多无人机广播通信实验

1. 实验目的

本实验旨在演示如何通过RflySim平台实现多无人机之间的广播通信。通过使用NetSimAPIV4库和PX4MavCtrlV4库，实现多架无人机的状态数据广播与接收。学习者将掌握：

- 如何建立与多架无人机的通信连接
- 如何配置无人机数据广播转发
- 如何接收并解析来自多架无人机的状态数据
- 分布式仿真环境下多无人机协同工作的基本原理
- 广播通信机制在多无人机系统中的应用

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\9.RflySimComm\0.ApiExps\e1.RflyNetAPIExps\2.BroadSendData

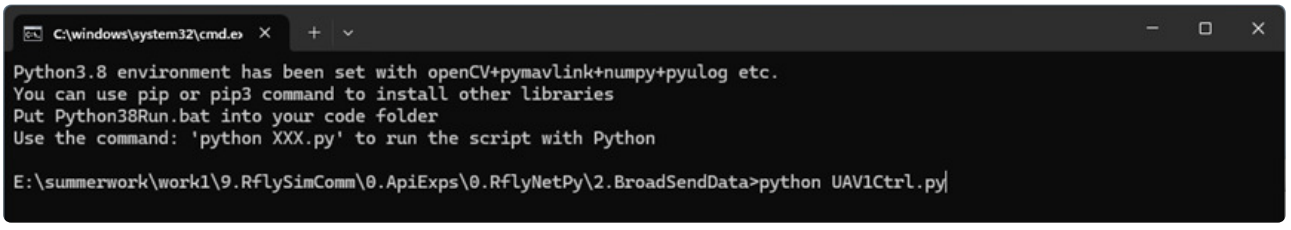
Python38Run.bat：RflySim Python运行脚本
SITLRun3MavlinkFull.bat：启动3架无人机SITL仿真环境脚本
UAV1Ctrl.py：第1架无人机控制脚本
UAV2Ctrl.py：第2架无人机控制脚本
UAV3Ctrl.py：第3架无人机控制脚本
UavPythonRunALL.bat：批量启动所有无人机控制脚本

4. 实验内容或步骤

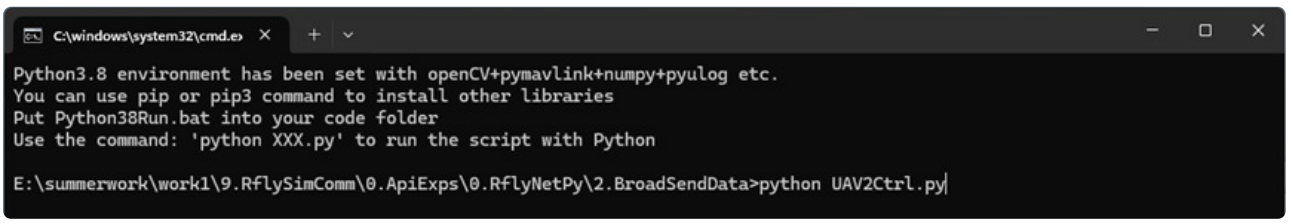
4.1 步骤1：本机实验

双击运行SITLRun.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：

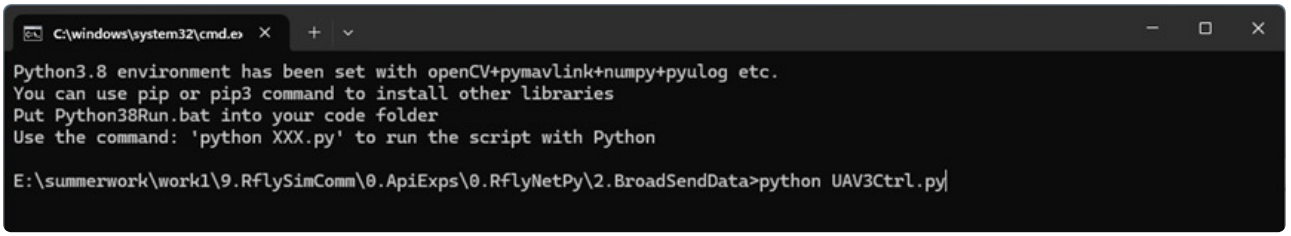
双击运行"Python38Run.bat"，在其中输入"python UAV1Ctrl.py"，启动1号飞机广播收发程序。



双击运行"Python38Run.bat"，在其中输入"python UAV2Ctrl.py"，启动2号飞机广播收发程序。

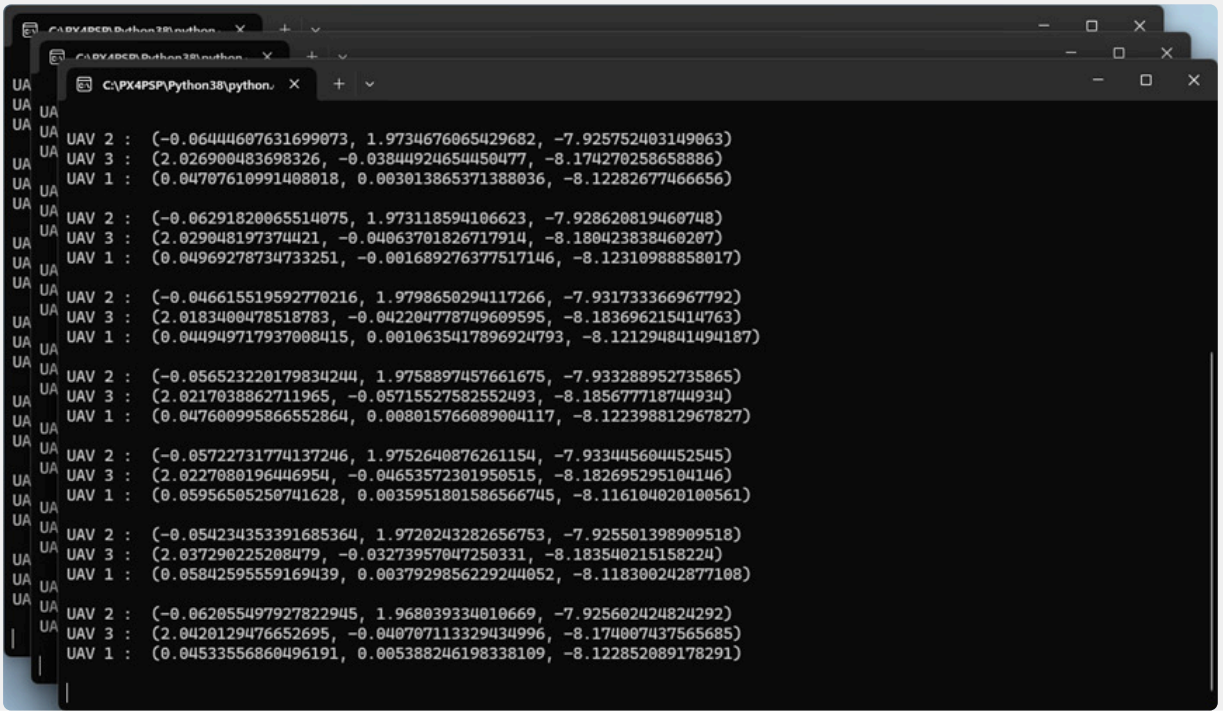


双击运行"Python38Run.bat"，在其中输入"python UAV3Ctrl.py"，启动3号飞机广播收发程序。



观察每个窗口，每个程序都打印自己收到的数据，并输出。

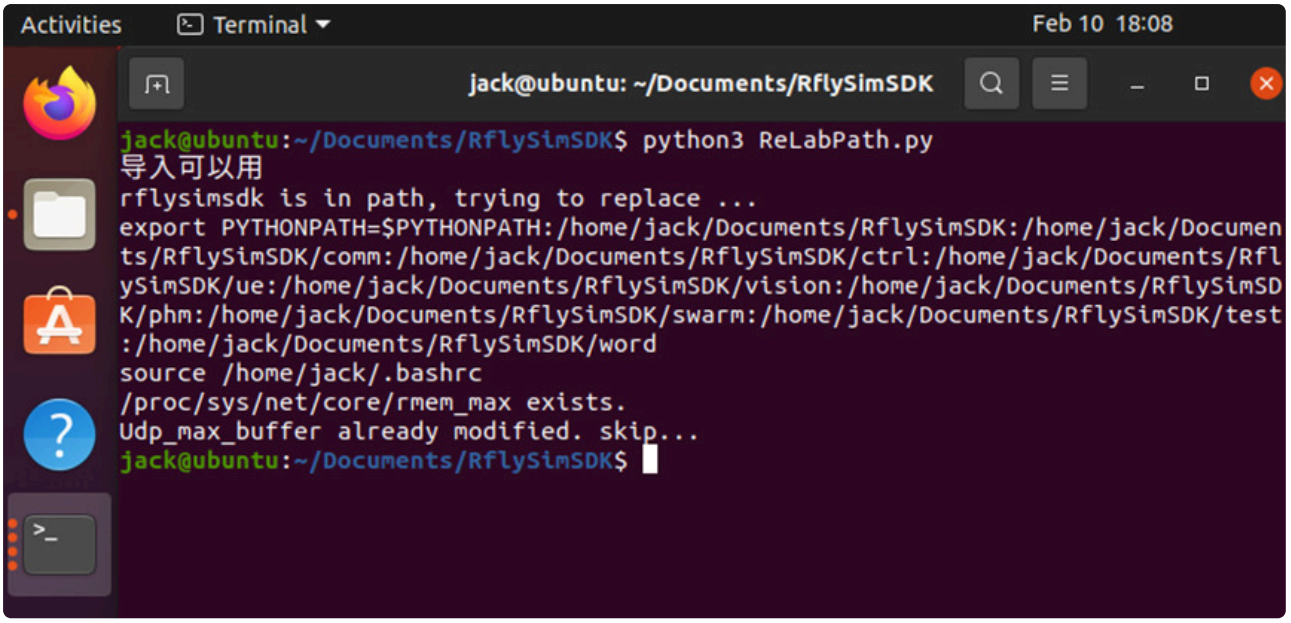
> 注：上述步骤可以用直接双击运行"UavPythonRunALL.bat"来替代，会自动完成上述操作。



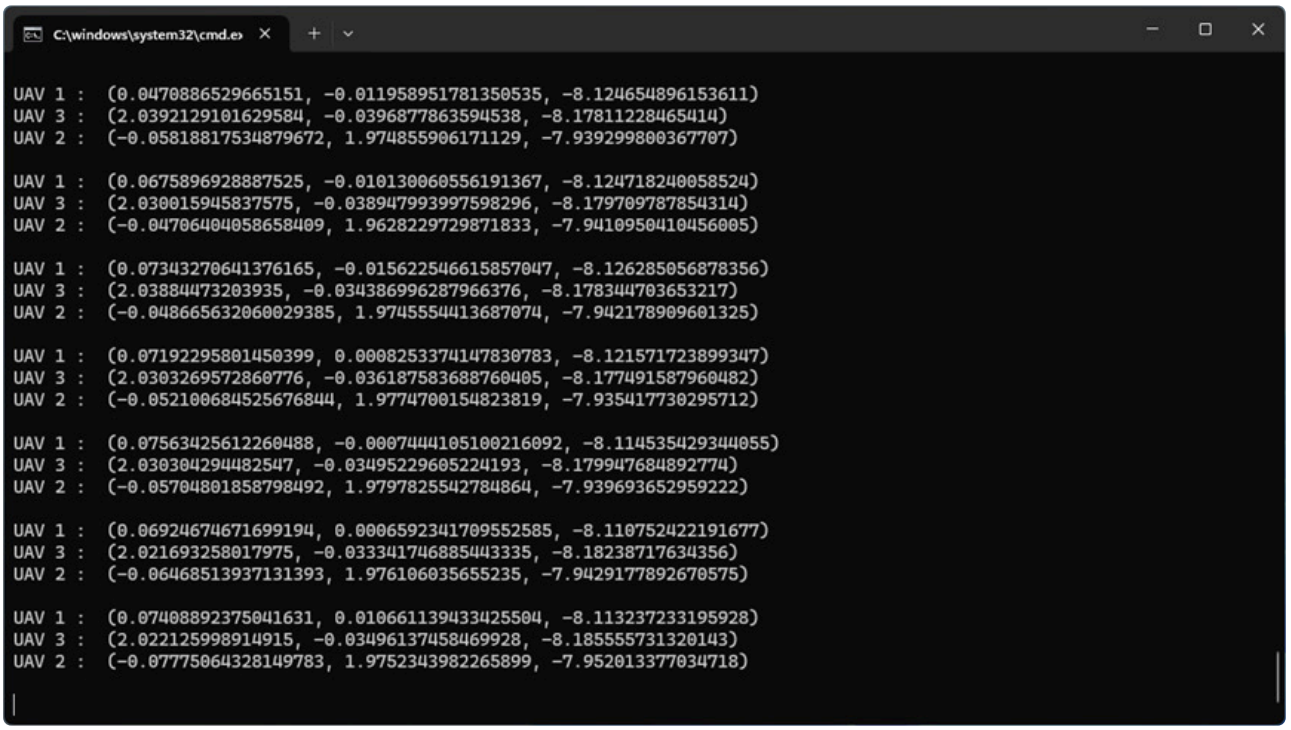
4.2 步骤2：联机实验

1）准备3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。

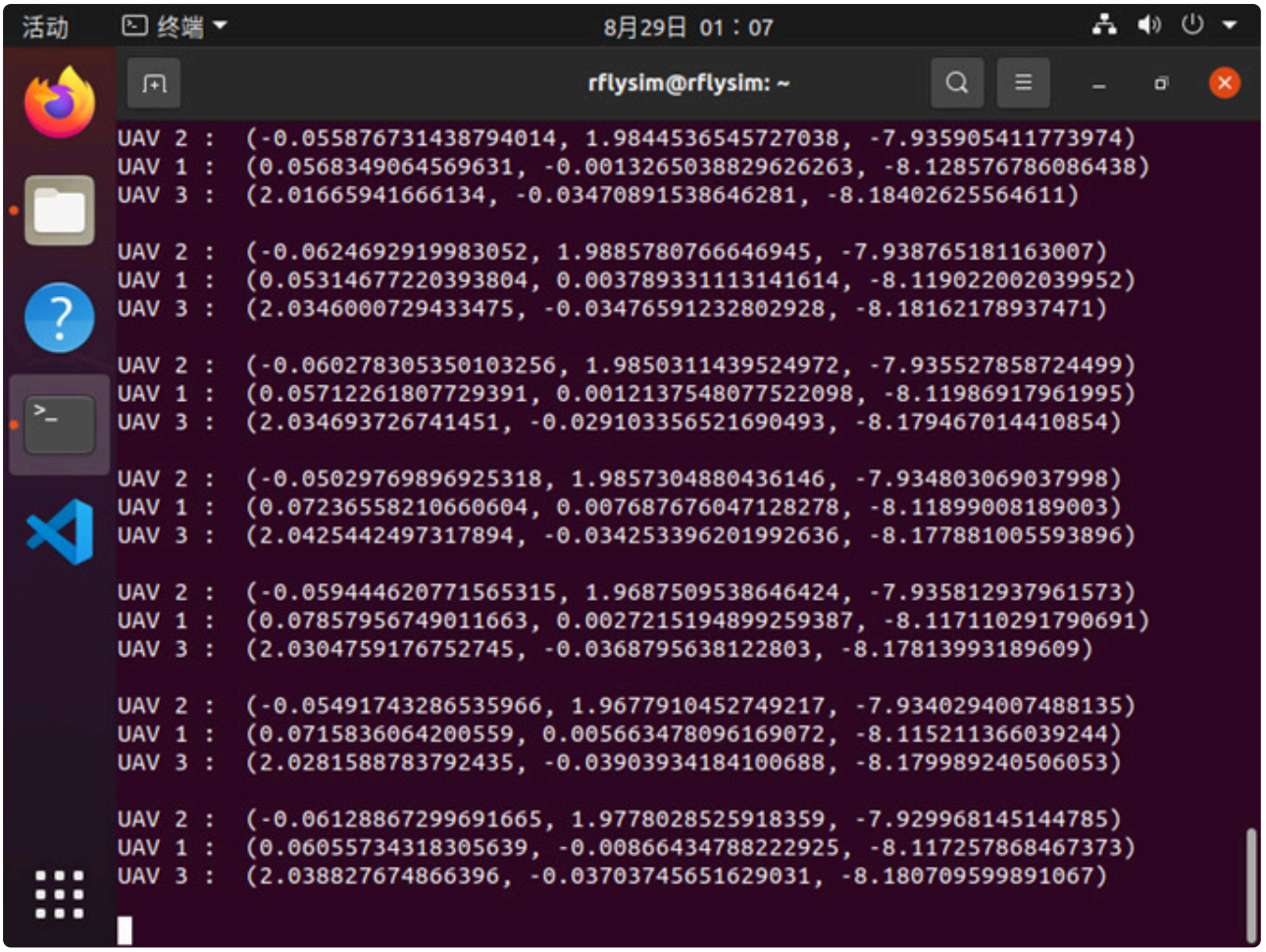
2）按照[RflySim按安装理解]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e3.PythonConfig实验将[RflySim安装路径]\RflySimAPIs\RflySimSDK\RflySimSDK库复制到每台主机上，并确保RflySimSDK已经导入当前主机的Python环境中。



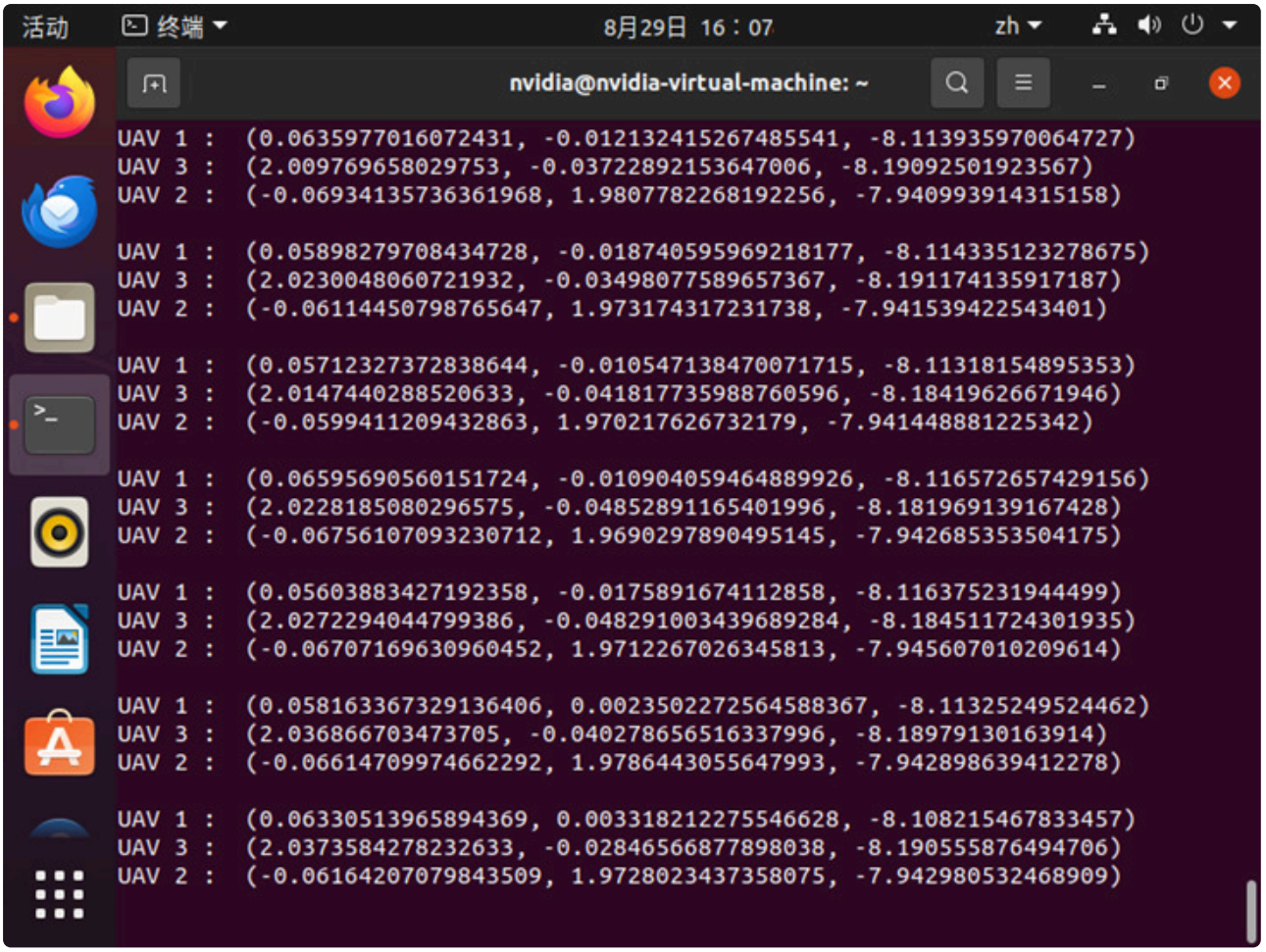
在Windows主机中，双击打开SITLRun3MavlinkFull.bat自动创建三个飞机。等待所有CopterSim的左下角打印"PX4: GPS 3D fixed & EKF initialization finished."。再运行UAV1Ctrl.py，启动1号飞机广播收发程序。



在一台Ubuntu虚拟机中，运行UAV2Ctrl.py，启动2号飞机广播收发程序。



在另一台Ubuntu虚拟机中，运行UAV3Ctrl.py，启动3号飞机广播收发程序。



能看到和本地测试一样的效果，说明测试正确。

5. 关键知识点

关键知识点1：整体思路和框架

本实验采用RflySim平台的NetSimAPIV4库实现多无人机广播通信。整体架构包括：

1. 通过ReqCopterSim模块动态获取CopterSim仿真器的IP地址
2. 使用PX4MavCtrlV4建立与特定无人机的MAVLink通信
3. 利用NetSimAPIV4启用广播转发并接收多无人机数据
4. 循环读取并显示所有无人机的状态信息

关键知识点2：Python程序解析

无人机控制脚本（UAV1Ctrl.py，UAV2Ctrl.py，UAV3Ctrl.py）

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

CopterID=1  # 定义无人机ID，UAV1Ctrl.py为1，UAV2Ctrl.py为2，UAV3Ctrl.py为3
```

以上是脚本的导入部分，引入了时间、数学、系统库以及RflySim专用模块。CopterID变量标识当前控制的是哪架无人机。

```
# 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
req = ReqCopterSim.ReqCopterSim()

# 获取目标电脑IP，并且配置CopterSim回传数据到本电脑
TargetIP = req.getSimIpID(CopterID)

# 请求目标CopterSim将数据返回到本电脑
req.sendReSimIP(CopterID)
```

这部分代码实现了动态IP获取功能。ReqCopterSim.ReqCopterSim()创建了一个实例来检测局域网内的CopterSim实例，getSimIpID()方法获取指定无人机所在的计算机IP，sendReSimIP()方法配置该无人机将数据回传至本机。

```
# 创建通信实例
mav = PX4MavCtrl.PX4MavCtrler(CopterID,TargetIP)
net = NetSimAPIV4.NetSimAPI(mav)

# 初始化MAVLink通信
mav.InitMavLoop()  # 连接飞控/CopterSim并准备数据交换
```

这里创建了MAVLink通信实例，PX4MavCtrler负责与指定无人机建立连接，NetSimAPI封装了更高层的网络通信功能。InitMavLoop()方法初始化MAVLink通信循环。

```python
time.sleep(2)

print('Broadcast uav data')
net.enUavForward()  # 启用无人机数据转发功能

time.sleep(2)
print('Start receving all uav data')
net.StartNetRec()   # 开始接收网络中的无人机数据

while True:
    for uav in net.UavData:
        print('UAV',uav.CopterID,': ',uav.uavGlobalPos)
    if len(net.UavData)>0:
        print('') # 空一行，便于观察数据
    time.sleep(1)
```

这是核心的数据广播和接收循环。enUavForward()启用数据转发功能，使本机可以接收其他无人机的数据；StartNetRec()开始接收网络中的无人机数据；循环体遍历net.UavData集合，打印每架无人机的ID和全局位置信息。

关键知识点3：技术要点

- 分布式仿真支持：通过ReqCopterSim模块动态获取IP，支持跨计算机的仿真环境
- 广播通信机制：启用数据转发后，各节点可以接收网络中其他无人机的数据
- 状态监控：实时显示所有无人机的位置信息，便于监控集群状态

6.参考资料

1. RflySim官方文档
2. MAVLink协议详解
3. PX4开源飞控系统

7.常见问题

Q1：运行Python脚本时报错"ModuleNotFoundError: No module named 'ReqCopterSim'"。

A1：这表示RflySim SDK库没有正确安装或未添加到Python环境中。请确认已安装RflySim工具链，并将RflySimSDK库路径添加到Python环境变量中。

Q2：运行脚本后无法看到其他无人机的数据，只显示自己的数据。

A2：可能是数据转发功能未正常启用或网络配置问题。请检查是否正确调用了net.enUavForward()方法，并确认所有计算机处于同一局域网中，防火墙设置允许相应端口通信。

Q3：多台计算机联合实验时，无人机状态数据更新延迟较大。

A3：这通常是由于网络延迟或带宽不足造成的。请确保局域网连接稳定，尝试减少同时运行的无人机数量，并检查防火墙和路由器设置是否限制了UDP广播包的传输。

---

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