

点对点无人机数据转发实验

1. 实验目的

本实验旨在演示如何使用RflySim平台实现多无人机系统的点对点通信与数据转发功能。通过本实验，学习者将掌握以下技能：

- 理解RflySim平台的分布式仿真机制
- 学会在局域网环境下配置多台计算机协同工作的无人机仿真系统
- 掌握无人机之间的数据转发技术
- 了解如何通过Python API实现无人机间的通信
- 学会使用自动IP检测和手动IP指定两种连接方式

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]，任意系统电脑2台（可是虚拟机）

3. 实验地址

例程目录：

[\[安装目录\]\RflySimAPIs\9.RflySimComm\0.ApiExps\e1.RflyNetAPIExps\1.P2PSendData](#)

[Python38Run.bat](#)：RflySim Python运行脚本

[SITLRun2MavlinkFull.bat](#)：启动2架无人机SITL仿真环境脚本

[UAV1Ctrl.py](#)：控制第1架无人机的Python脚本

[UAV2Ctrl.py](#)：控制第2架无人机的Python脚本

[SetIp/](#)：指定IP连接方式示例目录

[SetIp/SITLRun2MavlinkFull.bat](#)：修改版SITL启动脚本

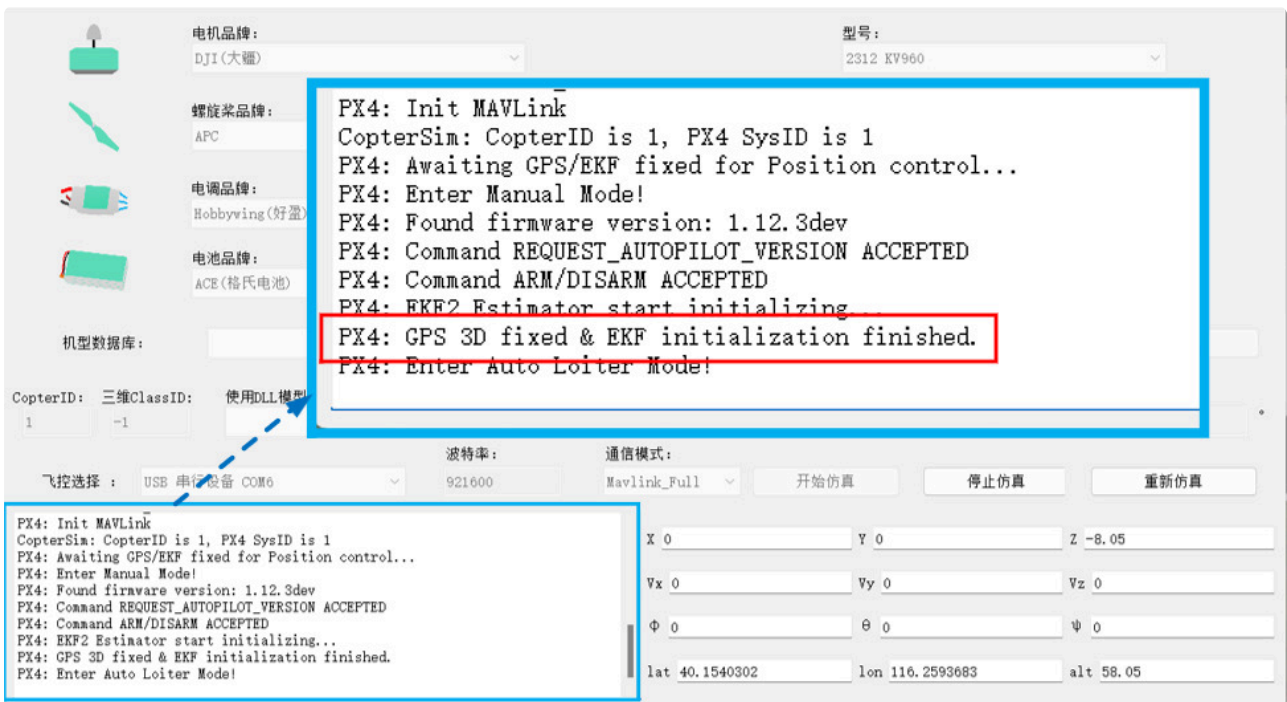
[SetIp/UAV1Ctrl.py](#)：指定IP的UAV1控制脚本

[SetIp/UAV2Ctrl.py](#)：指定IP的UAV2控制脚本

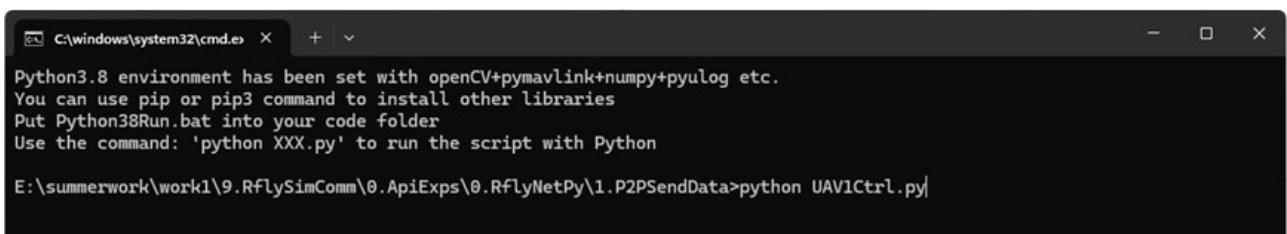
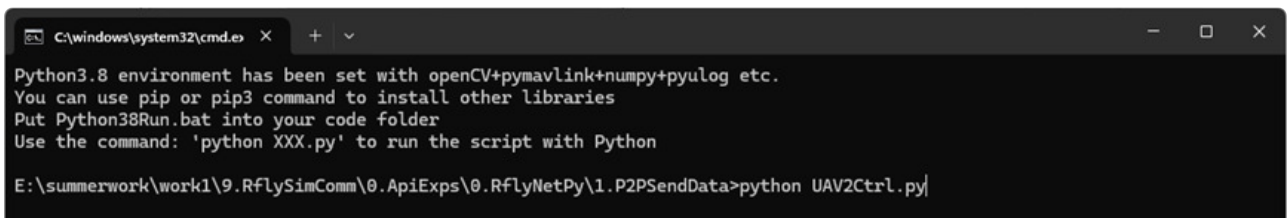
4. 实验内容或步骤

4.1 步骤1：本机仿真测试

双击运行 `SITLRun2MavlinkFull.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，2 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：



然后，双击运行"Python38Run.bat"，在其中输入"python UAV2Ctrl.py"，启动2号飞机监听程序；再双击运行"Python38Run.bat"，在其中输入"python UAV1Ctrl.py"，启动1号飞机发送程序



UAV2Ctrl.py 对应的CMD框将打印出UAV1的数据后，退出程序。

```
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\9.RflySimComm\0.ApiExps\0.RflyNetPy\1.P2PSendData>python UAV2Ctrl.py
HostIP is 192.168.31.100
Start listening CopterSim heartbeat Msg ...
Got time msg from CopterSim # 1 , not on this PC
Got time msg from CopterSim # 2 , running on this PC
Got time msg from CopterSim # 1 , running on this PC
End listening CopterSim heartbeat.
Got 2 CopterSim on the LAN.
Got UAV1 Data
UAV1 Data: 1 (0.00014527284656651318, -5.953012987447437e-06, -0.0019272005883976817) (0.004201848059892654, 0.00660126
3303309679, 0.0037166723050177097) (0.00014527284656651318, -5.953012987447437e-06, -0.0019272005883976817) (0.064770206
80904388, 0.06230045482516289, -0.005292274057865143) (0.04256248566940357, 0.03673971810547405, -7.975292272603776)
|
```

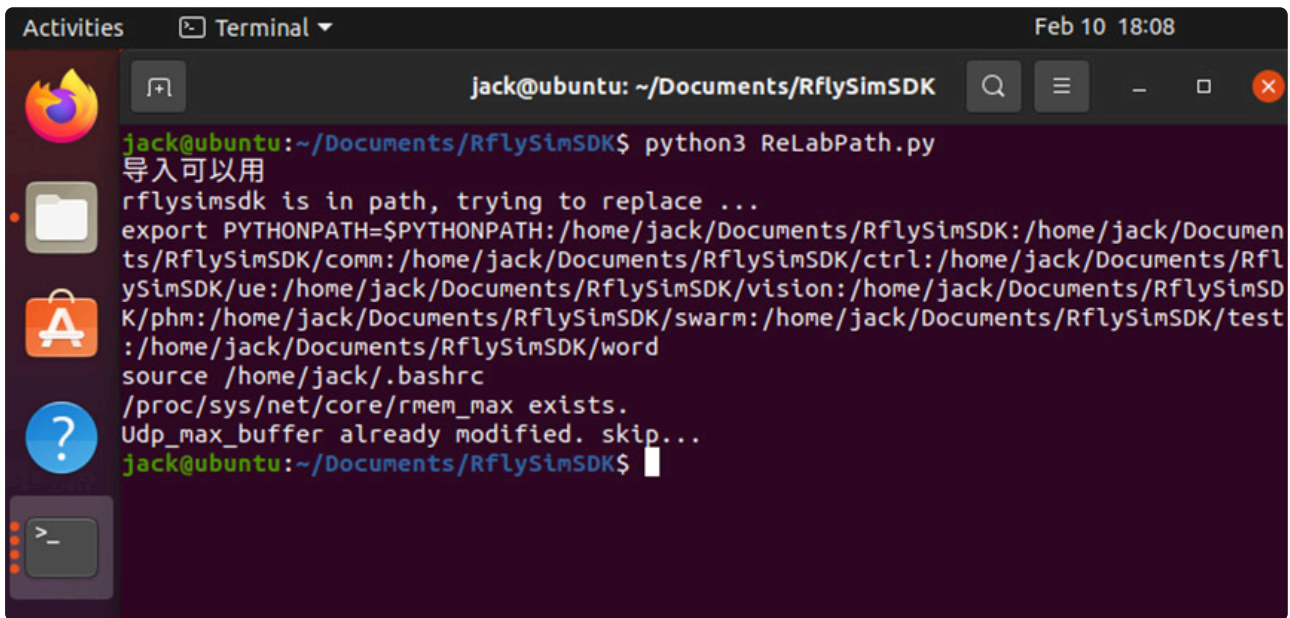
```
C:\windows\system32\cmd.exe x + v
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

E:\summerwork\work1\9.RflySimComm\0.ApiExps\0.RflyNetPy\1.P2PSendData>python UAV1Ctrl.py
HostIP is 192.168.31.100
Start listening CopterSim heartbeat Msg ...
Got time msg from CopterSim # 1 , running on this PC
Got time msg from CopterSim # 2 , running on this PC
End listening CopterSim heartbeat.
Got 2 CopterSim on the LAN.
Start sending UAV1 data
|
```

4.2 步骤2：联机测试

4.2.1 准备工作

- 1) 准备2台任意系统主机（可以是NX主机、Ubuntu虚拟机、Windows电脑）。确保两台机器是在同一个局域网内。
- 2) 按照[RflySim按安装理解]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e3.PythonConfig实验将[RflySim安装路径]\RflySimAPIs\RflySimSDK\RflySimSDK库复制到每台主机上，并确保RflySimSDK已经导入当前主机的Python环境中。



```
jack@ubuntu: ~/Documents/RflySimSDK
jack@ubuntu:~/Documents/RflySimSDK$ python3 ReLabPath.py
导入可以用
rflysimSDK is in path, trying to replace ...
export PYTHONPATH=$PYTHONPATH:/home/jack/Documents/RflySimSDK:/home/jack/Documents/RflySimSDK/comm:/home/jack/Documents/RflySimSDK/ctrl:/home/jack/Documents/RflySimSDK/ue:/home/jack/Documents/RflySimSDK/vision:/home/jack/Documents/RflySimSDK/phm:/home/jack/Documents/RflySimSDK/swarm:/home/jack/Documents/RflySimSDK/test:/home/jack/Documents/RflySimSDK/word
source /home/jack/.bashrc
/proc/sys/net/core/rmem_max exists.
Udp_max_buffer already modified. skip...
jack@ubuntu:~/Documents/RflySimSDK$
```

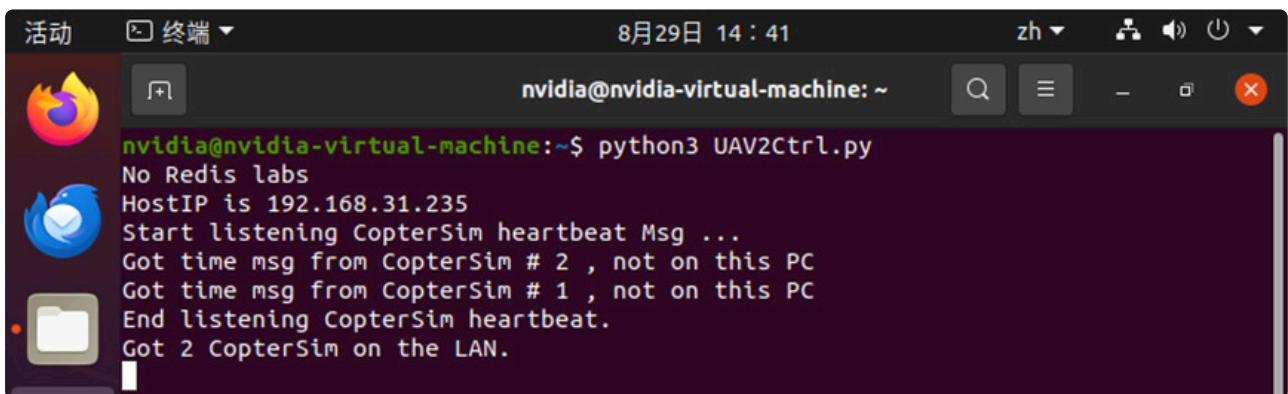
4.2.2 仿真初始化

在Windows主机中双击运行SITLRun2MavlinkFull.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，2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：



4.2.3 运行程序

在任意系统主机中，如：Ubuntu1中运行UAV1Ctrl.py；在Ubuntu2中，运行UAV2Ctrl.py收到并打印UAV1的数据后，退出程序。



能看到和本地测试一样的效果，说明测试正确。

```
nvidia@nvidia-virtual-machine: ~  
nvidia@nvidia-virtual-machine:~$ python3 UAV2Ctrl.py  
No Redis labs  
HostIP is 192.168.31.235  
Start listening CopterSim heartbeat Msg ...  
Got time msg from CopterSim # 2 , not on this PC  
Got time msg from CopterSim # 1 , not on this PC  
End listening CopterSim heartbeat.  
Got 2 CopterSim on the LAN.  
Got UAV1 Data  
UAV1 Data: 1 (0.00010257413669023663, -0.0004003295907750726, -0.0020041947718709707) (0.006854436360299587, 0.0014240576419979334, 0.0100273247808218) (0.00010257413669023663, -0.0004003295907750726, -0.0020041947718709707) (0.06760751456022263, 0.050279513001441956, 0.02328004688024521) (0.04539979342058231, 0.02471877628175312, -7.946719951665665)
```

注：UAV2Ctrl.py和UAV2Ctrl.py前面的 `req = ReqCopterSim.ReqCopterSim()` 库的调用,能够自动获取到局域网中CopterSim的IP地址，因此能够直接运行。

4.3 步骤3：指定IP的联机方式

4.3.1 准备工作

上面的例子，是自动识别IP的方法，也可以手动指定。进入SetIp目录，里面有需要设定IP地址的例程。本次实验中，所这里使用的Windows主机IP为192.168.31.100，两台Ubuntu虚拟机的IP分别为：192.168.31.83和192.168.31.235。

4.3.2 程序修改

修改SITLRun2MavlinkFull.bat的"SET IS_BROADCAST=,"字段，将***分别换成两台虚拟机的IP地址。

```
SITLRun2MavlinkFull.bat
File Edit View
REM Set the interval between two vehicle, unit:m
SET /a VEHICLE_INTERVAL=2

REM Set broadcast to other computer; IS_BROADCAST=0: only this computer, IS_BROADCAST=1: broadcast;
REM or use IP address to increase speed, e.g., IS_BROADCAST=192.168.3.1
REM Note: in IP mode, IS_BROADCAST=0 equals to IS_BROADCAST=127.0.0.1, IS_BROADCAST=1 equals to
IS_BROADCAST=255.255.255.255
REM You can also use a IP list with seperator "," or ";" to specify IPs to send, e.g.,
127.0.0.1,192.168.1.4,192.168.1.5
SET IS_BROADCAST=192.168.31.83,192.168.31.235

REM Set UDP data mode; 0: UDP_FULL, 1:UDP_Simple, 2: Mavlink_Full, 3: Mavlink_simple. input number
or string
REM e.g., UDPSIMMODE=1 equals to UDPSIMMODE=UDP_Simple
SET UDPSIMMODE=Mavlink_Full

:Top
ECHO.
ECHO -----
REM Max vehicle number 50
SET /a MAX_VEHICLE=50

Ln 63, Col 46 | 5,314 characters | 100% | Windows (CRLF) | UTF-8
```

修改 `UAV1Ctrl.py`，将 `TargetIP=` 的改成Windows电脑的IP地址。注意，这里应该是IP的字符串，所以需要引号括起来。

```
UAV1Ctrl.py
File Edit View
import time
import math
import sys
import ReqCopterSim
import PX4MavCtrlV4 as PX4MavCtrl
import NetSimAPIV4

CopterID=1
TargetIP="192.168.31.100" # 请在此处填入Windows电脑的IP地址

## 下面代码开始创建通信实例
#Create a new MAVLink communication instance, UDP sending port (CopterSim's receiving port) is 20100
# 创建#CopterID号飞机的通信实例, 和CopterSim#CopterID号相连, 使用TargetIP确保分布式仿真也能用
mav = PX4MavCtrl.PX4MavCtrl(CopterID,TargetIP)
net = NetSimAPIV4.NetSimAPI(mav)

#mav.InitMavLoop(UDPMode), where UDPMode=0,1,2,3,4
# Use UDP_Simple Mode to control PX4
# In this mode, this script will send struct inHILCMDData to CopterSim
# Then CopterSim convert it to MAVLinkOffboard message to PX4
# In this mode, PX4 send MAVLink data to CopterSim, which convert Struct data to this script
# Obviously, UDP_Simple is faster (little data, low delay) than UDP_Simple

Ln 9, Col 26 | 987 characters | 100% | Windows (CRLF) | UTF-8
```

修改 `UAV2trl.py`，将 `TargetIP=` 的改成Windows电脑的IP地址。注意，这里应该是IP的字符串，所以需要引号括起来。

```
UAV2Ctrl.py
File Edit View

import time
import math
import sys
import ReqCopterSim
import PX4MavCtrlV4 as PX4MavCtrl
import NetSimAPIV4

CopterID=2
TargetIP="192.168.31.100" # 请在此处填入Windows电脑的IP地址

## 下面代码开始创建通信实例
#Create a new MAVLink communication instance, UDP sending port (CopterSim's receiving port) is 20100
# 创建#CopterID号飞机的通信实例, 和CopterSim#CopterID号相连, 使用TargetIP确保分布式仿真也能用
mav = PX4MavCtrl.PX4MavCtrl(CopterID,TargetIP)
net = NetSimAPIV4.NetSimAPI(mav)

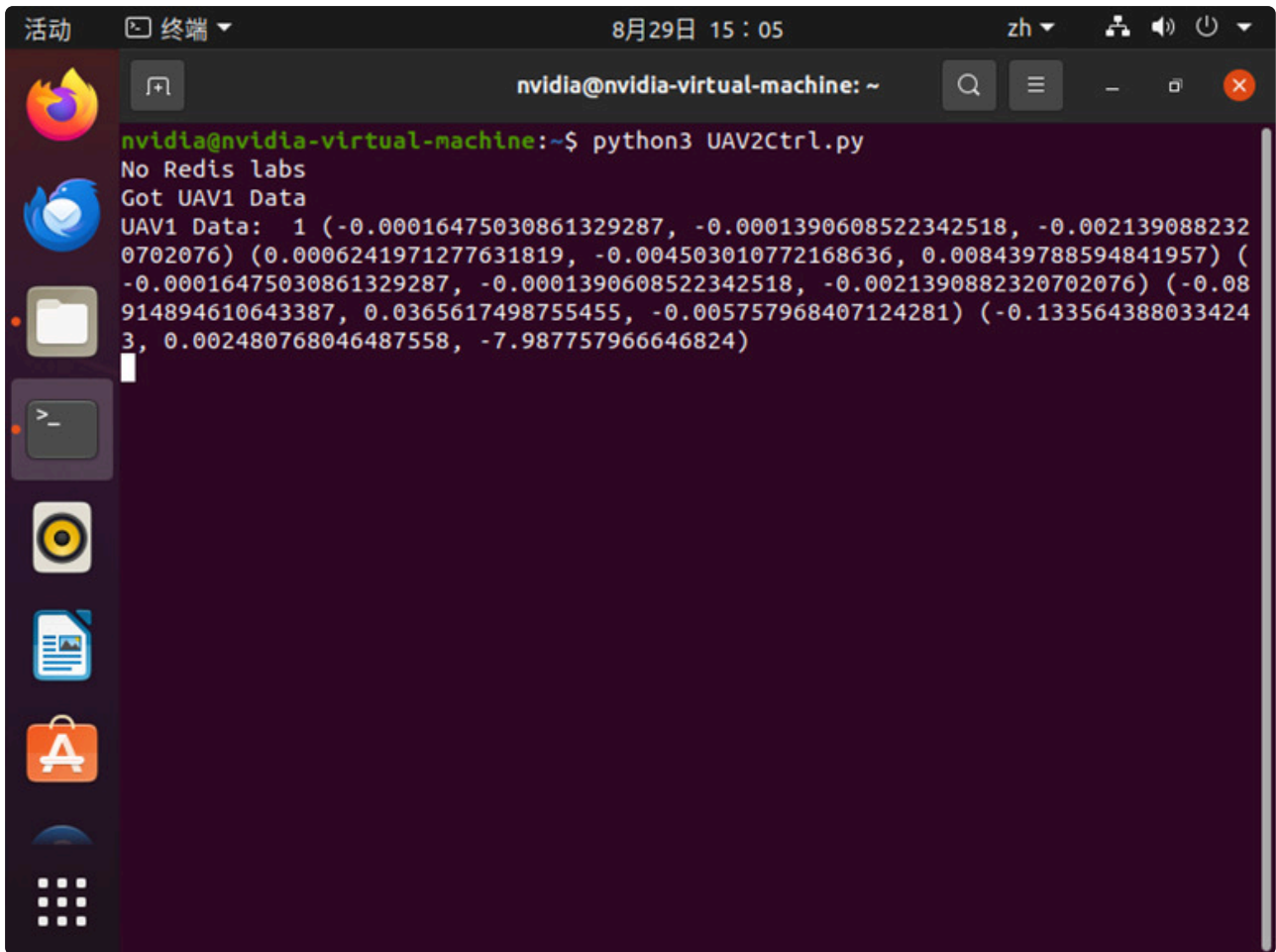
#mav.InitMavLoop(UDPMODE), where UDPMODE=0,1,2,3,4
# Use UDP_Simple Mode to control PX4
# In this mode, this script will send struct inHILCMDData to CopterSim
# Then CopterSim convert it to MAVLinkOffboard message to PX4
# In this mode, PX4 send MAVLink data to CopterSim, which convert Struct data to this script
# Obviously, UDP Simple is faster (little data, low delay) than UDP Simple

Ln 9, Col 26 | 1,277 characters | 100% | Windows (CRLF) | UTF-8
```

4.3.3 仿真运行

修改完成后，按照4.2节的内容进行实验，可以看到同样的效果。

```
活动 终端 8月29日 00:04
rflsysim@rflsysim: ~
rflsysim@rflsysim:~$ python3 UAV1Ctrl.py
Start sending UAV1 data
```



```
nvidia@nvidia-virtual-machine:~$ python3 UAV2Ctrl.py
No Redis labs
Got UAV1 Data
UAV1 Data: 1 (-0.00016475030861329287, -0.0001390608522342518, -0.0021390882320702076) (0.0006241971277631819, -0.004503010772168636, 0.008439788594841957) (-0.00016475030861329287, -0.0001390608522342518, -0.0021390882320702076) (-0.08914894610643387, 0.0365617498755455, -0.005757968407124281) (-0.1335643880334243, 0.002480768046487558, -7.987757966646824)
```

5. 关键知识点

关键知识点：实验整体思路和框架

本实验的核心是实现点对点无人机数据转发功能，主要分为以下几个部分：

1. **仿真环境搭建**: 通过 `SITLRun2MavlinkFull.bat` 脚本启动包含两架无人机的仿真环境
2. **自动IP检测机制**: 使用ReqCopterSim模块自动检测局域网内的无人机仿真器位置
3. **数据转发实现**: 通过NetSimAPIV4实现无人机间的数据转发
4. **分布式仿真**: 支持多台计算机协作的分布式仿真

关键知识点：自动IP检测与手动IP指定

本实验提供了两种连接方式：

1. **自动IP检测:** 使用 `ReqCopterSim.ReqCopterSim()` 库自动获取局域网中CopterSim的IP地址
2. **手动IP指定:** 在SetIp目录下的例程中，通过修改代码中的TargetIP参数，直接指定目标设备的IP地址。需要同时修改SITL启动脚本中的IS_BROADCAST字段以及控制脚本中的TargetIP参数。

关键知识点：跨平台联机测试

支持不同操作系统间的联机测试，包括Windows、Linux（Ubuntu）等多种系统，只要在同一局域网内即可实现数据传输。测试前需确保：

- 所有主机在同一个局域网内
- RflySimSDK库已正确复制到每台主机并导入Python环境
- 防火墙设置允许相关端口通信

程序解析

UAV1Ctrl.py 解析

```
1 import time
2 import math
3 import sys
4 import ReqCopterSim
5 import PX4MavCtrlV4 as PX4MavCtrl
6 import NetSimAPIV4
7
8 CopterID=1
9
10 ## 下面代码在分布式仿真时，能够请求指定飞机返回数据到本电脑
11
12 # 创建一个CopterSim状态获取实例，并监听2s钟，获取局域网内所有CopterSim所在电脑的IP
13 req = ReqCopterSim.ReqCopterSim()
14
15 ## 获取目标电脑IP，并且配置CopterSim回传数据到本电脑
16 # 获取到指定CopterID的CopterSim所在电脑的IP
17 # 注：Windows下运行本函数获取TargetIP=127.0.0.1，如果在其他电脑上运行，获取CopterID对应电脑的
18 IP
19 TargetIP = req.getSimIpID(CopterID)
20
21 # 请求目标CopterSim将数据返回到本电脑
22 # 通过本接口，可以不用再去bat脚本里面填写IP地址了
23 req.sendReSimIP(CopterID)
24
25
26 ## 下面代码开始创建通信实例
27 #Create a new MAVLink communication instance, UDP sending port (CopterSim's
28 receiving port) is 20100
29 # 创建#CopterID号飞机的通信实例，和CopterSim#CopterID号相连，使用TargetIP确保分布式仿真也能
30 用
31 mav = PX4MavCtrl.PX4MavCtrl(CopterID,TargetIP)
32 net = NetSimAPIV4.NetSimAPI(mav)
33
34 #mav.InitMavLoop(UDPMODE), where UDPMODE=0,1,2,3,4
35 # Use UDP_Simple Mode to control PX4
36 # In this mode, this script will send struct inHILCMDData to CopterSim
37 # Then CopterSim convert it to MAVLinkOffboard message to PX4
38 # In this mode, PX4 send MAVLink data to CopterSim, which convert Struct data to
39 this script
40 # Obviously, UDP Simple is faster (Little data, low delay) than UDP Simple.
41 # 这里使用MAVLink_Full模式来传输数据
42 mav.InitMavLoop() # 连上1号飞控/CopterSim并准备好数据
43
44
45 time.sleep(2)
```

```
print('Start sending UAV1 data')  
# 收集到的本飞机的数据，不断转发给2号飞机  
net.enUavForward(2)
```

该脚本的主要功能是：

- 创建无人机通信实例
- 自动检测CopterSim所在的IP地址
- 启动数据转发功能，将本机数据转发给2号无人机

I UAV2Ctrl.py 解析

```
1 import time
2 import math
3 import sys
4 import ReqCopterSim
5 import PX4MavCtrlV4 as PX4MavCtrl
6 import NetSimAPIV4
7
8 CopterID=2
9
10 ## 下面代码在分布式仿真时，能够请求指定飞机返回数据到本电脑
11
12 # 创建一个CopterSim状态获取实例，并监听2s钟，获取局域网内所有CopterSim所在电脑的IP
13 req = ReqCopterSim.ReqCopterSim()
14
15 ## 获取目标电脑IP，并且配置CopterSim回传数据到本电脑
16 # 获取到指定CopterID的CopterSim所在电脑的IP
17 # 注：Windows下运行本函数获取TargetIP=127.0.0.1，如果在其他电脑上运行，获取CopterID对应电脑的
18 IP
19 TargetIP = req.getSimIpID(CopterID)
20
21 # 请求目标CopterSim将数据返回到本电脑
22 # 通过本接口，可以不用再去bat脚本里面填写IP地址了
23 req.sendReSimIP(CopterID)
24
25
26 ## 下面代码开始创建通信实例
27 #Create a new MAVLink communication instance, UDP sending port (CopterSim's
28 receving port) is 20100
29 # 创建#CopterID号飞机的通信实例，和CopterSim#CopterID号相连，使用TargetIP确保分布式仿真也能
30 用
31 mav = PX4MavCtrl.PX4MavCtrl(CopterID,TargetIP)
32 net = NetSimAPIV4.NetSimAPI(mav)
33
34 #mav.InitMavLoop(UDPMODE), where UDPMODE=0,1,2,3,4
35 # Use UDP_Simple Mode to control PX4
36 # In this mode, this script will send struct inHILCMDData to CopterSim
37 # Then CopterSim convert it to MAVLinkOffboard message to PX4
38 # In this mode, PX4 send MAVLink data to CopterSim, which convert Struct data to
39 this script
40 # Obviously, UDP Simple is faster (Little data, low delay) than UDP Simple.
41 # 这里使用MAVLink_Full模式来传输数据
42 mav.InitMavLoop()
43
44
45 # 开始监听所有发给自己飞机的数据
46 # 注：2号飞机使用60000+2号监听端口
47 net.StartNetRecOwn()
48 # 或者使用net.StartNetRec(2)来指定接收发给2号飞机
49
50
```

```

51 # 下面的函数检查是否收到指定的无人机数据，这里以1 2 3 4号飞机为例
52 while True:
53
54     if net.UavDict.__contains__(str(1)): #
55         print('Got UAV1 Data')
56         break
57
58     time.sleep(1)
59
60 # 跳出等待循环，打印数据
uav1 = net.UavDict['1']
print('UAV1 Data:
',uav1.CopterID,uav1.uavAngEular,uav1.uavVelNED,uav1.uavAngEular,uav1.uavPosNED,uav
1.uavGlobalPos)

# 退出程序
net.endNetLoop()

```

该脚本的主要功能是：

- 监听2号无人机的数据接收
- 检测并接收来自1号无人机的数据
- 打印接收到的无人机数据
- 正确关闭网络连接

核心技术点

1. **自动IP检测**: 通过 `ReqCopterSim.ReqCopterSim()` 实现自动IP检测，简化了分布式部署流程
2. **数据转发**: 通过 `net.enUavForward(2)` 实现将1号无人机数据转发给2号无人机
3. **数据监听**: 通过 `net.StartNetRecOwn()` 和 `net.UavDict` 实现对特定无人机数据的监听和获取
4. **网络管理**: 通过 `net.endNetLoop()` 正确关闭网络连接，防止资源泄露

6.参考资料

1. [RflySim官方文档](#)
2. [MAVLink协议规范](#)
3. [PX4飞控开发指南](#)

7. 常见问题

Q1: 运行程序时提示找不到CopterSim IP地址怎么办?

A1: 检查网络连接是否正常，确认各设备是否在同一局域网内；如果是手动指定IP，则需要在代码中明确设置TargetIP参数。

Q2: 仿真环境初始化时无法显示"GPS 3D fixed & EKF initialization finished"信息?

A2: 可能是CopterSim仿真进程启动失败或者加载模型时间较长，请稍等片刻或重启仿真环境。确保系统资源充足，关闭不必要的应用程序。

Q3: 多台电脑联机时无法正常接收数据?

A3: 确认防火墙设置是否阻止了相关端口的通信；检查各台电脑是否处于同一网段；验证Python环境中的RflySimSDK库是否正确安装。

1. <https://rflysim.com/> ↩

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