三维场景交互接口场景控制python接口验证实验

1. 实验目的

了解在仿真时，如何通过python接口控制RflySim3D，实现包括发送命令、更新无人机状态、附加无人机等操作。

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\3.RflySim3DUE\0.ApiExps\e6_RflySim3DCtrlAPI\1.UECtrlPy

- PythonCMDDemo.py：此文件调用了"UE4CtrlAPI.py"中的接口
- PythonSendUE4Attatch.py：此文件调用了"UE4CtrlAPI.py"中的接口
- PythonSendUE4ExtDemo.py：此文件调用了"UE4CtrlAPI.py"中的接口
- PythonOthers.py：此文件调用了"UE4CtrlAPI.py"中的接口
- Python38Run.bat：Python环境启动脚本
- Ue4.bat：打开RflySim3D
- PythonSendUE4Pos.py：此文件调用了"UE4CtrlAPI.py"中的接口
- WestTransportC130J：烘焙好的、能被RflySim3D识别自定义的飞机模型

4. 实验内容或步骤

Python API实验（必做）

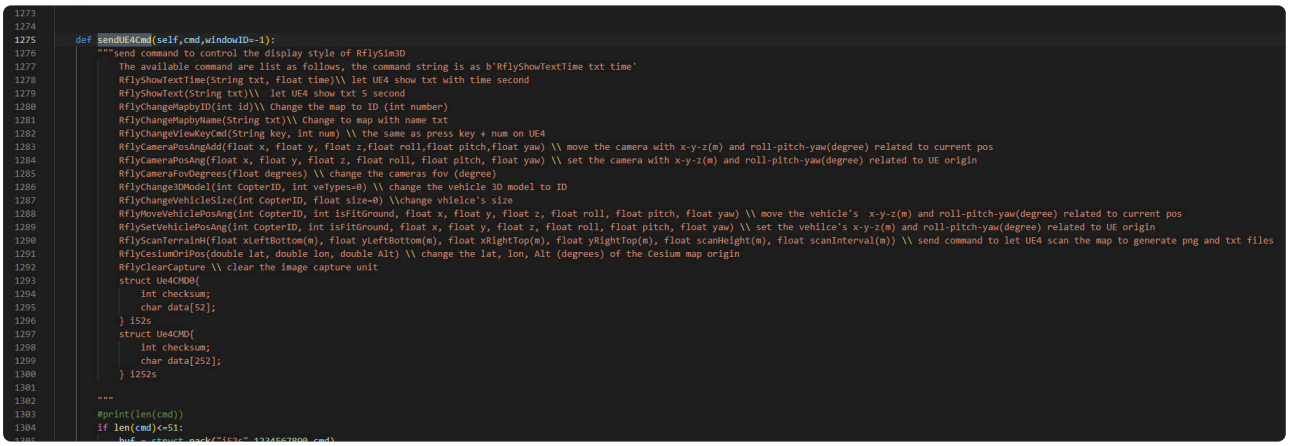
4.1 步骤1：下载附加文件

为了保证RflySim平台安装包的大小，本实验中所用到的三维场景、飞机模型等较大文件均已上传至百度网盘中，请在实验前进行下载，下载链接为：
https://pan.baidu.com/s/1O98i2oQmsE81pBRJQr_46A

提取码：z85c。下载完成后，进行解压放入本例程文件夹中。注：请勿修改文件夹名称。

4.2 步骤2：在库文件UE4CtrlAPI.py中找到对应接口函数定义和用法

找到"UE4CtrlAPI.py"文件，其中定义了大量RflySim平台内置的python接口函数，在其中的UE4CtrlAPI类下可以找到与RflySim3D三维场景内物体控制相关的接口，与RflySim3D场景内物体控制相关的函数都带着前缀"sendUE4"。



```
def sendUE4Cmd(self,cmd,windowID=-1):
    """send command to control the display style of RflySim3D
        The available command are list as follows, the command string is as b'RflyShowTextTime txt time'
        RflyShowTextTime(String txt, float time)\\ let UE4 show txt with time second
        RflyShowText(String txt)\\  let UE4 show txt 5 second
        RflyChangeMapbyID(int id)\\ Change the map to ID (int number)
        RflyChangeMapbyName(String txt)\\ Change to map with name txt
        RflyChangeViewKeyCmd(String key, int num) \\ the same as press key + num on UE4
        RflyCameraPosAngAdd(float x, float y, float z,float roll,float pitch,float yaw) \\ move the camera with x-y-z(m) and roll-pitch-yaw(degree) related to current pos
        RflyCameraPosAng(float x, float y, float z, float roll, float pitch, float yaw) \\ set the camera with x-y-z(m) and roll-pitch-yaw(degree) related to UE origin
        RflyCameraFovDegrees(float degrees) \\ change the cameras fov (degree)
        RflyChange3DModel(int CopterID, int veTypes=0) \\ change the vehicle 3D model to ID
        RflyChangeVehicleSize(int CopterID, float size=0) \\change vhielce's size
        RflyMoveVehiclePosAng(int CopterID, int isFitGround, float x, float y, float z, float roll, float pitch, float yaw) \\ move the vehicle's  x-y-z(m) and roll-pitch-yaw(degree) related to current pos
        RflySetVehiclePosAng(int CopterID, int isFitGround, float x, float y, float z, float roll, float pitch, float yaw) \\ set the vehilce's x-y-z(m) and roll-pitch-yaw(degree) related to UE origin
        RflyScanTerrainH(float xLeftBottom(m), float yLeftBottom(m), float xRightTop(m), float yRightTop(m), float scanHeight(m), float scanInterval(m)) \\ send command to let UE4 scan the map to generate png and txt files
        RflyCesiumOriPos(double lat, double lon, double Alt) \\ change the lat, lon, Alt (degrees) of the Cesium map origin
        RflyClearCapture \\ clear the image capture unit
        struct Ue4CMD0{
            int checksum;
            char data[52];
        } i52s
        struct Ue4CMD{
            int checksum;
            char data[252];
        } i252s

    """
    #print(len(cmd))
    if len(cmd)<=51:
        buf = struct.pack("i52s" 1234567890 cmd)
```

该函数是与RflySim3D相关的最重要的函数，从它的大量注释可以看到我们之前介绍的RflySim3D命令接口基本都在上面。

```python
    def sendUE4ExtAct(self,copterID=1,ActExt=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],windowID=-1):
        # struct Ue4ExtMsg {
        #     int checksum;//1234567894
        #     int CopterID;
        #     double runnedTime; //Current  stamp (s)
        #     double ExtToUE4[16];
        # }
        #struct.pack 2i1d16f
        runnedTime = time.time()-self.startTime
        checkSum=1234567894
        buf = struct.pack("2i1d16d",checkSum,copterID,runnedTime,*ActExt)
        if windowID<0:
            if self.ip=='127.0.0.1':
                for i in range(6):
                    self.udp_socket.sendto(buf, (self.ip, 20010+i))
            else:
                self.udp_socket.sendto(buf, ('224.0.0.10', 20009)) #multicast address, send to all RflySim3Ds on all PC in LAN
        else:
            if self.ip!='127.0.0.1' and self.ip!='255.255.255.255':
                self.udp_socket.sendto(buf, ('127.0.0.1', 20010+windowID)) #ensure this PC can reciver message under specify IP mode
            self.udp_socket.sendto(buf, (self.ip, 20010+windowID)) #specify PC's IP to send
        #print('Message Send')
```

该函数可以调用目标无人机的蓝图接口"ActuatorInputsExt"，要求完整版RflySim3D，有16维参数。

```python
    def sendUE4PosNew(self,copterID=1,vehicleType=3,PosE=[0,0,0],AngEuler=[0,0,0],VelE=[0,0,0],PWMs=[0]*8,runnedTime=-1,windowID=-1):
        """ send the position & angle information to RflySim3D to create a new 3D model or update the old model's states
            # //输出到模拟器的数据
            # struct SOut2SimulatorSimpleTime {
            #     int checkSum; //1234567890
            #     int copterID;  //Vehicle ID
            #     int vehicleType;  //Vehicle type
            #     float PWMs[8];
            #     float VelE[3];
            #     float AngEuler[3];  //Vehicle Euler angle roll pitch yaw (rad) in x y z
            #     double PosE[3];   //NED vehicle position in earth frame (m)
            #     double runnedTime; //Current Time stamp (s)
            # };
            #struct.pack 3i14f4d
        """
        # if runnedTime<0:
        #     runnedTime = time.time()-self.startTime
        checkSum=1234567890
        # pack for SOut2SimulatorSimpleTime
        buf = struct.pack("3i14f4d",checkSum,copterID,vehicleType,*PWMs,*VelE,*AngEuler,*PosE,runnedTime)
        #print(len(buf))
        if windowID<0:
            if self.ip=='127.0.0.1':
                for i in range(6):
                    self.udp_socket.sendto(buf, (self.ip, 20010+i))
            else:
                self.udp_socket.sendto(buf, ('224.0.0.10', 20009)) #multicast address, send to all RflySim3Ds on all PC in LAN
        else:
            if self.ip!='127.0.0.1' and self.ip!='255.255.255.255':
                self.udp_socket.sendto(buf, ('127.0.0.1', 20010+windowID)) #ensure this PC can reciver message under specify IP mode
            self.udp_socket.sendto(buf, (self.ip, 20010+windowID)) #specify PC's IP to send
        #print('Message Send')
```

它会发送一次无人机的数据，包括无人机的ID、ClassID、位置、姿态、速度、各电机数据等。当RflySim3D收到这样的数据后，会检测无人机的ID，如果发现该ID不存在于场景中，则会创建一个该飞机，如果已经存在该ID的飞机，则它会更新该无人机的各项数据。

```python
    def sendUE4Attatch(self,CopterIDs,AttatchIDs,AttatchTypes,windowID=-1):
        """ Send msg to UE4 to attach a vehicle to another (25 vehicles);
        CopterIDs,AttatchIDs,AttatchTypes can be a list with max len 25
        """
        # change the 1D variable to 1D list
        if isinstance(CopterIDs,int):
            CopterIDs=[CopterIDs]

        if isinstance(AttatchIDs,int):
            AttatchIDs=[AttatchIDs]

        if isinstance(AttatchTypes,int):
            AttatchTypes=[AttatchTypes]

        if not isinstance(CopterIDs,list) or not isinstance(AttatchIDs,list) or not isinstance(AttatchTypes,list):
            print('Error: Wrong sendUE4Attatch input Type');
            return

        if len(CopterIDs)!=len(AttatchIDs) or len(CopterIDs)!=len(AttatchTypes) or len(CopterIDs)>25:
            print('Error: Wrong sendUE4Attatch input dimension');
            return

        vLen=len(CopterIDs)
        if vLen<25: # Extend the IDs to 25D
            CopterIDs = CopterIDs + [0]*(25-vLen)
            AttatchIDs = AttatchIDs + [0]*(25-vLen)
            AttatchTypes = AttatchTypes + [0]*(25-vLen)

        if vLen>25:
            CopterIDs=CopterIDs[0:25]
            AttatchIDs=AttatchIDs[0:25]
            AttatchTypes=AttatchTypes[0:25]

        # struct VehicleAttatch25 {
        #   int checksum;//1234567892
        #   int CopterIDs[25];
        #   int AttatchIDs[25];
        #   int AttatchTypes[25];//0: 正常模式, 1: 相对位置不相对姿态, 2: 相对位置+偏航（不相对俯仰和滚转）, 3: 相对位置+全姿态（俯仰滚转偏航）
        # }i25i25i25i

        buf = struct.pack("i25i25i25i",1234567892,*CopterIDs,*AttatchIDs,*AttatchTypes)
        if windowID<0:
```

它的作用是将一组无人机附加在另一组无人机上。

附加有几种模式：0：正常模式，1：相对位置不相对姿态，2：相对位置+偏航（不相对俯仰和滚转），3：相对位置+全姿态（俯仰滚转偏航）

4.3 步骤3：sendUE4Cmd函数（发送控制台命令）

该函数的逻辑其实就是把一个字符串"cmd"发送给RflySim3D，如果该cmd是正确的"命令接口函数"则会直接调用RflySim3D的对应函数，产生的效果与RflySim3D控制台命令完全一样。

打开该文档目录下的"PythonCMDDemo.py"文件，它非常简单，只有3句代码。

```python
import UE4CtrlAPI as UE4CtrlAPI

ue = UE4CtrlAPI.UE4CtrlAPI()

ue.sendUE4Cmd(b'RflyChangeMapbyName Grasslands')
```

第一句就是将UE4CtrlAPI.py文件的UE4CtrlAPI模块引入当前py程序；第二句代码表示创建了一个对象，在库文件中已包含了RflySim平台的通信的端口，这两句基本是RflySim3D的python脚本所必备的，不必太关心。第三句就是给RflySim3D发送了一个字符串"RflyChangeMapbyName Grasslands"，在RflySim3D控制台命令中我们也介绍过了，

该字符串表示调用了接口RflyChangeMapbyName(String txt)，它的作用是将地图切换为"Grasslands"。

双击ue.bat打开RflySim3D，然后运行文档目录下"PythonCMDDemo.py"。

在文件夹下，双击Python38Run.bat，打开集成好的python环境，输入 python PythonCMDDemo.py，回车运行可以看到地图被切换了：

现在可以自由尝试RflySim3D控制台命令中介绍的所有命令，或者把这些命令组合起来一同发送。

该函数用UDP发送的结构体形式如下：

struct Ue4CMD0{

int checksum;

char data[52];

} i52s

struct Ue4CMD{

int checksum;

char data[252];

} i252s

其中char数组保存的就是发送的命令。

4.4 步骤4：sendUE4ExtAct函数（触发扩展蓝图接口）

之前使用RflySim3D控制台命令接口输入命令，发送信号告诉飞机产生爆炸效果的实验，我们在这里可以使用python的方式再做一次：

将此目录下的"WestTransportC130J"文件夹复制到"\PX4PSP\RflySim3D\RflySim3D\Content"目录下，它是一个烘焙好的、能被RflySim3D识别自定义的飞机模型。然后打开RflySim3D，鼠标双击地面，快速按下字母O+数字202，即可在地面创建出该飞机。

然后我们再打开此目录下的"PythonSendUE4ExtDemo.py"文件运行，我们可以发现它与直接输入RflySim3D控制台命令的效果是一样的。

4.5 步骤5：sendUE4PosNew函数（创建模型并传入8位电机数据）

打开RflySim3D，再启动并运行此目录下的"PythonSendUE4Pos.py"，我们可以看见一个飞机被创建出来了：

该文件也只有三行：

```
import UE4CtrlAPI as UE4CtrlAPI

ue = UE4CtrlAPI.UE4CtrlAPI()

ue.sendUE4PosNew(10,3,[0,0,-10],[0,0,0],[0,0,0],[0,0,0,0,0,0,0,0])
```

第三句的意思是发送了一个ID为10的飞机的数据，它的ClassID为3，表示我们创建的是一个四旋翼无人机，[0,0,-10]表示它距离水平面高10米。

该函数用UDP发送的结构体形式如下：

struct SOut2SimulatorSimpleTime {

int checkSum; //1234567890

int copterID; //Vehicle ID

int vehicleType; //Vehicle type

float PWMs[8];

float VelE[3];

float AngEuler[3]; //Vehicle Euler angle roll pitch yaw (rad) in x y z

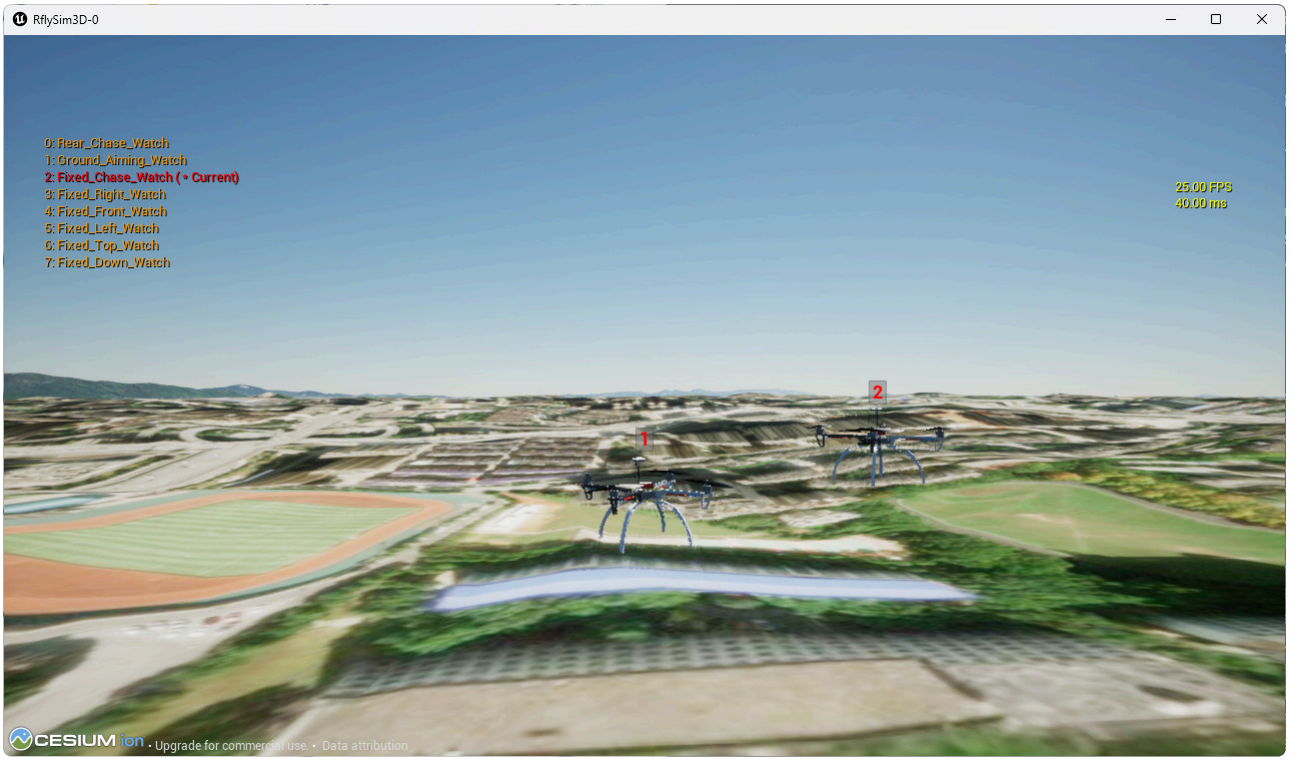
double PosE[3]; //NED vehicle position in earth frame (m)

double runnedTime; //Current Time stamp (s)

};

4.6 步骤6：sendUE4Attatch（将一个Copter附加到另一个Copter上）

打开RflySim3D，再启动并运行目录下的"PythonSendUE4Attatch.py"，我们可以看见创建了两个飞机，1号飞机在前进，并且2号飞机在围绕着1号飞机旋转并随之移动。



4.7 步骤7：其他函数

还有许多函数，但功能大同小异，或者曾介绍过类似的功能，这里就只简单介绍它们的功能了，可以将例程稍加修改实验它们的作用。

运行PythonOther.py以查看以下其他函数的效果。

1. sendUE4LabelID(self,CopterID=0,Txt='',fontSize=30,RGB=[255,0,0],windowID=-1)

它与之前介绍的"RflySetIDLabel"命令作用一致，在目标无人机头上显示一个字符串。

2. sendUE4LabelMsg(self,CopterID=0,Txt='',fontSize=30,RGB=[255,0,0],dispTime=0,dispFlag=-1,windowID=-1)

它与之前介绍的"RflySetMsgLabel"命令作用一致，在目标无人机头顶的ID标签下方再显示消息标签。

3. sendUE4Pos(self,copterID=1,vehicleType=3,MotorRPMSMean=0,PosE=[0,0,0],AngEuler=[0,0,0],windowID=-1)

该函数是sendUE4PosNew函数的简化版本，但作用是一样的。

4. sendUE4Pos2Ground(self,copterID=1,vehicleType=3,MotorRPMSMean=0,PosE=[0,0,0],AngEuler=[0,0,0],windowID=-1)

该函数与sendUE4PosNew相似，但由它生成的无人机的z坐标不起作用，其z坐标会贴合当前的地面。

5. sendUE4PosScale(self,copterID=1,vehicleType=3,MotorRPMSMean=0,PosE=[0,0,0],AngEuler=[0,0,0],Scale=[1,1,1],windowID=-1)

该函数作用与sendUE4PosNew也相似，也是发送无人机的数据，只是更新的数据有所不同，它额外发送了一个缩放数据Scale，可以控制无人机的缩放大小。

6. sendUE4PosScale2Ground(self,copterID=1,vehicleType=3,MotorRPMSMean=0,PosE=[0,0,0],AngEuler=[0,0,0],Scale=[1,1,1],windowID=-1)

该函数与sendUE4Pos2Ground函数相似，也是发送的数据稍有不同，它也发送了一个无人机的缩放数据Scale

7. sendUE4PosFull(self,copterID,vehicleType,MotorRPMS,VelE,PosE,RateB,AngEuler,windowID=-1)

该函数也是发送无人机的数据，更新的数据稍有不同，附加了无人机的速度、八个PWM值等信息。

8. sendUE4PosSimple(self,copterID,vehicleType,PWMs,VelE,PosE,AngEuler,runnedTime=-1,windowID=-1)

该函数也是发送无人机的数据，与sendUE4PosNew很相似。

9. sendUE4PosScale100(self,copterID,vehicleType,PosE,AngEuler,MotorRPMSMean,Scale,isFitGround=False,windowID=-1)

该函数可以一次发送100架无人机的数据，将它们放在同一个结构体里，增加大规模集群仿真时的信息密度。该函数不可传输带有style和class的车辆种类数据。

10. sendUE4PosScalePwm20(self,copterID,vehicleType,PosE,AngEuler,Scale,PWMs,isFitGround=False,windowID=-1):

该函数可以一次发送20架无人机的数据。该函数不可传输带有style和class的车辆种类数据。

4.8 步骤8：Vscode调试运行实验（选做）

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，在运行python文件时，可使用VS Code（或Pycharm等工具）来打开python文件文件，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阅读例程中的python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。
- 请尝试修改代码，尝试API不同的参数。

5. 关键知识点

关键知识点1：sendUE4Cmd函数

用于发送控制台命令到RflySim3D，将字符串命令发送给RflySim3D执行相应功能。

关键知识点2：sendUE4PosNew函数

用于创建或更新无人机状态，可以发送无人机的位置、姿态、速度等数据。

关键知识点3：sendUE4Attatch函数

用于将一个无人机附加到另一个无人机上，支持多种附加模式。

6.参考资料

1. RflySim3D快捷键接口总览
2. RflySim3D控制台命令接口总览
3. UE场景控制python接口总览
4. RflySim3D外部交互接口数据协议

5. RflySim3D模型导入及蓝图控制规则

7.常见问题

Q1：如何确保RflySim3D与Python脚本通信正常?

A1：确保RflySim3D正在运行，检查网络连接和端口是否正确配置，确认UE4CtrlAPI.py库文件中的端口设置与RflySim3D的接收端口一致。

Q2：Python脚本运行后RflySim3D没有反应怎么办?

A2：首先确认RflySim3D是否已经启动并处于运行状态；其次检查Python脚本中的参数设置是否正确；最后检查是否有防火墙阻止了程序间的通信。

Q3：如何同时控制多架无人机?

A3：可以通过sendUE4PosScale100或sendUE4PosScalePwm20等函数批量发送多架无人机的数据，也可以通过循环调用单机控制函数实现多机控制。

---

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