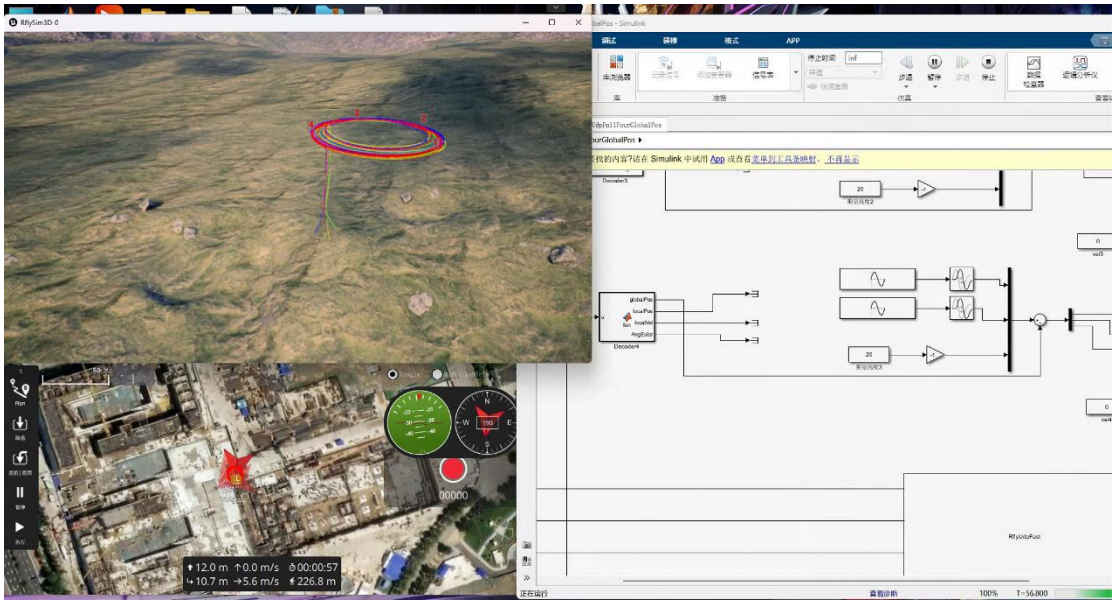
1、实验名称及目的

通信接口 FullData 模式全局坐标控制 4 机实验：通过平台提供的 RflyUdpFast 传输模块，接收无人机的状态信息，然后进行对无人机的全局位置运动控制进行 Simulink 建模发送控制指令到该模块，然后进行仿真。

2、实验效果

该实验可以看到 4 架无人机在进行全局位置状态的获取与控制的圆周运动。



3、文件目录

文件夹/文件名称	说明
RflyUdpFullFourGlobalPos.bat	启动仿真配置文件
RflyUdpFullFourGlobalPos.slx	实现功能主文件
RflyUdpFast.mexw64	RflyUdpFast 传输模块编译文件
Init.m	参数初始文件
RflyUdpFullFourGlobalPos.exe	EXE 格式的 Simulink 控制器文件。
HITLRunUdpFull.bat	硬件在环仿真一键启动脚本文件

4、运行环境

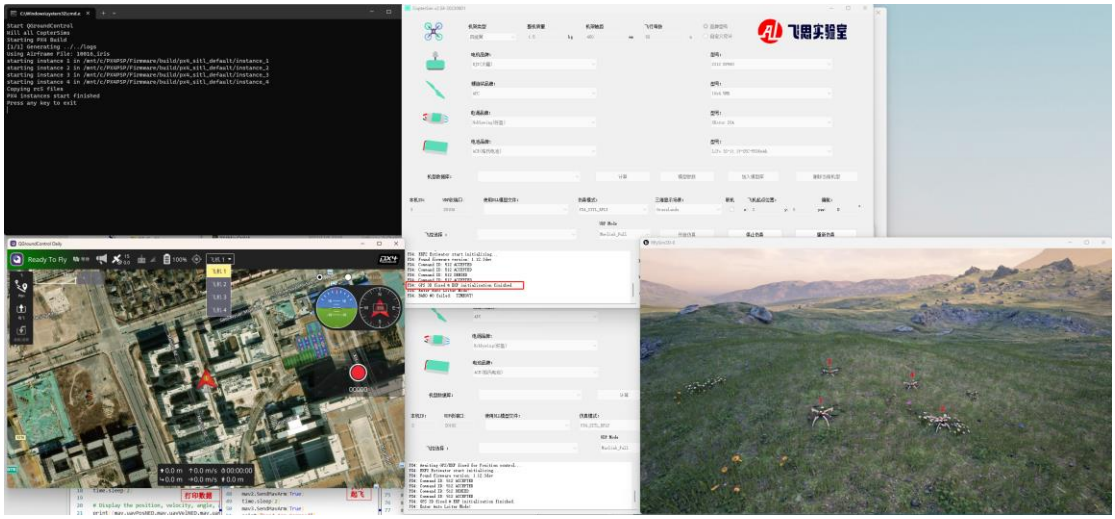
序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑①	1
2	RflySim 平台免费版		
3	MATLAB 2017B		

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

5、软件在环仿真实验步骤

Step 1：

执行 RflyUdpFullFourGlobalPos.bat 文件。将会启动 QGC 地面站，4 个 CopterSim 软件和 1 个 RflySim3D，等待 4 个 CopterSim 软件消息栏日志必须打印出 GPS 3D fixed & EKF in itialization finished 字样代表初始化完成，且 RflySim3D 软件内有 4 架飞机。如下图所示：



Step 2：

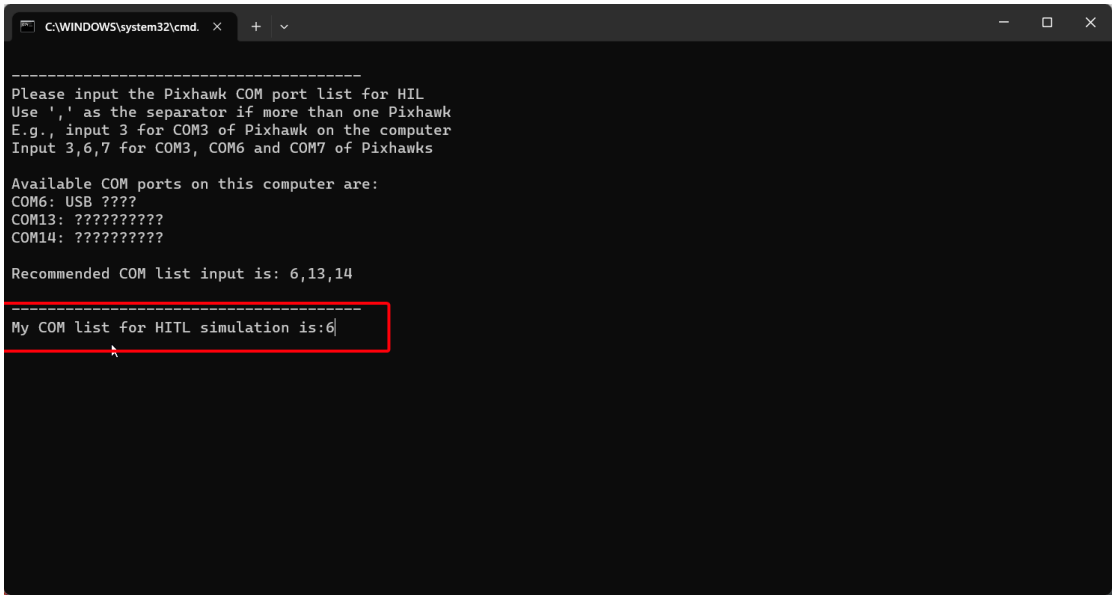
用 MATLAB2017B 及以上版本将工作空间打开到当前实验路径，首先运行 Init.m 文件，然后运行 RflyUdpFullFourGlobalPos.slx 模型。或者直接双击运行 RflyUdpFullFourGlobalPos.exe 文件也可直接启动仿真。即可在 UE 中看到无人机的运动状态，其效果如下图。

6、硬件在环仿真实验步骤

Step 1：

双击运行 HITLRunUdpFull.bat 脚本一键启动硬件在环仿真，在弹出的对话框中。输入

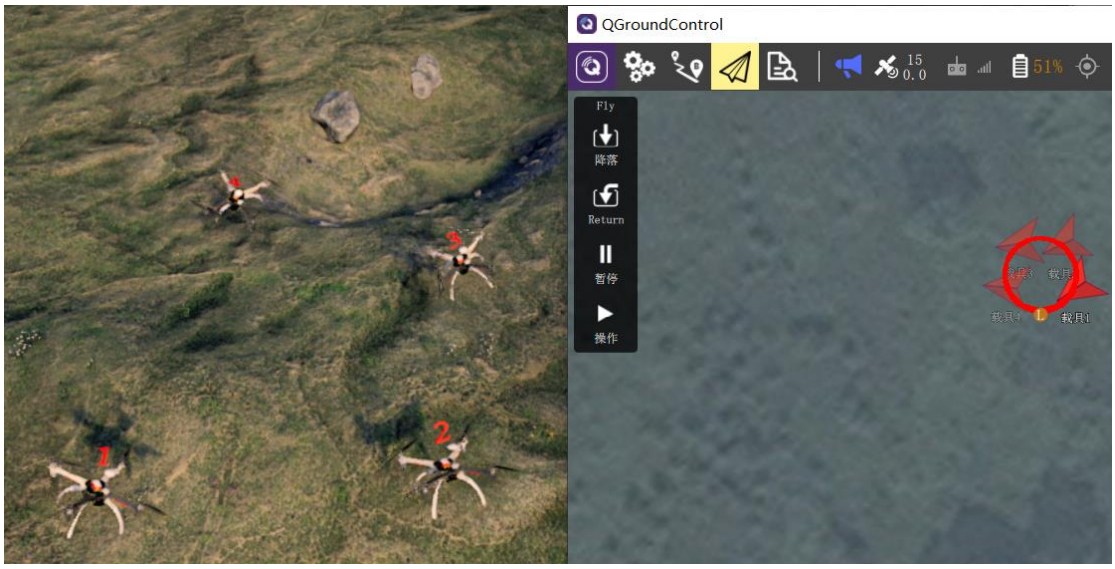
与本实验相同飞机数量的飞控的端口号。



即可与 SIL 仿真实验类似，打开相同数量的 RflySim3D、QGC、CopterSim 软件。

Step 2：

通过遥控器或 QGC 即可解锁无人机起飞。



注：硬件在环实验遥控器设置与飞控数据线链接方式请见本平台实验：*\PX4PSP\RflySimAPIs\2.RflySimUsage\1.BasicExps\e11_RC-Config\Readme.pdf

7、参考资料

[1] 通信接口的 FullData 模式数据协议

模块输入为 15 维的 double 型向量，具体定义（实现 MAVLink 的 Offboard 消息）如下：

- 第 1 维：time_boot_ms; %当前时间戳（填 0 即可，目前没有使用）
- 第 2 维：copterID; %飞机 ID（填 1 即可，目前没有使用）

- 第 3 维：type_mask; %输入控制模式（同 Offboard 定义）
- 第 4 维：coordinate_frame; %坐标系模式（同 Offboard 定义）
- 第 5~15 维：ctrls[11]; %分别对应了 3 维的期望位置 pos,3 维的期望速度 vel，3 维的期望加速度 acc，1 维的期望偏航角 yaw，1 维的期望偏航角速度 yawRate。（同 Offboard 定义）

模块输出为 28 维的 double 型向量（全部转发自 Pixhawk 内部滤波值），具体定义如下:

- 第 1~3 维：gpsHome[3]; %Home 点（上电之后不会变）的经纬高坐标，经纬度需要除以 1e7 才能得到度为单位的经纬度，高需要除以 1e3 才能得到 m 为单位的高（向上为正）
- 第 4~6 维：AngEular[3]; %Pixhawk 估计得到的姿态欧拉角，单位弧度
- 第 7~9 维：localPos[3]; %Pixhawk 估计得到的以 gpsHome 为原点的相对北东地位置向量，单位 m，z 轴向下为正
- 第 10~12 维：localVel[3]; %北东地的运动速度向量，单位 m/s
- 第 13~15 维：GpsPos[3]; %实时的 GPS 位置，单位和 gpsHome 相同，但是会实时变化
- 第 16~18 维：GpsVel[3]; %GPS 速度，需要除以 100 得到 m/s 为单位的速度
- 第 19 维：time_boot_ms；%上电时间
- 模块输出为 28 维的 double 型向量（全部转发自 Pixhawk 内部滤波值），具体定义如下
- 第 1~3 维：gpsHome[3]; %Home 点（上电之后不会变）的经纬高坐标，经纬度需要除以 1e7 才能得到度为单位的经纬度，高需要除以 1e3 才能得到 m 为单位的高（向上为正）
- 第 4~6 维：AngEular[3]; %Pixhawk 估计得到的姿态欧拉角，单位弧度
- 第 7~9 维：localPos[3]; %Pixhawk 估计得到的以 gpsHome 为原点的相对北东地位置向量，单位 m，z 轴向下为正
- 第 10~12 维：localVel[3]; %北东地的运动速度向量，单位 m/s
- 第 13~15 维：GpsPos[3]; %实时的 GPS 位置，单位和 gpsHome 相同，但是会实时变化
- 第 16~18 维：GpsVel[3]; %GPS 速度，需要除以 100 得到 m/s 为单位的速度
- 第 19 维：time_boot_ms；%上电时间。

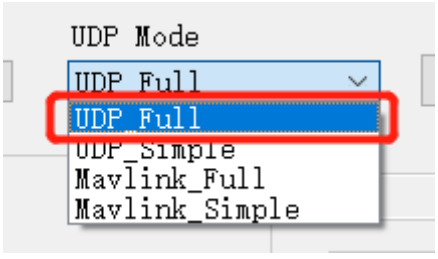
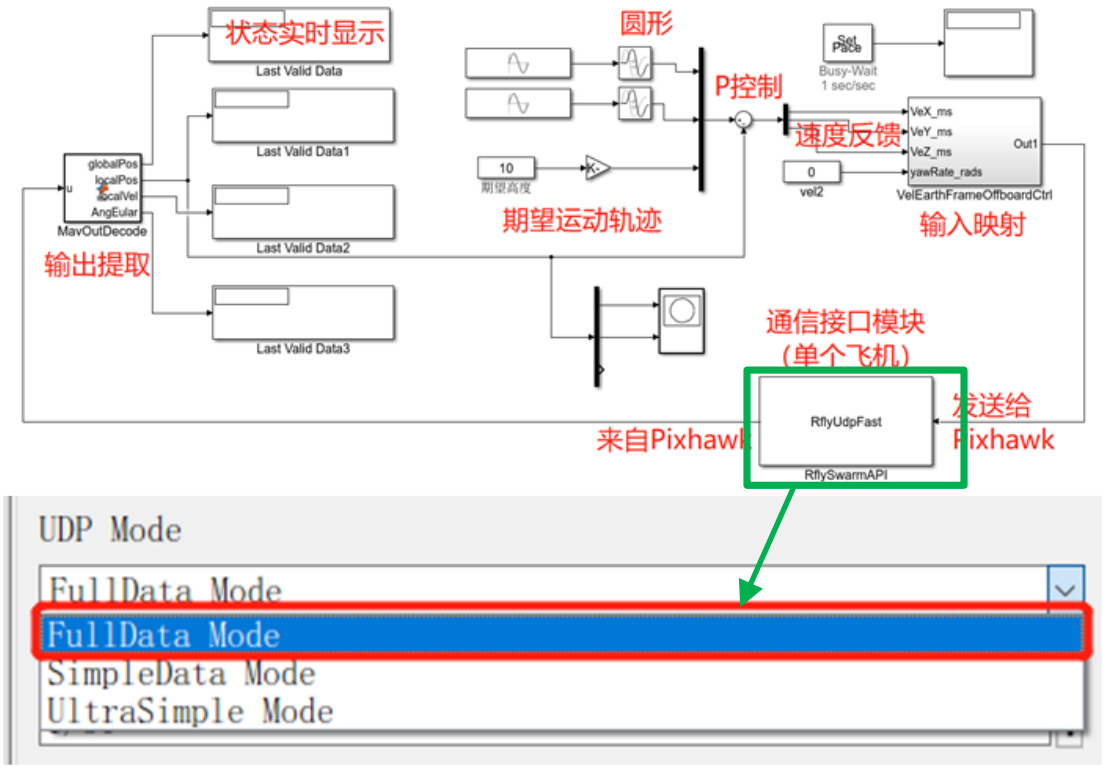


图 1 CopterSim 中 UDP 模式选择



[2] 将 RflySwarmAPI 模块的 28 维输出（见 2.5 节定义）提取出感兴趣的值: GpsHome、LocalPos、AngEular 等。

将 GpsHome 的经纬高数据转为 CopterSim 的全局坐标 GlobalPos 的 xyz 单位 m 的数据。代码如下：

```
function [globalPos,localPos,localVel,AngEular] = fcn(u)
GpsHomePos = u(1:3);
AngEular = u(4:6);
localPos = u(7:9);
localVel = u(10:12);
globalPos = [0,0,0];
if ~(abs(GpsHomePos(1))<1&&abs(GpsHomePos(2))<1)
   globalPos(1)=(GpsHomePos(1)*1e-7-GPSOrigin(1))/180*pi*6362000+localPos(1);
   globalPos(2)=(GpsHomePos(2)*1e-7-GPSOrigin(2))/180*pi*4.8823e6+localPos(2);
   globalPos(3)=-(GpsHomePos(3)*1e-3-GPSOrigin(3))+localPos(3);
end
```

注： GPSOrigin=[40.1540302,116.2593683,50]是 RflySim 仿真的 GPS 原点，在 Init.m 和 RflyUdpFast.cpp 等的初始化区域定义。如果通过 CopterSim 的 txt 地形文件（见第 5 讲 6.4 节）修改了 GPS 原点坐标，这个地方需要根据实际情况修正。