



智能无人集群系统开发与实践

基于RflySim工具链的全栈开发案例

第6讲 外部控制与轨迹规划



大纲

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结



1. 安装方法

1.1 需要安装的组件

- Visual Studio 2017（体验版和完整版都需要安装）
- 为MATLAB配置C++ 编译器（体验版和完整版都需要安装）
- Matlab 2023a*（高级完整版安装）

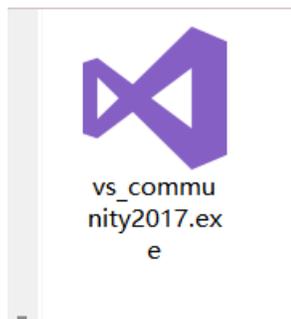
下面介绍Visual Studio 2017的安装方法（需要联网）：
在本平台中，已经放置了Visual Studio 2017的安装包



1. 安装方法

1.2 Visual Studio 2017的安装方法

- 首先，我们可以打开平台安装的位置，找到*:\PX4PSP\RflySimAPIs此处位置，此处放置的是平台中的一些例程以及软件的安装包
- 之后，我们可以打开第四章的内容，找到基础版的例程，4.RflySimModel\1.BasicExps，我们可以在其中找到名为VS2017Installer的文件夹，其中便是Visual Studio 2017的安装包。



在线安装步骤（需联网）如下：双击
“RflySimAPIs\SimulinkControlAPI\VS2017Installer\vs_community2017.exe”



1. 安装方法

1.2 Visual Studio 2017的安装方法

- 安装 Visual Studio 2017 (也可以用其他版本, MATLAB能识别即可)。
- 后续课程很多地方都需要用到 Visual Studio 编译器, 例如 MATLAB S-Function Builder 模块的使用、Simulink 自动生成 C/C++ 模型代码等
- 本课程内容只需勾选右图的“C++的桌面开发”即可。





1. 安装方法

1.2 Visual Studio 2017的安装方法

- 注意：高版本MATLAB也可安装VS2019，但是MATLAB只能识别到低于自己版本的Visual Studio，因此MATLAB 2017b无法识别VS 2019。
- 注意：请不要更改VS默认安装目录（例如装到D盘），会导致MATLAB无法识别。
- 不能使用Mingw编译器，需VS



1. 安装方法

• 1.3 为MATLAB配置C++ 编译器

• 在MATLAB的命令行窗口中输入指令“mex -setup”

• 一般来说会自动识别并安装上VS 2017编译器，如右图所示显示“MEX 配置使用‘Microsoft Visual C++ 2017’以进行编译”说明安装正确

• 若有其他编译器，本页面还可以切换选择 VS 2013/2015等其他编译器

```
命令窗口
>> mex -setup
MEX 配置为使用 'Microsoft Visual C++ 2017 (C)' 以进行 C 语言编译。
警告: MATLAB C 和 Fortran API 已更改, 现可支持
包含 2^32-1 个以上元素的 MATLAB 变量。您需要
更新代码以利用新的 API。
您可以在以下网址找到更多的相关信息:
http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit

要选择不同的 C 编译器, 请从以下选项中选择一种命令:
Microsoft Visual C++ 2013 (C) mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2013.xml C
Microsoft Visual C++ 2015 (C) mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2015.xml C
Microsoft Visual C++ 2017 (C) mex -setup:C:\Users\dream\AppData\Roaming\MathWorks\MATLAB\R2

要选择不同的语言, 请从以下选项中选择一种命令:
mex -setup C++
mex -setup FORTRAN
fx >>
```



1. 安装方法

- 1.4 Matlab 2023a的安装方法
- MATLAB 安装包下载路径:
- <https://ww2.mathworks.cn/products/matlab.html>





大纲

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结



2. 关键接口介绍

2.0 基础实验总览

包括基础功能接口“RflySimAPIs/6.RflySimExtCtrl/0.ApiExps”以及基础例程“RflySimAPIs\6.RflySimExtCtrl\1.BasicExps”

详细参见[API.pdf](#)以及[README.pdf](#)

1.PX4MavCtrlAPITest	2023/11/16 10:17	文件夹
2.PX4ComAPITest	2023/11/16 10:59	文件夹
3.PX4MavGPSCTest	2023/11/16 10:17	文件夹
4.PX4RcCtrlAPITest	2023/11/16 10:17	文件夹
5.PX4MultiUavTest	2023/11/16 10:17	文件夹
6.PX4MavAccCtrlTest	2023/11/16 10:17	文件夹
7.PX4MavAttCtrlTest	2023/11/16 10:17	文件夹
8.GeoAPITest	2023/11/16 10:17	文件夹
9.UDPMode1TestShootBall	2023/11/16 10:48	文件夹
10.UDPMode0Test	2023/11/16 10:17	文件夹
11.UDPMode1Test	2023/11/16 10:17	文件夹
12.UDPMode2DefaultTest	2023/11/16 10:17	文件夹
13.UDPMode3Test	2023/11/16 10:51	文件夹
14.UDPMode4Test	2023/11/16 10:58	文件夹
15.CamObjGet	2023/11/16 11:14	文件夹
16.ReadTimeStmpGet	2023/11/16 10:17	文件夹
e0_ExtAPIUsage	2023/11/16 10:17	文件夹
e1_PosCtrl	2023/11/16 10:17	文件夹
e2_VelCtrl	2023/11/16 10:17	文件夹
e3_RCCtrl	2023/11/16 10:17	文件夹
e4_PyOffboardCtrl	2023/11/16 17:41	文件夹
e5_RackFlyCtrl	2023/11/16 10:17	文件夹
e6_PathTrackingCtrl	2023/11/16 18:19	文件夹
e7_MutUAVRemoteCtrl	2023/11/16 10:17	文件夹





2. 关键接口介绍

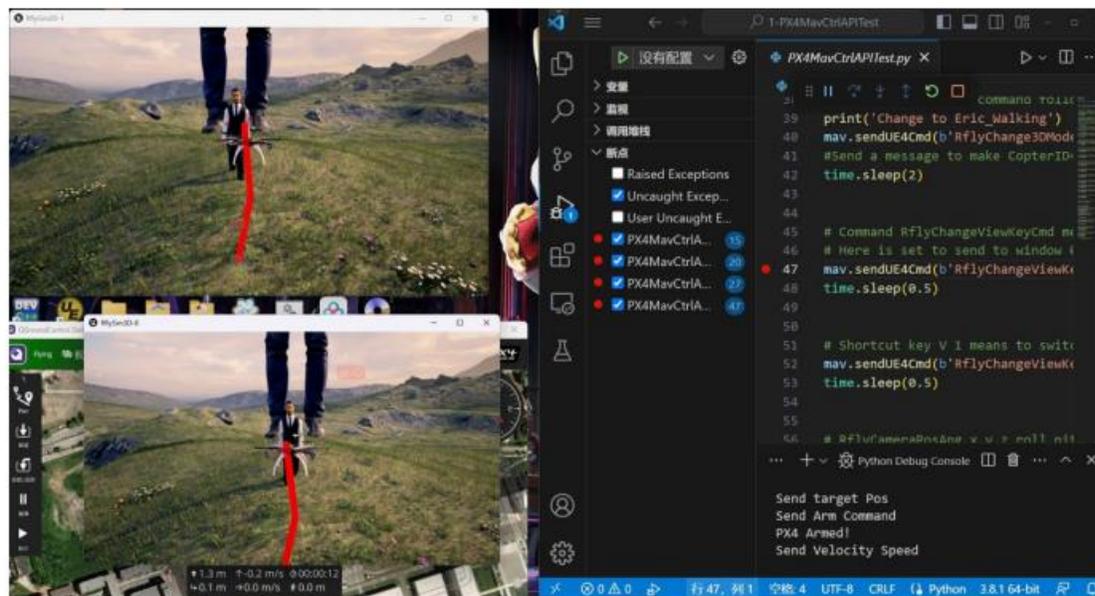
2.1 无人机控制接口调试实验

熟悉无人机 offboard 模式控制、
状态数据获取和 RflySim3

D 的控制接口，了解 SITL 通信框
架。

详细操作及实验效果见

[0.ApiExps\1.PX4MavCtrlAPITest\Readme.pdf](#)





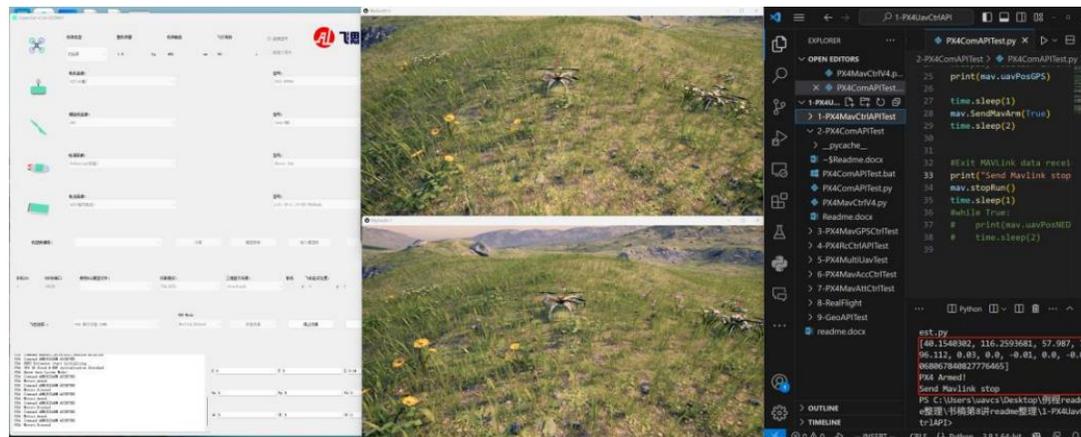
2.关键接口介绍

2.2数传连接 Pixhawk 6C 飞控 硬件在环仿真实验

用 MicroUSB 线连接电脑和
Pixhawk 6C飞控，开启一个飞
机的硬件在环仿真。

详细操作及实验效果见

[0.ApiExps\2.PX4ComAPITest\Readme.pdf](#)





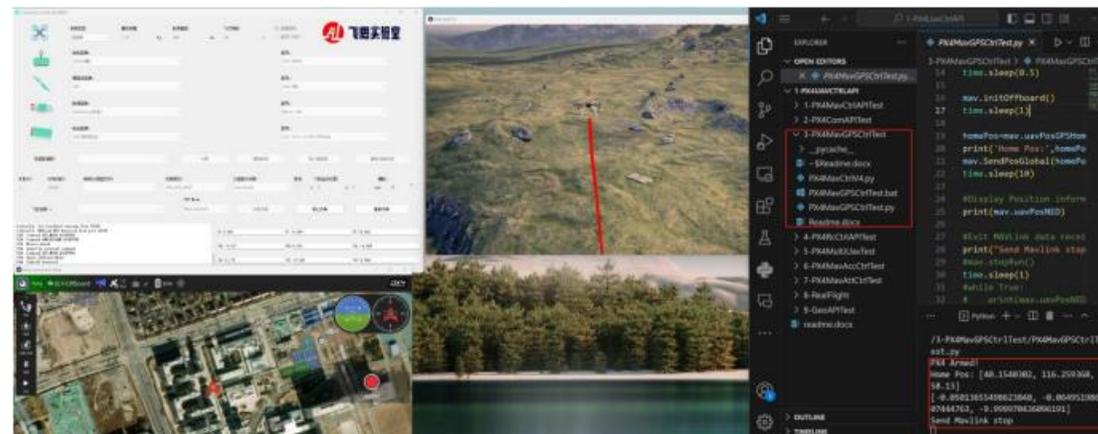
2.关键接口介绍

2.3无人机飞行控制实验

通过 RflySim 平台提供的
SendPosGlobal 函数接口实现控
制无人机移动。

具体实验操作及效果见

[0.ApiExps\3.PX4MavGPSCtrlTest\Readme.pdf](#)





2.关键接口介绍

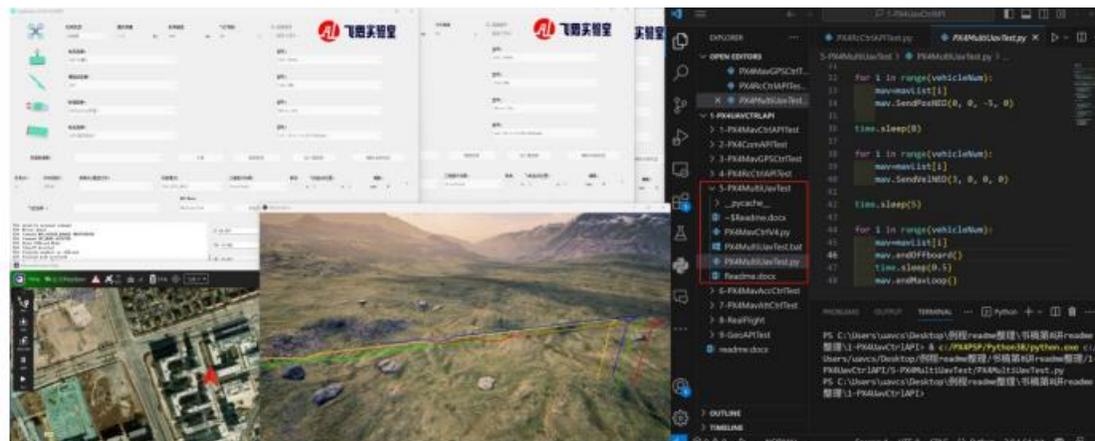
2.5多机 SITL 软件在环控制实验

根据平台提供的接口函数进行四个飞机的 offboard 模式下的位置控制以及速度控制 SITL 软件在环仿真。

具体实验操作及效果见

[0.ApiExps\5.PX4MultiUavTest\Readme.pdf](#)

```
16 mavList[0].sendUE4Cmd(b'RflyChangeViewKeyCmd S')
17 mavList[0].sendUE4Cmd(b'RflyChangeViewKeyCmd T')
18
```





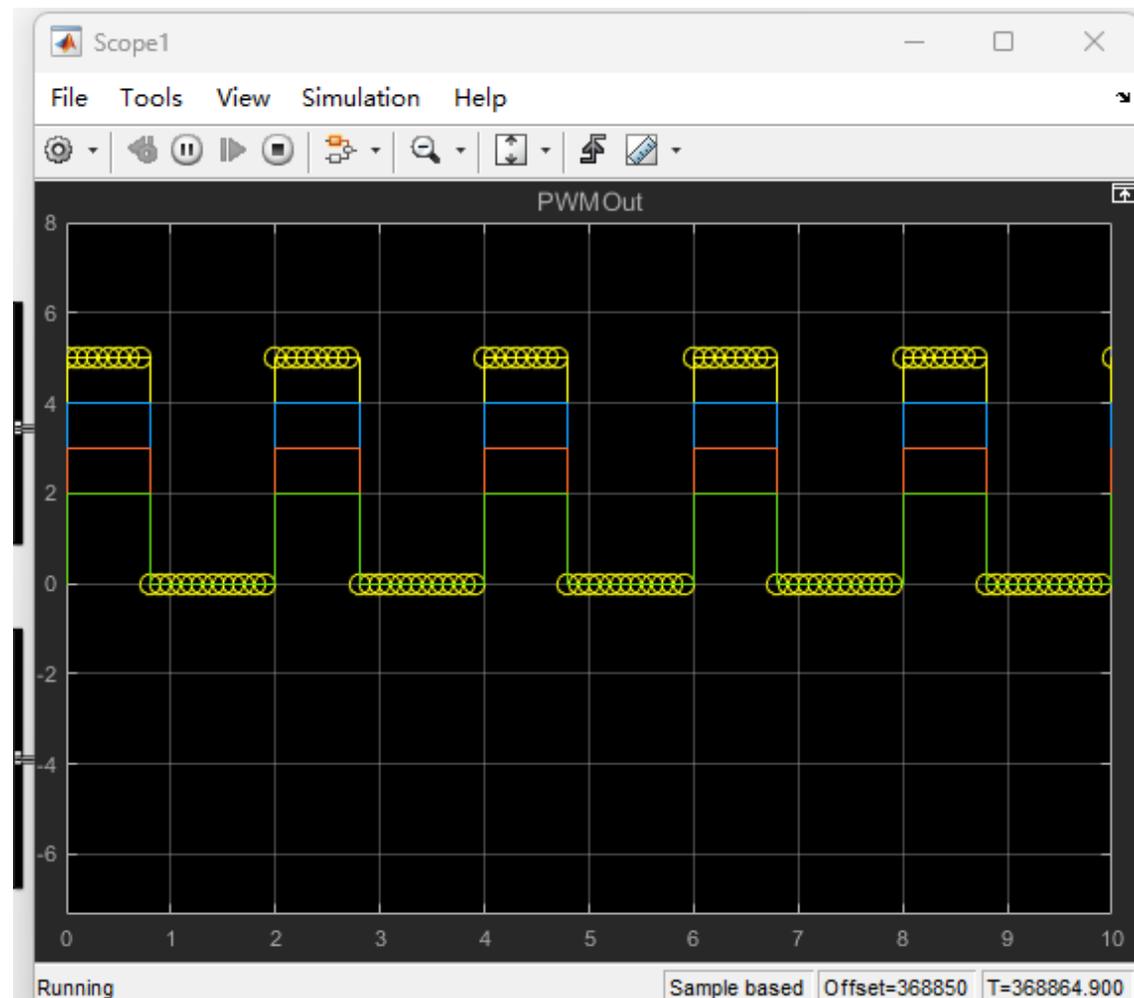
2.关键接口介绍

2.6无人机飞行加速度控制实验

通过使用平台提供的接口函数，通过 SendAccPX4 接口给飞机发送加速度指令。

具体实验操作见文件

[0.ApiExps\6.PX4MavAccCtrlTest\Readme.pdf](#)





2.关键接口介绍

2.7无人机飞行控制实验

通过利用 RflySim 平台提供的 SendAttPX4 接口给飞机发送期望姿态和油门数据。

具体实验操作见文件

[0.ApiExps\7.PX4MavAttCtrlTest\Readme.pdf](#)

```
13 mav.InitMavLoop()
14 time.sleep(0.5)
15
16 mav.initOffboard()
17 time.sleep(1)
18
19 mav.SendPosNED(0,0,-20)# 原地起飞, 到20米高度
20 time.sleep(15)
```

```
22 print('Current Thrust: ',mav.uavThrust) # 获取当前的悬停油门
23 mav.SendAttPX4([0,-10,0],mav.uavThrust)# 设置俯仰角为10度, 油门为悬停值
24 print('Send attitude command!')
25 time.sleep(2)
26 print('Current attitude: ', mav.uavAngEular[0]/math.pi*180, mav.uavAngEular[1]/math.pi*180, mav.uavAngEular[2]/math.pi*
27 print('Current altitude: ',mav.uavPosNED[2])
28 time.sleep(5)
29
30
31 # From PX4 Web: Acceleration setpoint values are mapped to create a normalized
32 # thrust setpoint (i.e. acceleration setpoints are not "properly" supported).
33
34 #Display Position information received from CopterSim
35 print(mav.uavPosNED)
36 time.sleep(8)
37
38 mav.SendAttPX4([0,-10,0],-5,0,1)# 设置俯仰角为10度, 保持高度为-5米
```



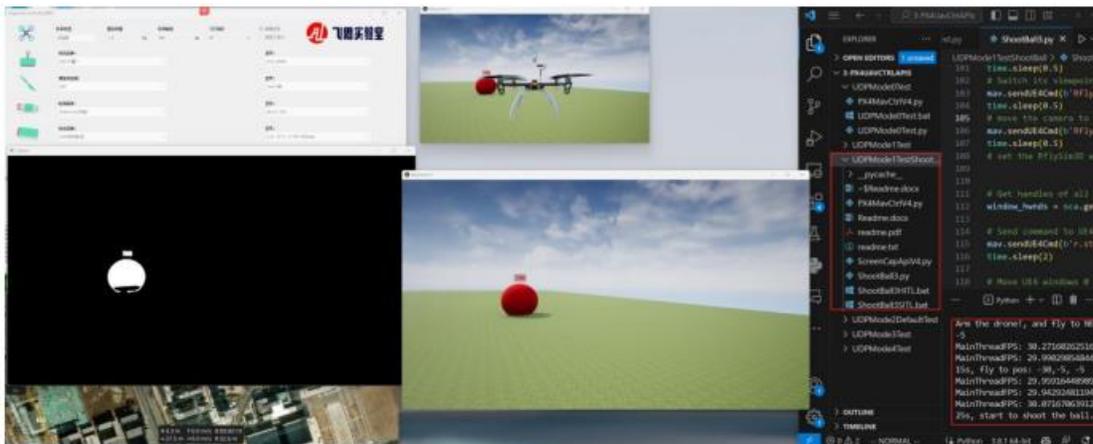

2.关键接口介绍

2.9视觉控制撞击小球实验

通过调用平台接口进行对 RflySim3D 软件内图像的捕获，并利用 opencv 进行图像处理，并进行控制指令解算，控制无人机运动。

具体实验操作见文件

[0.ApiExps\9.UDPMode1TestShootBall\Readme.pdf](#)





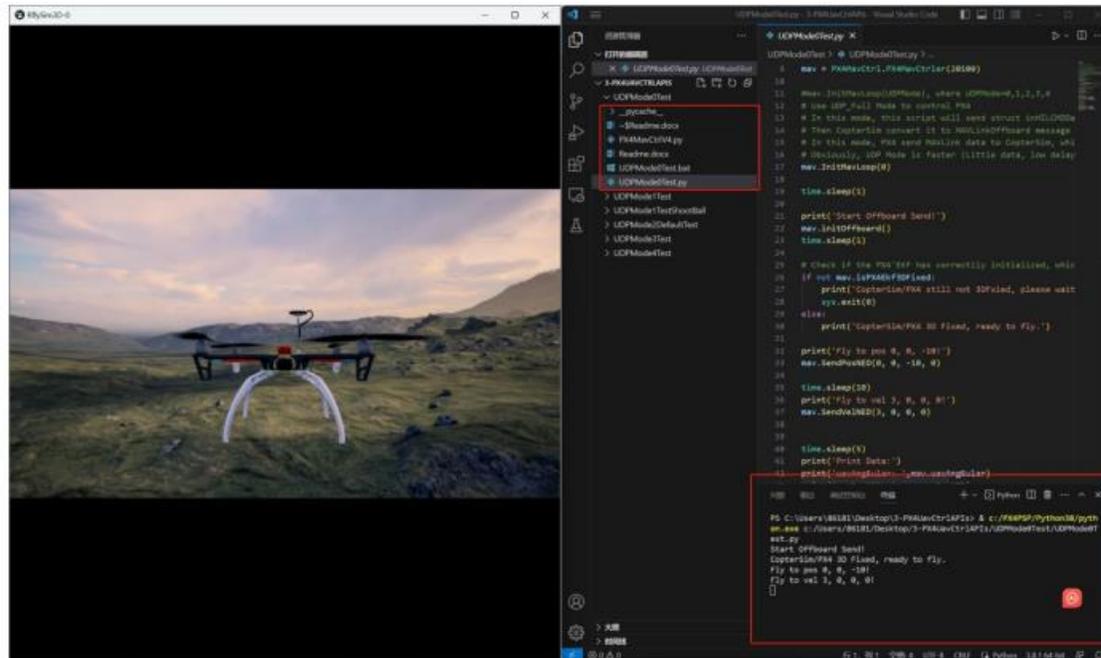
2.关键接口介绍

2.10 无人机通过 UDP_Full 通信实验

通过使用平台提供的接口函数，通过 UDP_Full 通信给飞机发送指令。

具体实验操作见文件

[0.ApiExps\10.UDPMODE0Test\Readme.pdf](#)





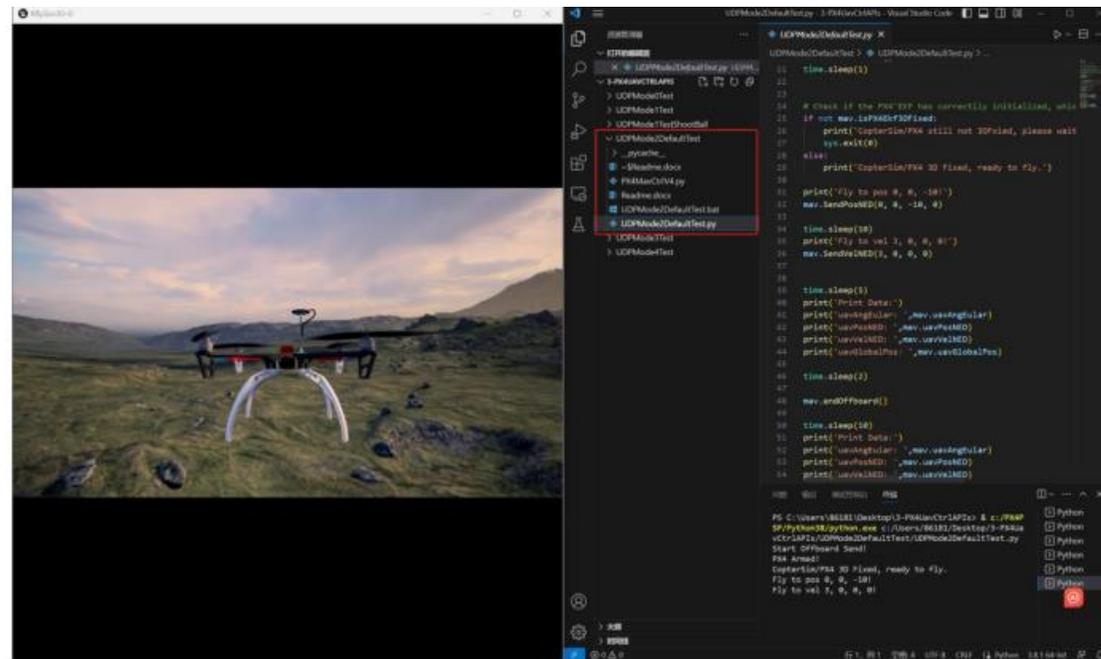
2.关键接口介绍

2.12无人机通过 MAVLink_Full 通信实验

通过使用平台提供的接口函数，通过 MAVLink_Full 通信给飞机发送指令。

具体实验操作见文件

[0.ApiExps\12.UDPMode2DefaultTest\Readme.pdf](#)





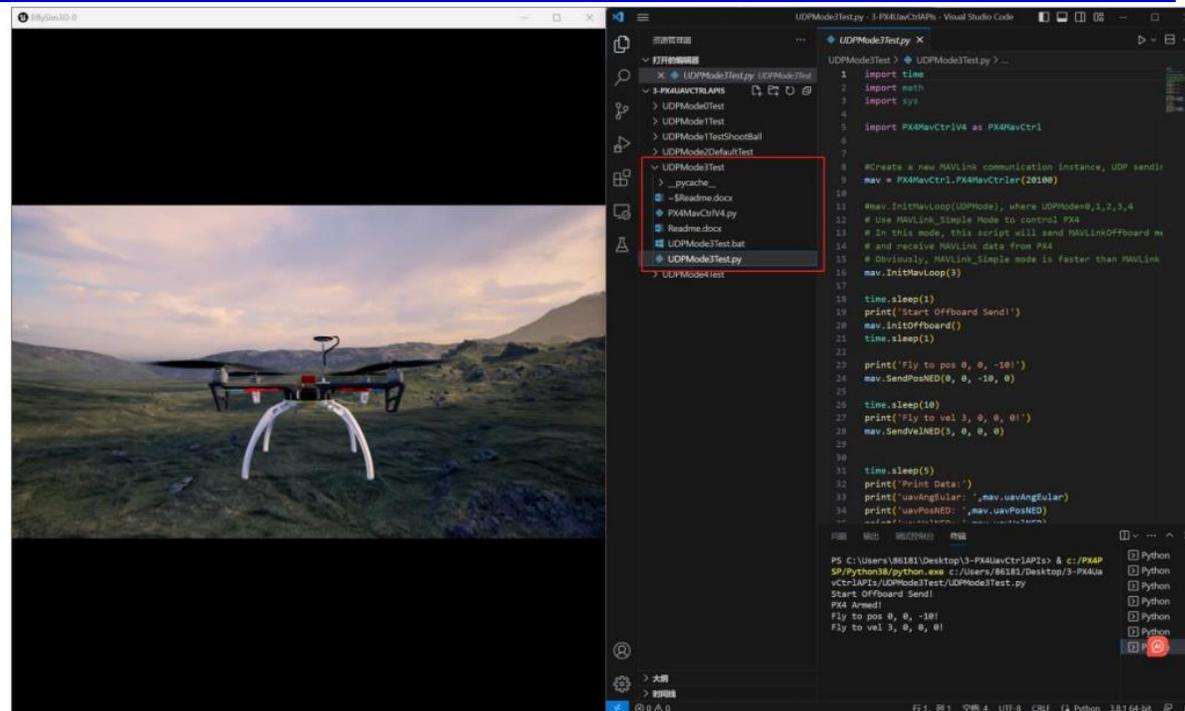
2.关键接口介绍

2.13 无人机通过 MAVLink_Simple 通信实验

通过使用平台提供的接口函数，通过 MAVLink_Simple 通信给飞机发送指令。

具体实验操作见文件

[0.ApiExps\13.UDPMode3Test\Readme.pdf](#)





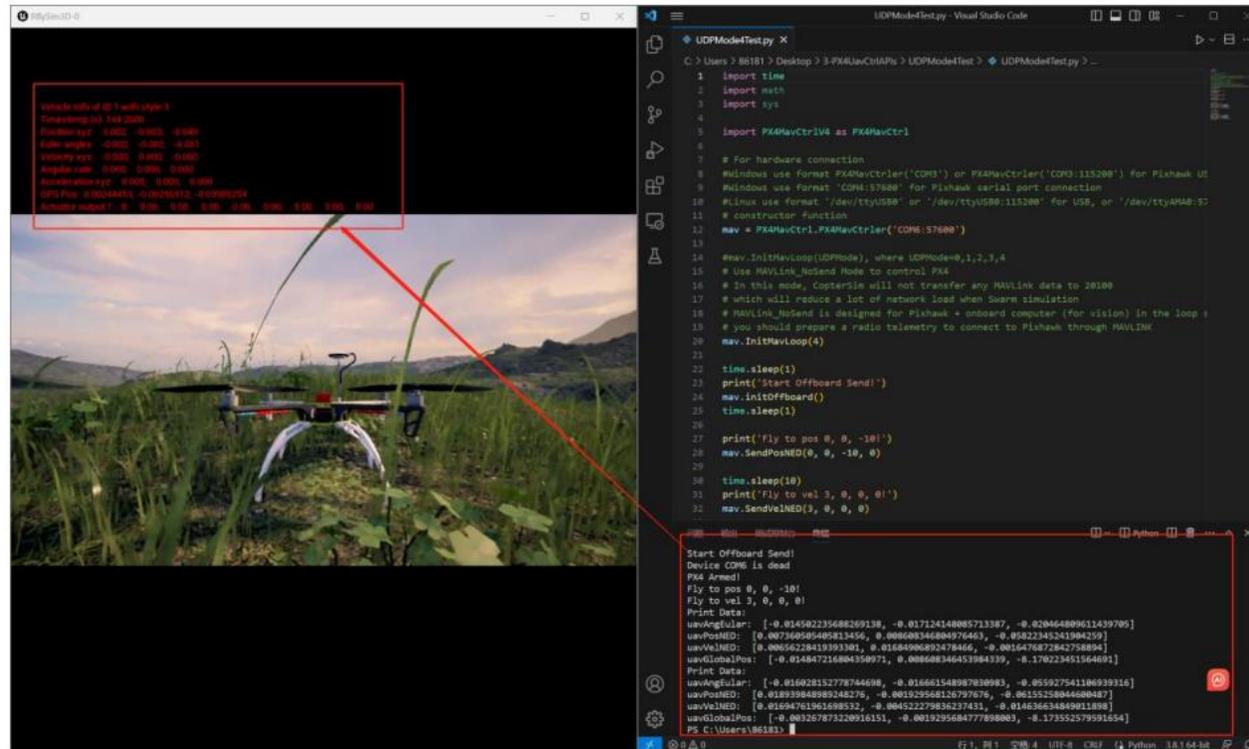
2.关键接口介绍

2.14 CopterSim-UDP 通信模式

通过使用平台提供的接口函数，
通过 MAVLink_NoSend模式对
CopterSim 给飞机发送指令。

具体实验操作见文件

[0.ApiExps\14.UDPMODE4Test\Readme.pdf](#)





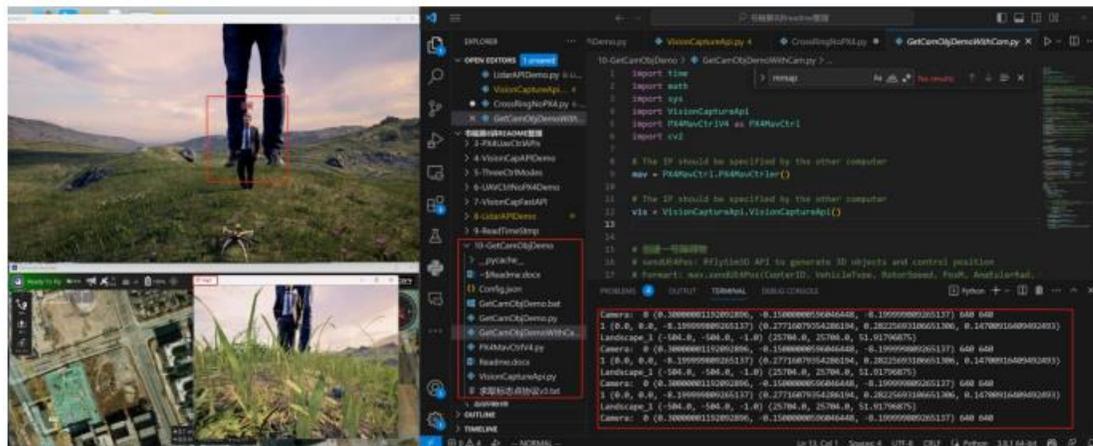
2.关键接口介绍

2.15飞机、物体、相机信息获取实验

通过 python 接口获取飞机、物体和相机的信息。

具体实验操作见文件

[0.ApiExps\15.CamObjGet\Readme.pdf](#)





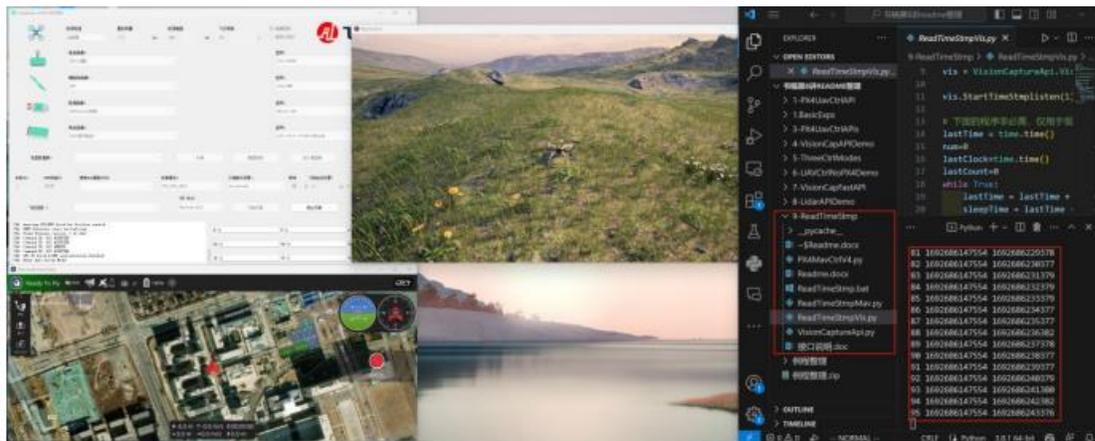
2.关键接口介绍

2.16时间戳获取实验

通过调用

StartTimeStmplisten(self,cpID=0)
接口，开始监听 20005 端口以获取 CopterID的 rflytimestamp 如果 $cpID == 0$ ，则只监听当前 CopterID。如果 $cpID > 0$ ，则监听所需 CopterID 的时间戳。然后调用 `mav.RflyTime.SysCurrentTime` 属性即为无人机的当前时间戳。具体实验操作见文件

[0.ApiExps\16.ReadTimeStmpGet\Readme.pdf](#)





大纲

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结

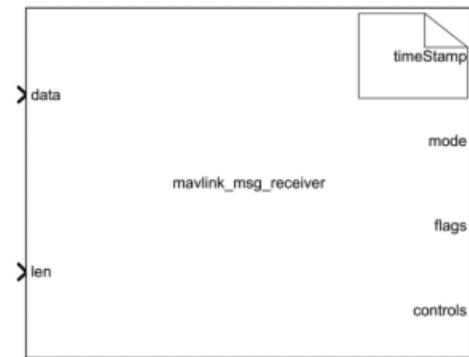
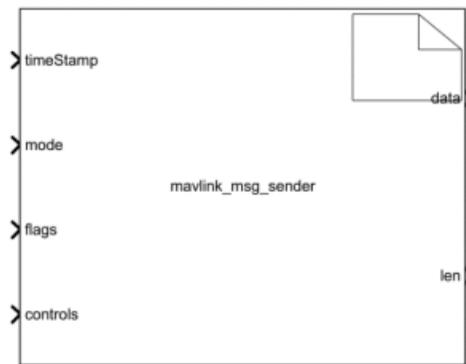


3.基础实验案例

3.1.1 MAVLink 模块封装实验

MAVLink (Micro Air Vehicle Link) 是一种用于小型无人载具的通信协议，于 2009 年首次发布。该协议广泛应用于地面站 (Ground Control Station, GCS) 与无人载具 (Unmanned vehicles) 之间的通信，同时也应用在载具上机载计算机与Pixhawk之间的内部通信中，协议以消息库的形式定义了参数传输的规则。MAVLink 协议支持无人固定翼飞行器、无人旋翼飞行器、无人车辆等多种载具。本实验将基于 Simulink 对 MAVLINK MSG ID HIL ACTUATOR CONTROLS 消息进行数据发送模块和数据解析模块两部分。

具体实验操作见文件 [1.BasicExps/e0 ExtAPIUsage/1.MavLinkPackSimulink/Readme.pdf](#)





3.基础实验案例

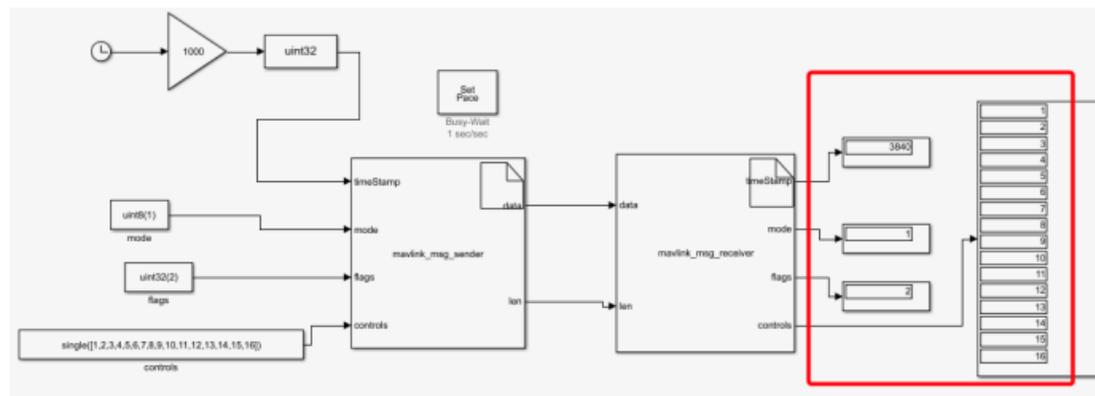
3.1.2 MAVLink 数据发接实验

MAVLink (Micro Air Vehicle Link) 是一种用于小型无人载具的通信协议, 于 2009 年首次发布。该协议广泛应用于地面站 (Ground Control Station, GCS) 与无人载具 (Unmanned vehicles) 之间的通信, 同时也应用在载具上机载计算机与 Pixhawk 之间的内部通信中, 协议以消息库的形式定义了参数传输的规则。MAVLink 协议支持无人固定翼飞行器、无人旋翼飞行器、无人车辆等多种载具。本实验将基于

“*\PX4PSP\RflySimAPIs\7.RflySimExtCtrl\1.BasicExps\e0_ExtAPIUsage\1.MavLinkPackSimulink”实验中建立的两个模块, 模拟发送 MAVLINK_MSG_ID_HIL_ACTUATOR_CONTROLS 消息并进行接收消息。

具体实验操作见文件

[1.BasicExps\e0_ExtAPIUsage\2.MavlinkCodeDecode\Readme.pdf](#)





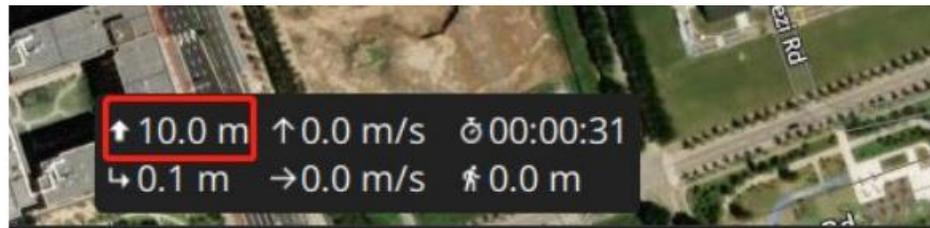
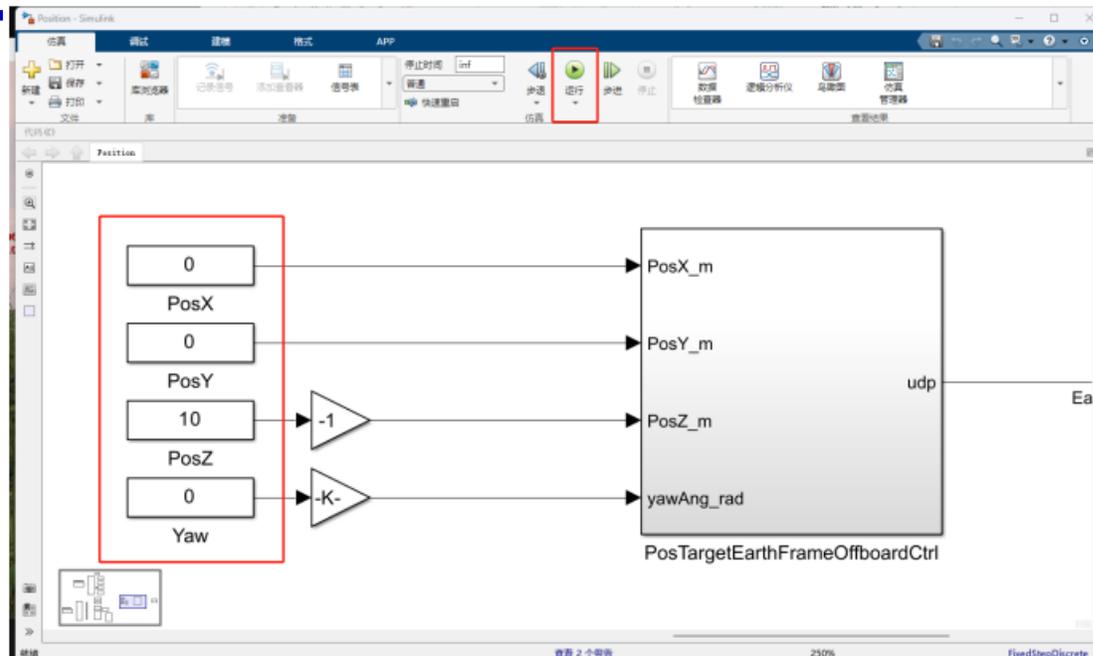
3.基础实验案例

3.2 Offboard 模式控制无人机位置控制实验

Offboard 模式是无人机的一种控制模式，通常给机载计算机或地面计算机（上位机）实时控制飞机的速度、位置、姿态等，可以把飞机当成一个整体对象，专注于顶层的视觉与集群算法开发。该实验主要讲位置控制实验。

具体实验操作见文件

1.BasicExps\e1_PosCtrl\readme.pdf



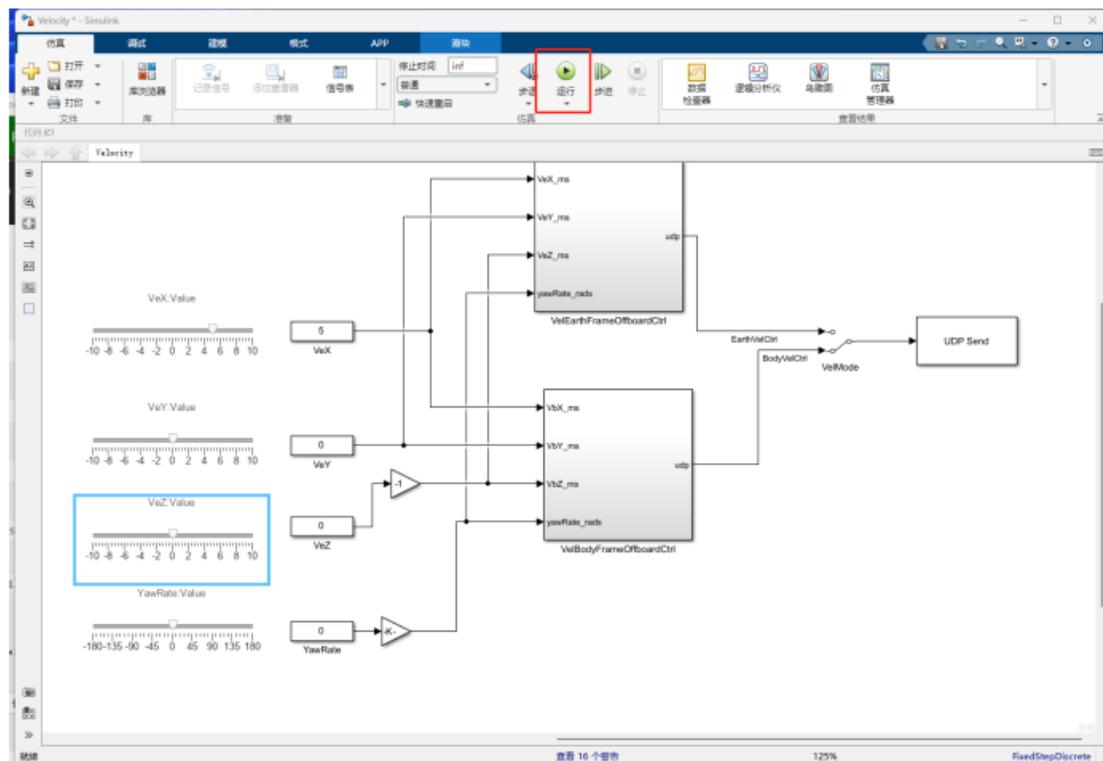


3.基础实验案例

3.3 Offboard 模式控制无人机速度控制实验

Offboard 模式是无人机的一种控制模式，通常给机载计算机或地面计算机（上位机）实时控制飞机的速度、位置、姿态等，可以把飞机当成一个整体对象，专注于顶层的视觉与集群算法开发。该实验主要讲速度控制实验。

具体实验操作见文件[1.BasicExps/e2_VelCtrl/readme.pdf](#)



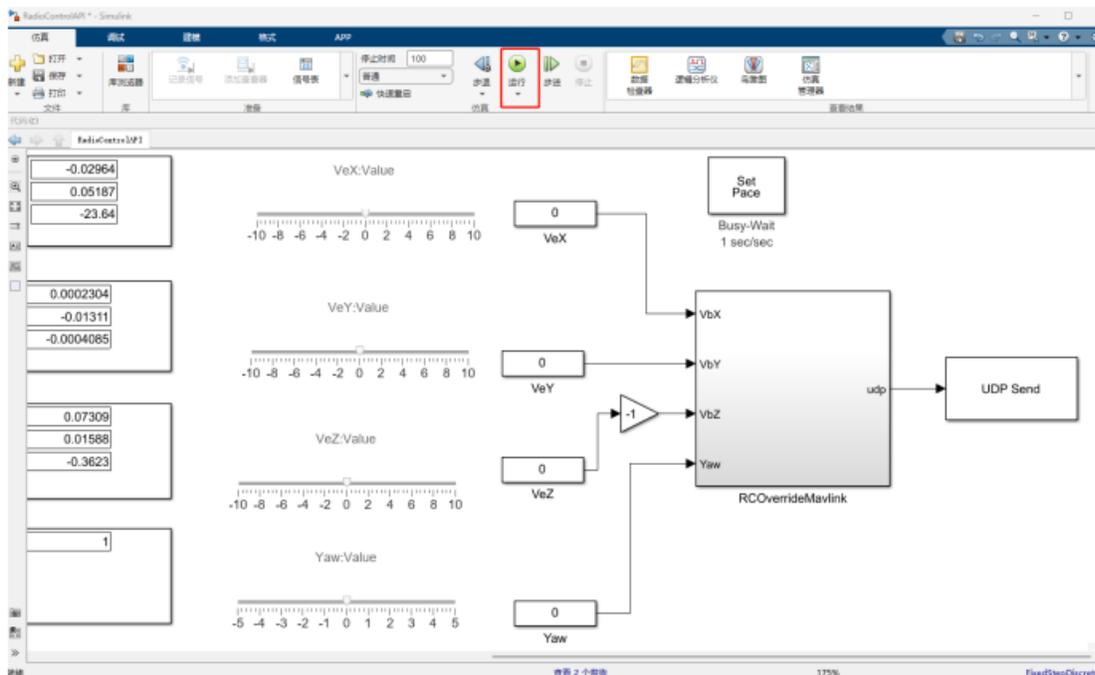


3.基础实验案例

3.4遥控器模式单机控制

遥控器模式是人为操作无人机的一种控制方式，在一些无人机特技表演中有较好的效果，本节使用的遥控器是“美国手”的操作方式，即左侧摇杆对应的油门与偏航控制量，而右侧摇杆对应滚转与俯仰。本次实验由控制器代替遥控器进行试验。

具体实验操作见文件[1.BasicExps/e3_RCCTRL/readme.pdf](#)





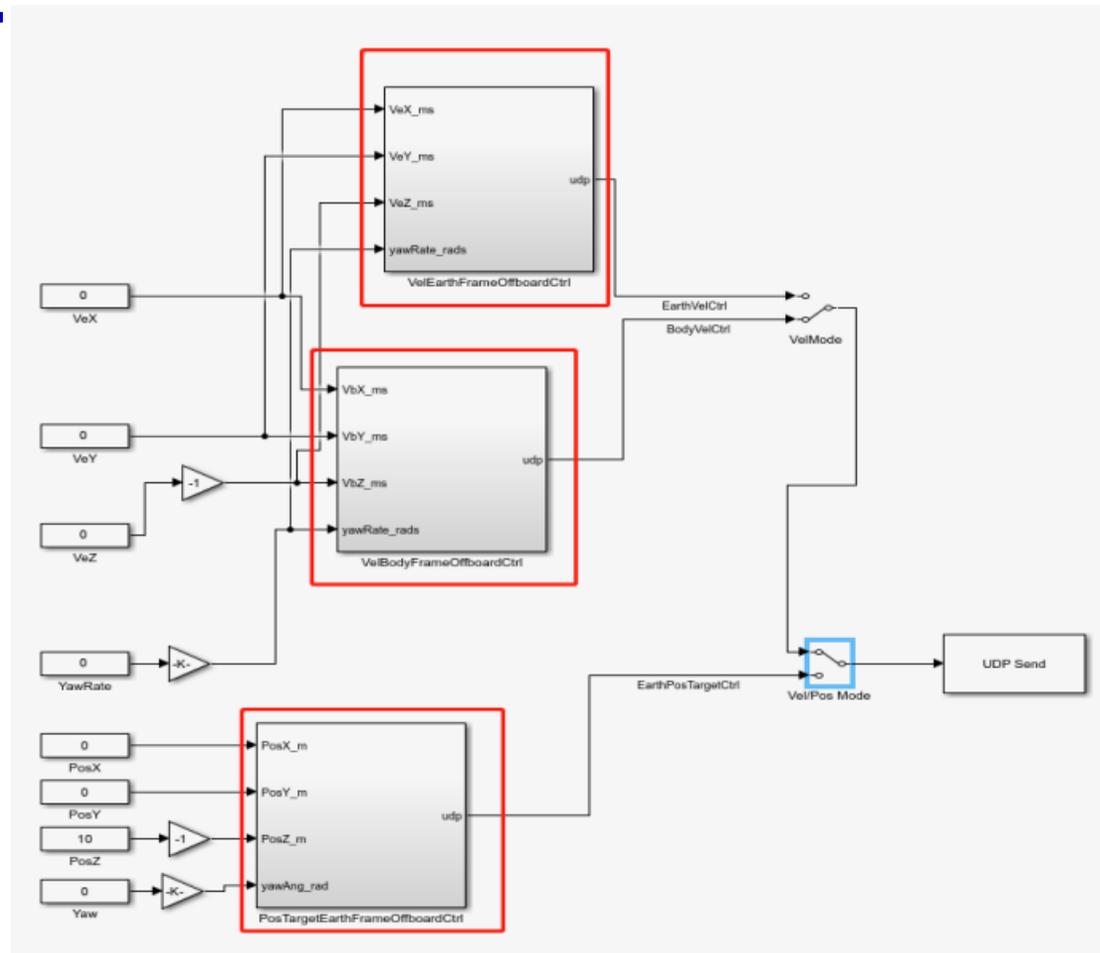
3.基础实验案例

3.5 Python-Offboard 单机控制实验

Offboard 模式是无人机的一种控制模式，通常给机载计算机或地面计算机（上位机）实时控制飞机的速度、位置、姿态等，可以把飞机当成一个整体对象，专注于顶层的视觉与集群算法开发。Python 控制无人机是通过编程语言与无人机进行通信，其基本原理是通过串口或网络连接无人机建立通信，以获取无人机的状态信息和执行命令。使用 PX4 的 OffboardAPI 来控制车辆预期速度和位置的演示程序。

具体实验操作见文件

1.BasicExps/e4_PyOffboardCtrl/readme.pdf

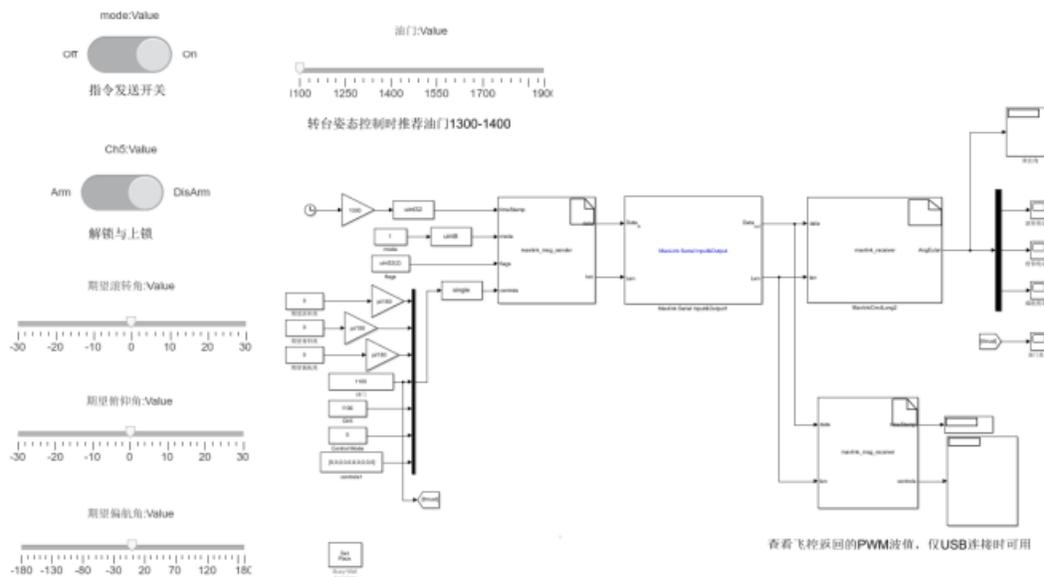




3.基础实验案例

3.6 单机控制台架实验

本实验通过在 MATLAB/Simulink 中搭建多旋翼飞行控制器，并通过 Simulink 发送控制指令，控制转台上的四旋翼无人机姿态。熟练掌握 MAVLINK 通信运用，熟练掌握四旋翼无人机姿态控制与参数整定。具体实验操作见文件[1.BasicExps\e5_RackFlyCtrl\Readme.pdf](#)



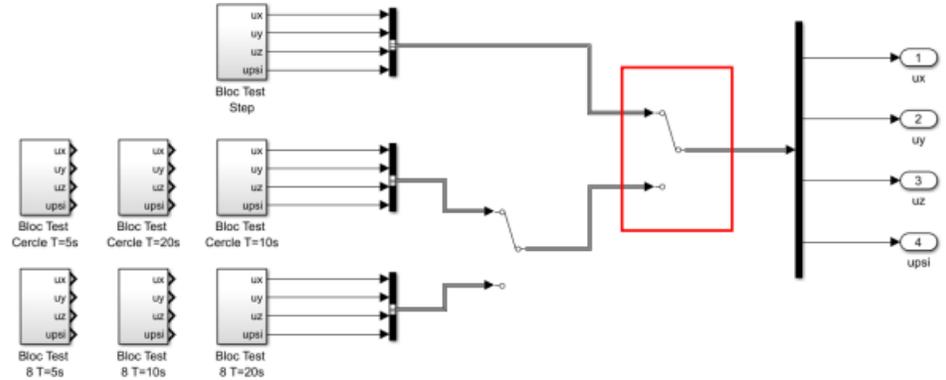


3.基础实验案例

3.7多旋翼路径跟踪控制器仿真实验

了解给定的多旋翼三通道线性化传递函数仿真模型和相应的轨迹跟踪控制器，进行轨迹跟踪。

具体实验操作见文件[1.Basic Exps\e6_PathTrackingCtrl\Readme.pdf](#)





大纲

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结

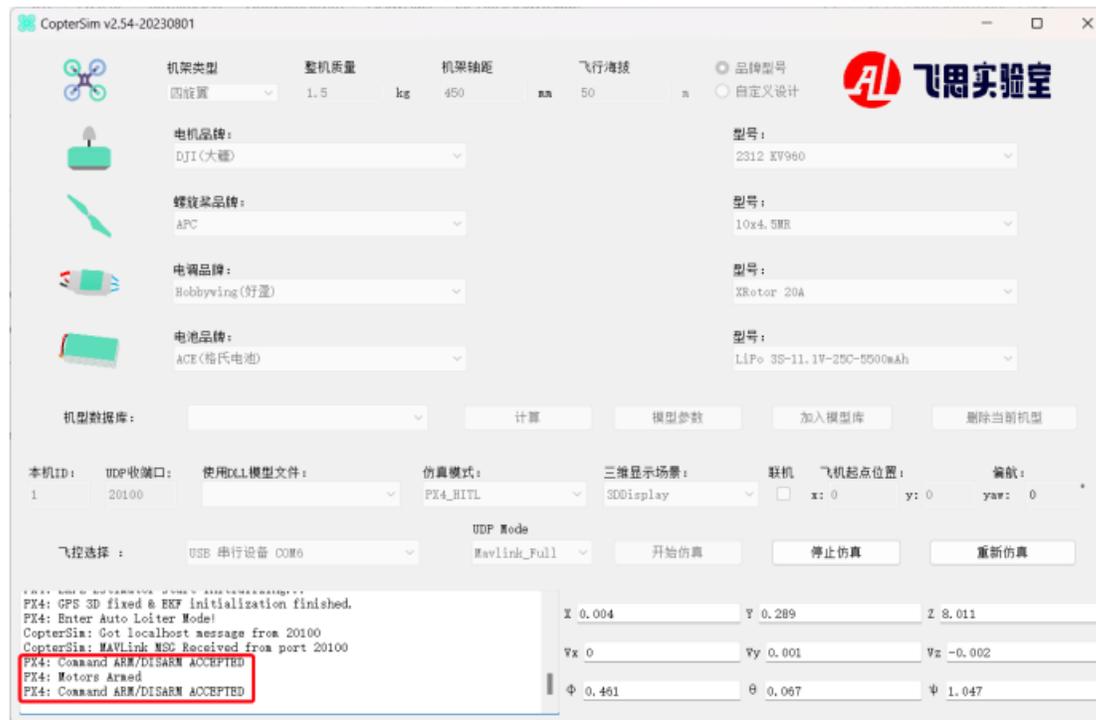


4.进阶案例实验

4.1 MAVSfun 解锁 HIL 实验

MAVLink (Micro Air Vehicle Link) 是一种用于小型无人载具的通信协议，于 2009 年首次发布。该协议广泛应用于地面站 (Ground Control Station, GCS) 与无人载具 (Unmanned vehicles) 之间的通信，同时也应用在载具上机载计算机与Pixhawk之间的内部通信中，协议以消息库的形式定义了参数传输的规则。MAVLink 协议支持无人固定翼飞行器、无人旋翼飞行器、无人车辆等多种载具。本实验将通过 CopterSim 软件在硬件在环仿真时，通过 MAVLink 封装模块何 UDP 的方式，在 CopterSim 软件中显示解锁信息。

具体实验操作见文件 [2.AdvExps/e1 ExtAPIAdvUsage\1.MavSfunTest Arm\Readme.pdf](#)



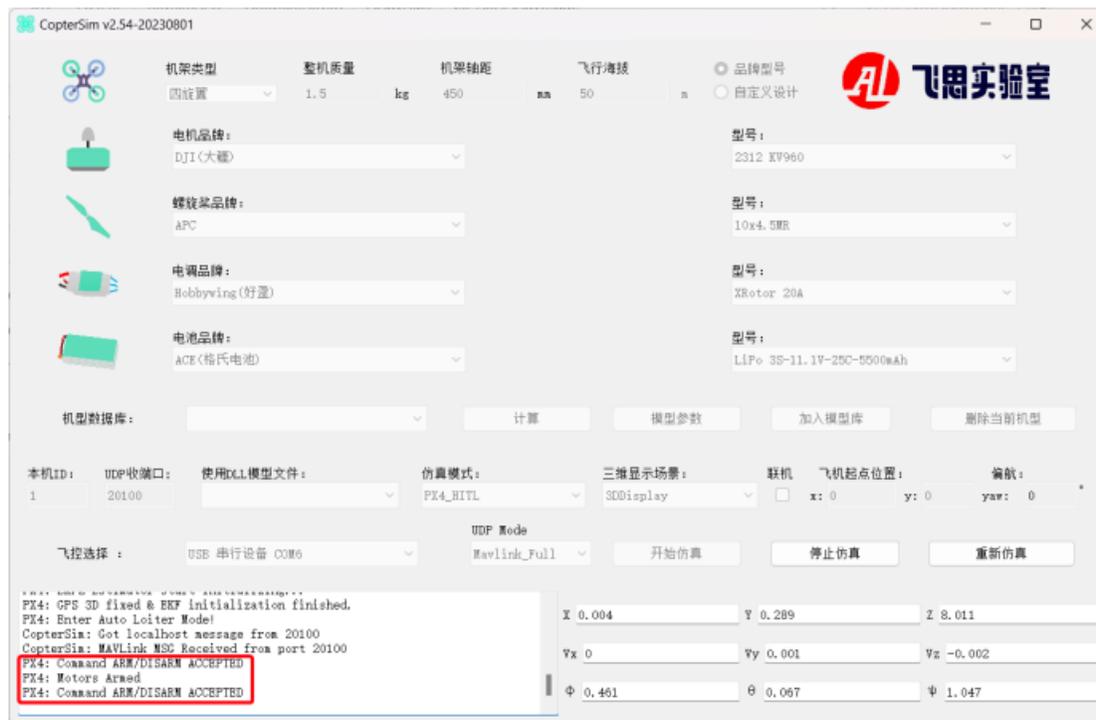


4.进阶案例实验

4.2 MAVLink 控制 HIL 实验

MAVLink (Micro Air Vehicle Link) 是一种用于小型无人载具的通信协议，于 2009 年首次发布。本实验将基于 CopterSim 软件在硬件在环仿真时，通过 MAVLink 封装模块 UDP 的方式，实现无人机姿态控制。

具体实验操作见文件 [2.AdvExps\e1_ExtAPIAdvUsage\2.MavSfunTest_Con\Readme.pdf](#)





大纲

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结



5.扩展案例

正在开发中



大纲

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结



7. 小结

- 本讲主要对无人机系统的外部控制与轨迹规划进行讲解，分为基础实验、进阶实验和扩展案例三部分，可以实现模型故障注入以及飞控源码注入教程。

如有疑问，请到<https://doc.rflysim.com/>查询更多信息。



RflySim更多教程



扫码咨询与交流



飞思RflySim技术交流群



谢谢！