	接口测试程序
<a href="#">MulticopterNoCtrl_SITLRun.bat</a>	四旋翼软件在环启动脚本

文件夹/文件名称	说明
HexarotorModelCTRL.dll	六旋翼模型dll动态链接库

## 4. 运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Windows 10及以上版本	笔记本/台式电脑 ①	1
2	RflySim工具链	\	\
3	MATLAB 2017b及以上③	\	\
: 推荐配置请见: <a href="https://rflysim.com/doc/en">https://rflysim.com/doc/en</a>			

## 5. 实验步骤

### Step 1

为了保证sendReSimDllName()函数能成功修改CopterSim当前所调用的DLL模型，在运行testReqCopterSim.py前，需要将目标DLL模型文件复制到 “[安装目录]\CopterSim\external\model” 文件目录（\*由平台安装路径决定，默认为C盘）。

在该例程中，目标DLL模型文件（即需要CopterSim重新调用的DLL模型）为HexarotorModelCTRL.dll。

Config.json	2024/2/5 15:56	JSON 源文件	2 KB
HexarotorModelCTRL.dll	2024/2/5 15:56	应用程序扩展	226 KB
MulticopterNoCtrl.dll	2024/2/5 15:56	应用程序扩展	226 KB
MulticopterNoCtrl_SITLRun.bat	2024/5/28 12:21	Windows 批处理...	6 KB
Readme.pdf	2024/5/27 13:06	Foxit PhantomP...	706 KB
ReqCopterSim.py	2024/2/5 15:56	Python 源文件	12 KB
testReqCopterSim.py	2024/6/12 10:21	Python 源文件	3 KB
__pycache__	2024/6/12 10:01	文件夹	

在实验开始前，将其复制到 “[安装目录]\CopterSim\external\model” 文件目录下，这样即可通过sendReSimDllName()将CopterSim当前调用的DLL模型修改为目标DLL模型。

Windows (C:) > PX4PSP > CopterSim > external > model 在 model 中搜

名称	修改日期	类型	大小
HexarotorModelCTRL.dll	2024/2/5 15:56	应用程序扩展	226 KB
MulticopterNoCtrl.dll	2024/2/5 15:56	应用程序扩展	226 KB

## Step 2

启动四旋翼的软件在环仿真并创建一个飞机。

__pycache__	2024/3/18 15:53	文件夹	
Config.json	2024/2/5 12:14	JSON 源文件	2 KB
HexarotorModelCTRL.dll	2023/11/15 10:06	应用程序扩展	226 KB
MulticopterNoCtrl.dll	2024/2/5 12:14	应用程序扩展	226 KB
MulticopterNoCtrl_SITLRun.bat	2024/3/11 12:40	Windows 批处理...	6 KB
PX4MavCtrlV4.py	2024/2/5 12:14	Python 源文件	103 KB
Readme.pdf	2024/3/8 12:09	Foxit PhantomP...	654 KB
ReqCopterSim.py	2024/2/5 12:14	Python 源文件	12 KB
testReqCopterSim.py	2024/3/18 16:35	Python 源文件	3 KB

```
C:\Windows\system32\cmd.e: X + v
已复制      1 个文件。
-----
Please input UAV swarm number:1|
```

## Step 3

等待载具生成后运行“[testReqCopterSim.py](#)”程序。



打开例程目录中的“Python38Run.bat”，输入“python3 testReqCopterSim.py”并按回车。

手动启动python38环境，见：

[\[安装目录\]\RflySimAPIs\2.RflySimUsage\0.ApiExps\e1\\_RflySimSoftwareReadme\Python38\Readme.pdf](#)

。

使用vscode和pycharm等编辑器运行python文件，见：

[\[安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e3.PythonConfig\Readme.pdf](#)

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

E:\summerwork\work1\2.RfLySimUsage\0.ApiExps\e3_ReqCopterSim>python3 testReqCopterSim.py
```

## Step 4

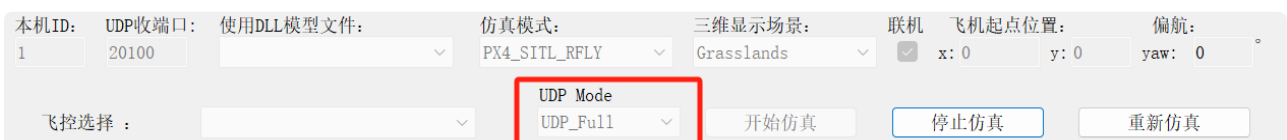
可以看到该程序获取了CopterSim所在电脑的IP，并进行配置。

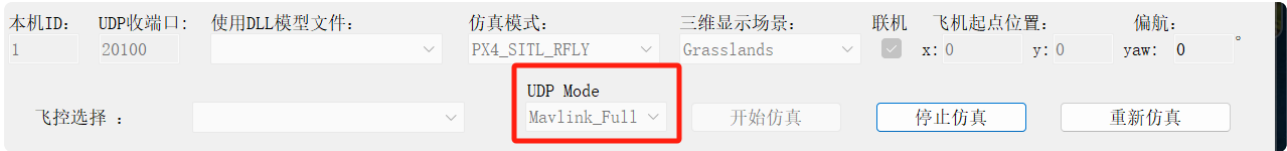
```
HostIP is 192.168.47.1
Start listening CopterSim heartbeat Msg ...
End listening CopterSim heartbeat.
Got 1 CopterSim on the LAN.
[(1, '192.168.1.100')]
192.168.1.100
发送联机模式重设命令
```

## Step 5

运行至“sendReSimUdpMode”时可以在CopterSim上观察到UDP\_Mode由UDP Full转换为了MAVLink Full。

```
# 通过本接口，可以强制修改CopterSim的UDP_Mode，这里只有要时，才发送 new_udp_mode=2 # 注，SITLRun模式是0模式，也就是UDP_Full，这里我们强制修改为2模式 req.sendReSimUdpMode(CopterID,new_udp_mode) # 强制申请CopterSIM转换为MAVLink_Full通信模式 print('发送UDP仿真模式重设命令') time.sleep(5)
```

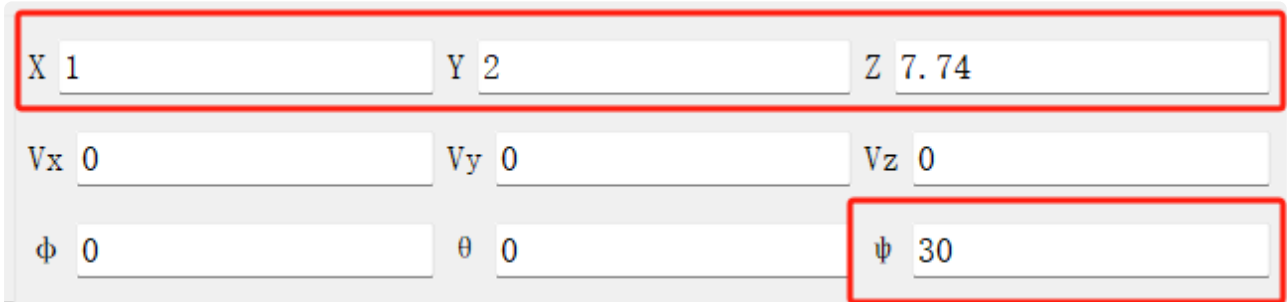
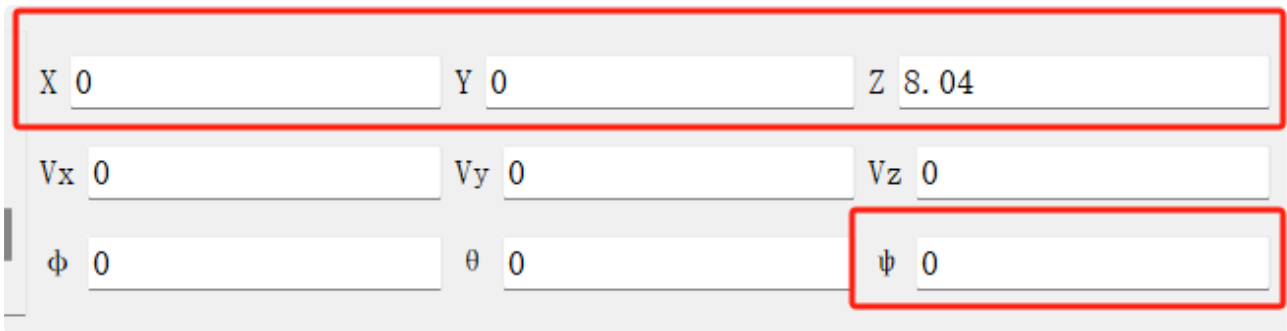




## Step 6

运行至“sendReSimXYyaw”时可以在CopterSim看到载具的x y 位置和yaw的初始值的改变，同时也可以看到在UE界面观察到载具的突变。

```
req.sendReSimXYyaw(CopterID,[1,2,30]) # 强制申请CopterSIM切换初始位置为x=1m,y=2m,yaw=30degree print('发送XYyaw初始值重设命令')
```



## Step 7

运行至“sendReSimXyzRPYaw”时可以在CopterSim看到载具位置与姿态角的变化，同时也可以看到在UE界面观察到载具的突变，此接口相较于“sendReSimXYyaw”能修改z方向位置与滚转和俯仰角度

```
req.sendReSimXyzRPYaw(CopterID,[4,5,-30],[0,0,45]) # 强制申请CopterSIM切换初始位置为30米高，偏航角为45度 print('发送XyzRPYaw初始值重设命令')
```

X	1	Y	2	Z	7.74
Vx	0	Vy	0	Vz	0
$\phi$	0	$\theta$	0	$\psi$	30

X	4	Y	5	Z	7.501
Vx	0	Vy	0	Vz	0.001
$\phi$	0	$\theta$	0	$\psi$	45

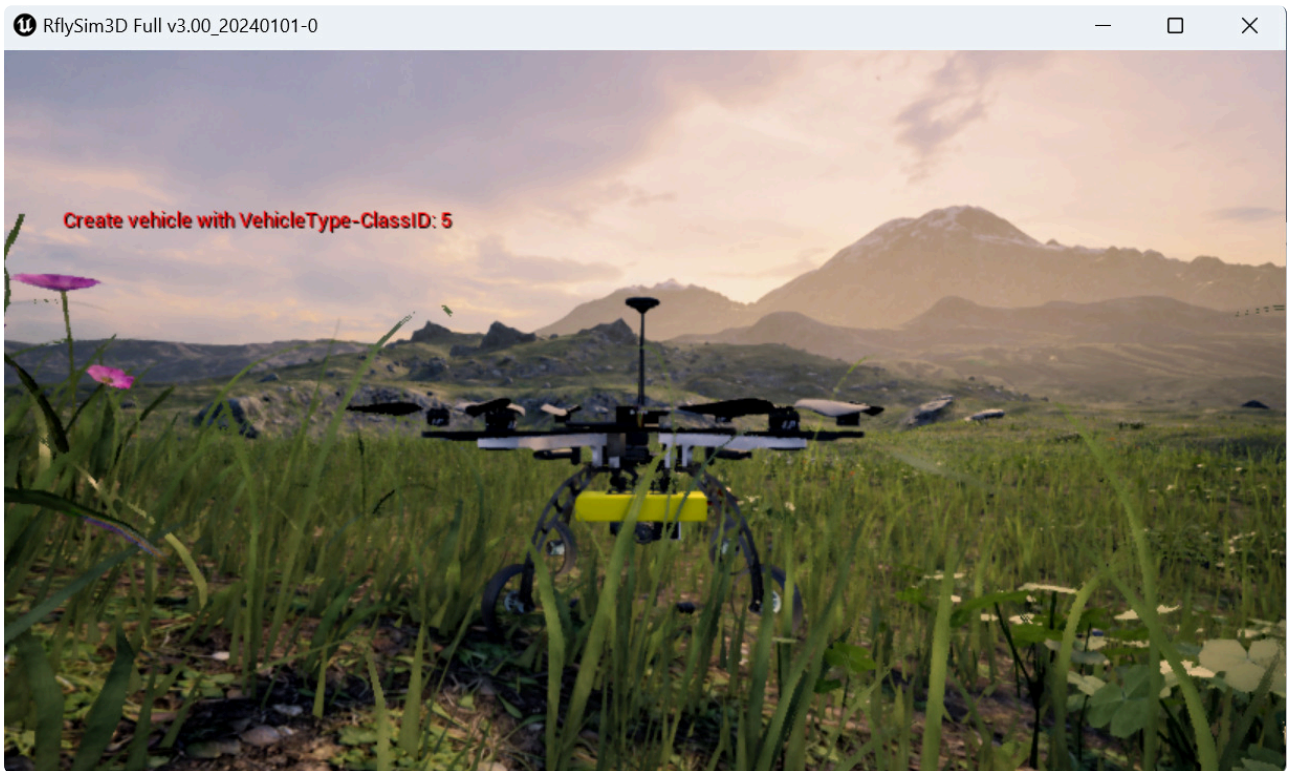
## Step 8

运行至“sendReSimDllName”时可以在CopterSim看到dll模型由四旋翼变为六旋翼，同时也可以从UE界面观察到载具类型由四旋翼修改为了六旋翼（注：在使用该函数前，一定要保证C:\PX4PSP\CopterSim\external\model文件目录下是存在目标DLL模型文件的，否则会导致修改不成功！）。

```
time.sleep(5) req.sendReSimDllName(CopterID,'HexarotorModelCTRL') print('发送DLL模型重设命令')
```

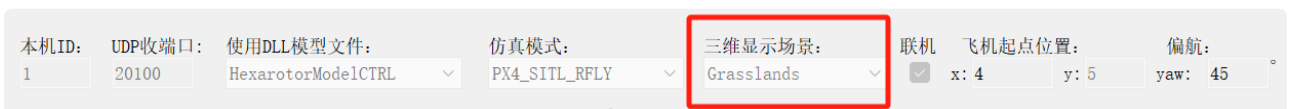
本机ID:	UDP收端口:	使用DLL模型文件:	仿真模式:	三维显示场景:	联机	飞机起点位置:	偏航:
1	20100	MulticopterNoCtrl	PX4_SITL_RFLY	Grasslands	<input checked="" type="checkbox"/>	x: 4 y: 5	yaw: 45°
飞控选择:		UDP Mode	Mavlink_Full	开始仿真	停止仿真	重新仿真	

本机ID:	UDP收端口:	使用DLL模型文件:	仿真模式:	三维显示场景:	联机	飞机起点位置:	偏航:
1	20100	HexarotorModelCTRL	PX4_SITL_RFLY	Grasslands	<input checked="" type="checkbox"/>	x: 4 y: 5	yaw: 45°
飞控选择:		UDP Mode	Mavlink_Full	开始仿真	停止仿真	重新仿真	

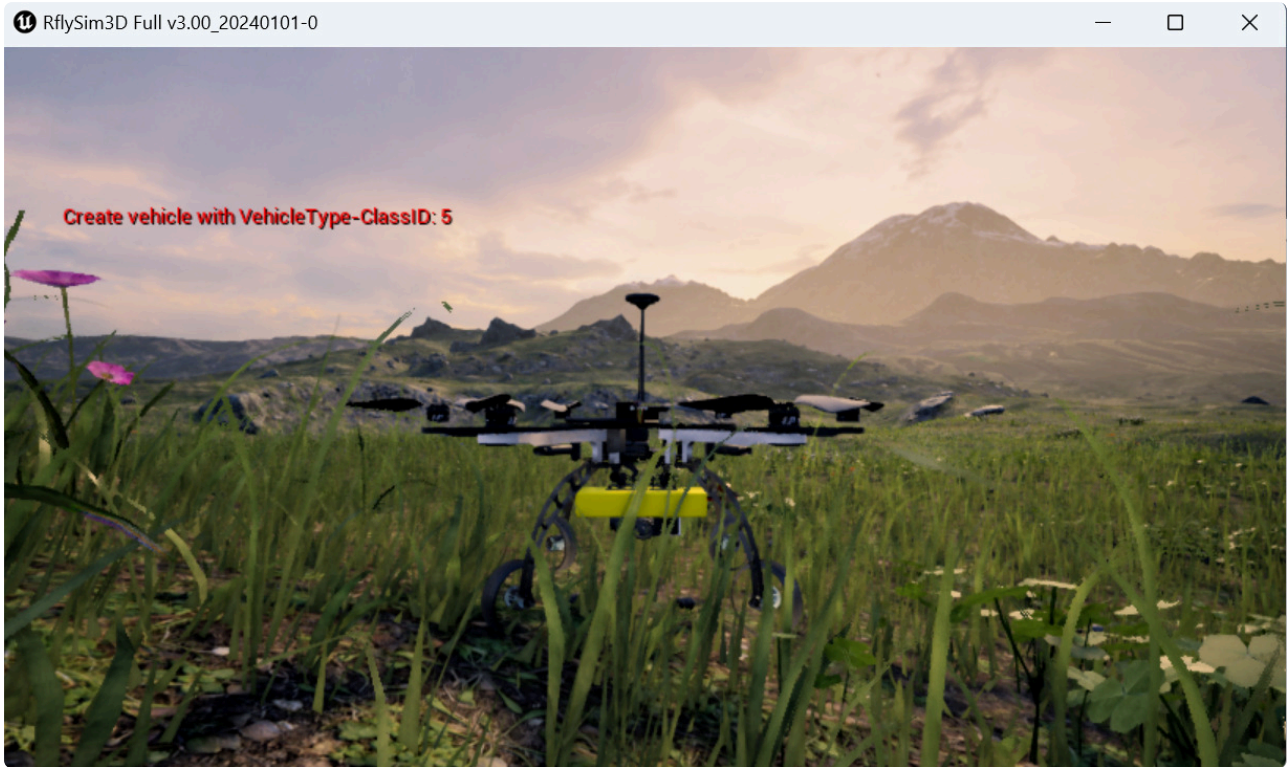


## Step 9

运行至“sendReSimMapName”时可以在CopterSim看到地图由Grasslands变化为VisionRing，同时也可以可以在UE界面观察到地图显示的变化。



本机ID:	UDP收端口:	使用DLL模型文件:	仿真模式:	三维显示场景:	联机	飞机起点位置:	偏航:
1	20100	HexarotorModelCTRL	PX4_SITL_RFLY	VisionRing	<input checked="" type="checkbox"/>	x: 4 y: 5	yaw: 45
飞控选择:			UDP Mode				
			Mavlink_Full				
				开始仿真		停止仿真	重新仿真



## Step 10

完成以上初始化修改操作，Python脚本运行到此处时，能让飞机重新EKF fixed完成，并且进行Offboard控制；用户也可以通过QGC来控制飞机。

```
# 开始控制飞机 print("5s, Arm the drone") mav.initOffboard() time.sleep(0.5)
mav.SendMavArm(True) # Arm the drone print("Arm the drone!, and fly to NE
D 0,0,-5") time.sleep(0.5) mav.SendPosNED(0, 0, -20, 0) # Fly to target positio
n 0,0, -1.5 time.sleep(5)
```

```
问题 输出 终端 调试控制台 端口

发送map地图重设命令
waiting for EKF fixed...
EKF fixed, continue to control
5s, Arm the drone
Arm the drone!, and fly to NED 0,0,-5
PX4 Armed!
```

X	0	Y	0	Z	20.071
Vx	0	Vy	0	Vz	-0.001
$\phi$	0	$\theta$	0	$\psi$	11.736



注：在一台电脑重新启动Step1的软件在环后，可以在另一台电脑运行“[testReqCopterSim.py](#)”程序，程序在获取到进行仿真电脑的IP后，可以远程实现Step3~9

现象。

## 6.参考资料

无。

## 7.常见问题

Q1: \*\*\*

A1: \*\*\*