
1、实验名称及目的

固定翼以固定俯仰角飞行实验：该例程通过平台固定翼控制接口控制固定翼俯仰角，让固定翼以固定 10° 的俯仰角前飞。

2. 实验原理

2.1. 软/硬件在环仿真（SIL/HIL）的实现[1][2]

从实现机制的角度分析，可将 RflySim 平台分为运动仿真模型、底层控制器、三维引擎、外部控制四部分。

- **运动仿真模型：**这是模拟飞行器运动的核心部分。在 RflySim 平台中，运动仿真模型是通过 MATLAB/Simulink 开发的，然后通过自动生成的 C++ 代码转化成 DLL（动态链接库）文件。在使用 RflySim 平台进行软硬件在环仿真时，会将 DLL 模型导入到 CopterSim，形成运动仿真模型。这个模型在仿真中负责生成飞行器的运动响应，它拥有多个输入输出接口与底层控制器、三维引擎、地面控制站和外部控制进行数据交互，具体数据链路、通信协议及通信端口号见 [API.pdf 中的通信接口部分](#)。
- **底层控制器：**在软/硬件在环仿真（SIL/HIL）中，真实的飞行控制硬件（如 PX4 飞行控制器）被集成到一个虚拟的飞行环境中。在软件在环仿真（SIL）中，底层控制器（通过 wsl 上的 PX4 仿真环境运行）通过网络通信与运动仿真模型交互数据。在硬件在环仿真（HIL）中，它（将 PX4 固件在真实的飞行控制器（即飞控）硬件上运行）则通过串口通信与运动仿真模型进行数据交互。飞控与 CopterSim 通过串口（硬件在环 HITL）或网络 TCP/UDP（软件在环 SITL）进行连接，使用 MAVLink 进行数据传输，实现控制闭环。
- **三维引擎：**这部分负责生成和处理仿真的视觉效果，提供仿真环境和模型的三维视图，使用户能够视觉上跟踪和分析飞行器的运动。CopterSim 发送飞机位姿、电机数据到三维引擎，实现可视化展示。
- **外部控制（offboard）：**从仿真系统外部对飞行器进行的控制，包括自动飞行路径规划、远程控制指令等。在平台例程中主要通过地面控制站（QGC）、MATLAB 和 Python 调用对应接口实现。

2.2. 通过外部控制接口（python）进行单机姿态控制

单机控制脚本 AircraftMathworksAttCtrl.py 中依次调用了 RflySim 平台飞机控制接口协议文件 PX4MavCtrlV4.py 中定义的以下接口函数

创建通信示例

```
mav1 = PX4MavCtrl.PX4MavCtrl(1)
```

创建一架飞机的通信示例

启用 Mavlink 消息监听循环

```
mav1.InitMavLoop()
```

配置 CopterSim 通信模式，该函数的参数定义如下：

```
def InitMavLoop(self,UDPMode=2):
    """ Initialize MAVLink listen loop from CopterSim
        0 and 1 for UDP_Full and UDP_Simple Modes, 2 and 3 for MAVLink_Full and
        MAVLink_Simple modes, 4 for MAVLink_NoSend
        The default mode is MAVLink_Full
    """
```

默认通信模式为 **Mavlink_Full**：Python 直接发送 MAVLink 消息给 CopterSim，再转发给 PX4，数据量较大适合单机控制；适合单机或少量飞机仿真，无人机数量小于 4；

设定航路点

```
n = 30
r = 400
missionPoints=[]
for i in range(n):
    angle = 2*math.pi*i/n
    x=r*math.sin(angle)
    y=r*math.cos(angle)
    missionPoints.append([x,y,-100])
```

用一组离散的点模拟圆形运动轨迹，并在循环中通过 `append` 方法逐个将相应的轨迹点存入目标点列表（`missionPoints`）。`missionPoints.append([x,y,-100])`表示在 `missionPoints` 列表的末尾添加一个新的列表`[x,y,-100]`。

根据欧拉公式：

$$e^{ix} = \cos x + i\sin x$$

这些点将在 x-y 平面上形成一个圆形轨迹。

飞行阶段

完成上述设置后，程序会通过检查一个 `flag` 变量的值来决定无人机应该执行哪些动作。

当 `flag == 0` 时，解锁飞机

解锁飞机

```
mav1.SendMavArm(True)
```

设定起飞目标点

```
targetPos=[200, 0, -100]
mav1.sendMavTakeOff(targetPos[0],targetPos[1],targetPos[2])
```

发送绝对的 GPS 坐标作为起飞目标点，使用 `sendMavTakeOffGPS` 命令，最后三位分别是经度、维度、和高度，会先从 `uavPosGPSHome` 向量中提取解锁 GPS 坐标，在此基础上用绝对坐标

当 `flag == 1` 时，无人机起飞和初始航迹

位置检测

```
curPos=mav1.uavPosNED
```

```
dis = math.sqrt((curPos[0]-targetPos[0])**2+(curPos[1]-targetPos[1])**2)
```

计算飞机当前位置和起飞目标位置的水平距离，用于判断是否到达目标位置，以开始下一阶段任务。

启动外部控制（offboard）

```
mav1.initOffboard()
```

使 px4 控制器进入外部控制模式，且以 30HZ 的频率发送 offboard 指令

航路寻迹模式

```
targetPos=missionPoints[flagI]
mav1.SendPosNED(targetPos[0], targetPos[1], targetPos[2])
```

会通过航路点索引 flagI 的值从 missionPoints 列表中读取相应的航点，并通过 SendPosNED 函数更新为下一个目标点。

当 `flag == 2` 时，无人机姿态和油门的控制

```
mav1.SendAttPX4([0,10,0],mav1.uavThrust)
```

这里设置无人机的俯仰角为 10 度，并保持油门值为悬停油门。这是为了改变无人机的飞行方向。

2.3. 通过外部控制接口（python）进行多机姿态控制

多机控制脚本 AircraftMathworksAttCtrl.py 脚本的实现逻辑与单机控制相同，只是需要创建 3 架飞机，再将相同的控制指令复制 3 份

创建通信示例

```
VehilceNum = 3
mav=[]
for i in range(VehilceNum):
    mav=mav+[PX4MavCtrl.PX4MavCtrler(1+i)]
```

创建 3 架飞机的通信示例

启用 Mavlink 消息监听循环

```
for i in range(VehilceNum):
    mav[i].InitMavLoop()
```

配置 3 架飞机的 CopterSim 通信模式

3. 实验效果

固定翼在软硬件在环仿真中，能按照期望俯仰角度飞行。

4. 文件目录

文件夹/文件名称	说明
AircraftMathworksAttCtrl.py	单个固定翼固定俯仰角飞行文件。
AircraftMathworksMavlinkHITLRun.bat	硬件在环仿真批处理文件。
AircraftMathworksMavlinkSITLRun.bat	软件在环仿真批处理文件。
AircraftMathworksAttCtrl3.py	多个固定翼固定俯仰角飞行文件。

PX4MavCtrlV4.py	程序调用接口文件。
AircraftMathworks.dll	固定翼无人机 DLL 模型文件

5. 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	PX4 飞控 ^②	1
3	Python3.11	数据线	1

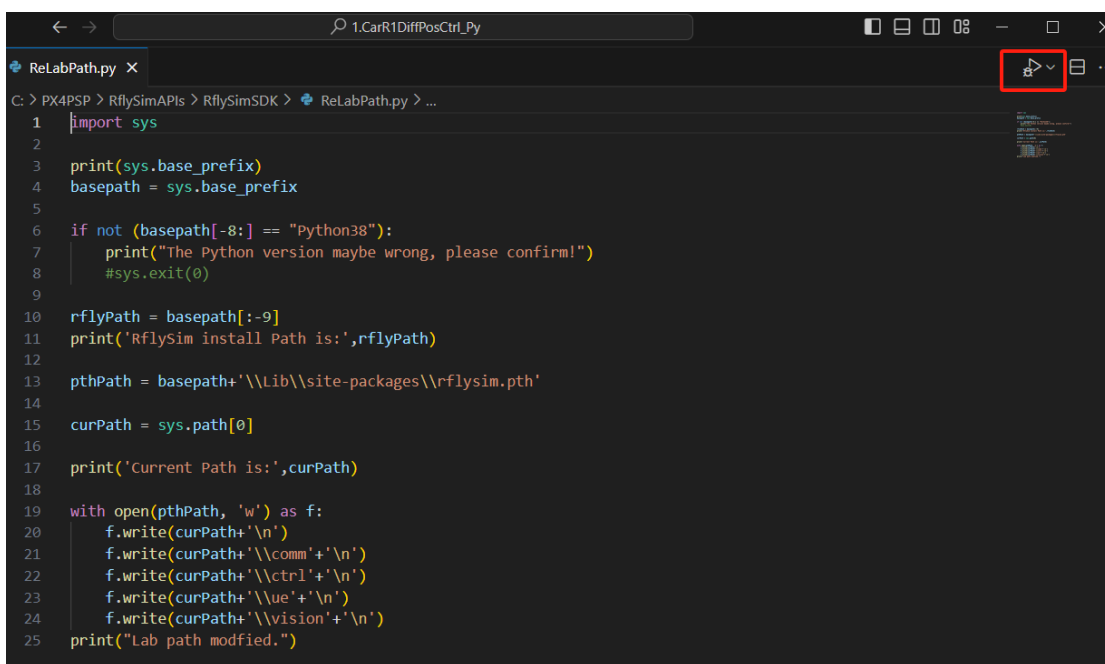
① 推荐配置请见：<https://doc.rflysim.com>

② 须保证平台安装时的编译命令为：px4_fmu-v6c_default，固件版本为：1.13.3。其他配套飞控请见：<http://doc.rflysim.com/hardware.html>

6. 实验步骤

6.1. Python 库文件部署

以 VsCode 打开 “C:\PX4PSP\RflySimAPIs\RflySimSDK\ReLabPath.py”，并运行。



```

1  import sys
2
3  print(sys.base_prefix)
4  basepath = sys.base_prefix
5
6  if not (basepath[-8:] == "Python38"):
7      print("The Python version maybe wrong, please confirm!")
8      #sys.exit(0)
9
10 rflyPath = basepath[:-9]
11 print('Rflysim install Path is:', rflyPath)
12
13 pthPath = basepath+'\\Lib\\site-packages\\rflysim.pth'
14
15 curPath = sys.path[0]
16
17 print('Current Path is:', curPath)
18
19 with open(pthPath, 'w') as f:
20     f.write(curPath+'\n')
21     f.write(curPath+'\\comm'+'\n')
22     f.write(curPath+'\\ctrl'+'\n')
23     f.write(curPath+'\\ue'+'\n')
24     f.write(curPath+'\\vision'+'\n')
25 print("Lab path modified.")

```

完成 Python 公共库环境部署。

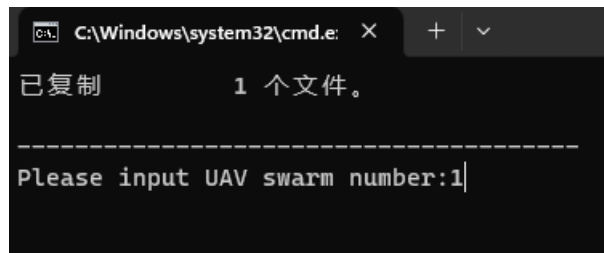
6.2. 软件在环仿真

6.2.1. 单架固定翼仿真

Step 1:

右键以管理员身份运行“AircraftMathworksMavlinkSITLRun.bat”批处理文件，在弹出的

终端窗口中输入 1，启动 1 架飞机的软件在环仿真。



```
C:\Windows\system32\cmd.e: X + v
已复制      1 个文件。
-----
Please input UAV swarm number:1|
```

Step 2:

完成初始化。

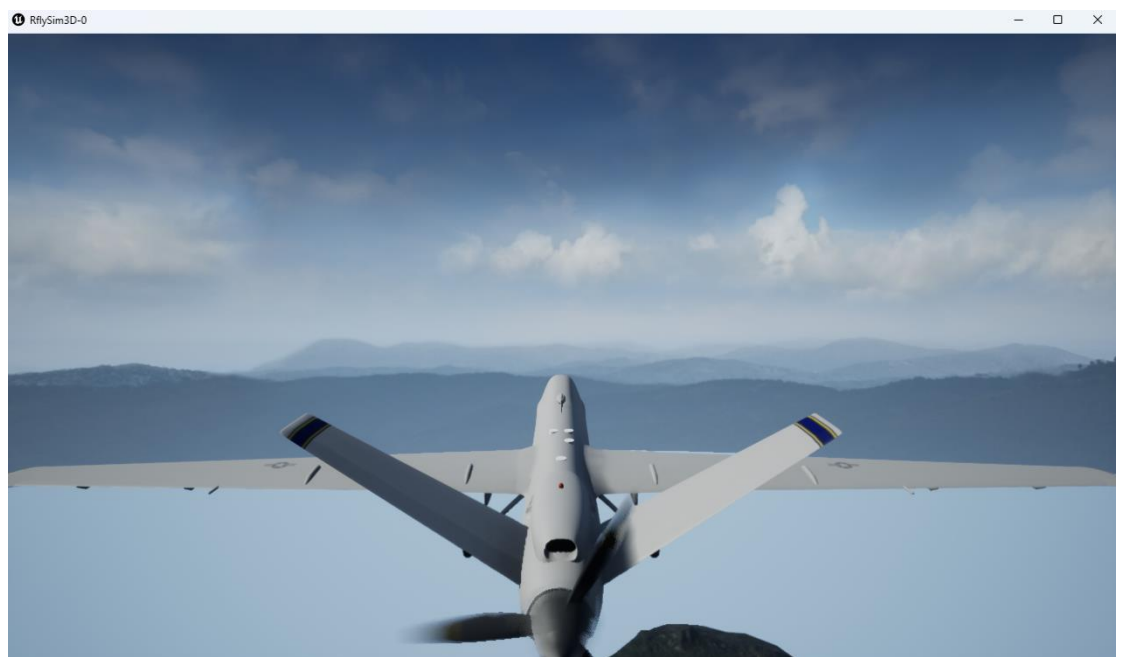


Step 3:

打开 AircraftMathworksAttCtrl.py 文件并运行。

```
AircraftMathworksAttCtrl.py 9- X
D:\OneDrive\桌面\Python-固定角> AircraftMathworksAttCtrl.py ...
14
15 # Init MAVLink data receiving loop
16 mav1.InitMavLoop()
17 #mav2.InitMavLoop(), ...
18
19 lastTime = time.time()
20 startTime = time.time()
21 # time interval of the timer
22 timeInterval = 1/30.0 #there is 0.0333s (30Hz)
23
24 # 圆形轨迹
25 n = 30
26 r = 400
27 missionPoints=[]
28 for i in range(n):
29     angle = 2*math.pi*i/n
30     x=r*math.sin(angle)
31     y=r*math.cos(angle)
32     missionPoints.append([x,y,-100])
33
34 # flags for vehicle 1
35 flag = 0
36 flagTime=startTime
37 flag1=0
38
39 # flags for vehicle 2

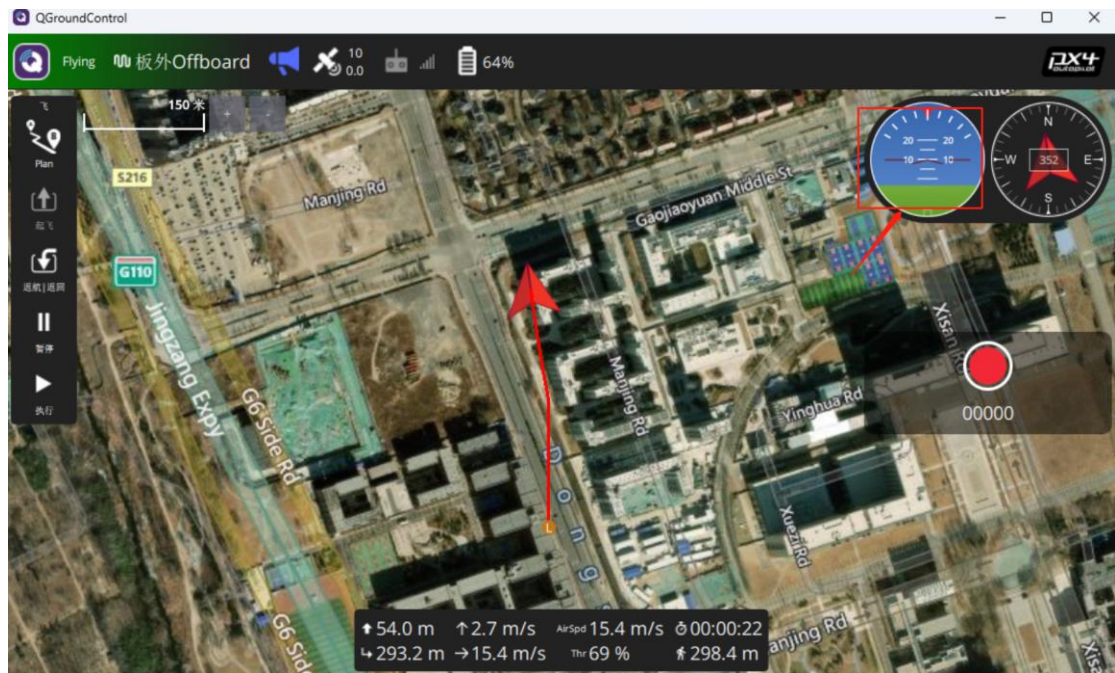
PROBLEMS 64 OUTPUT TERMINAL DEBUG CONSOLE Python Debug Console
Windows PowerShell
版权所有 (C) Microsoft Corporation. 保留所有权利。
安装最新的 PowerShell, 了解新功能和改进! https://aka.ms/PSWindows
PS D:\OneDrive\桌面\Python-固定角> & 'E:\python38\python.exe' 'c:\Users\admin\.vscode\extensions\ms-python.python-2023.12.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher
' -c 'D:\OneDrive\桌面\Python-固定角\AircraftMathworksAttCtrl.py'
S5: Arm the drone
Arm the drone!
PX4 Armed!
开始起飞
```



可从 CopterSim 右下角看到固定翼按期望指令以 10° 俯仰角前飞。

X	251.216	Y	-190.854	Z	93.031
Vx	16.696	Vy	-2.263	Vz	3.159
ϕ	0.01	θ	10.808	ψ	-7.576

也可从 QGC 罗盘处看到固定翼按期望指令以 10° 俯仰角前飞。



6.2.2. 多架固定翼仿真

Step 1:

多架固定翼仿真在运行“AircraftMathworksMavlinkSITLRun.bat”文件后，在弹出的终端窗口中输入 3。

```
C:\Windows\System32\cmd.exe
已复制 1 个文件。
-----
Please input UAV swarm number:3
```

Step 2:

完成初始化。



Step 3:

三架固定翼初始化完成后，打开 AircraftMathworksAttCtrl3.py 文件并运行

```

AircraftMathworksAttCtrl3.py 9+ X
z: > 工程师 > 2022-叶颖鑫-模型组 > 4-项目文件 > 5-例程整理 > 童俊豪 > PythonCtrlAPI > 固定翼航向角控制 > AircraftMathworksAttCtrl3.py > ...
1 # import required libraries
2 import time
3 import math
4
5 # import RflySim APIs
6 import PX4MavCtrlV4 as PX4MavCtrl
7
8 VehicNum = 3
9 mav=[]
10 # Create MAVLink control API instance
11 for i in range(VehicNum):
12     mav=mav+[PX4MavCtrl.PX4MavCtrl(20100+i*2)]
13
14 # Init MAVLink data receiving loop
15 for i in range(VehicNum):
16     mav[i].InitMavLoop()
17
18 lastTime = time.time()
19 startTime = time.time()
20 # time interval of the timer
21 timeInterval = 1/30.0 #here is 0.0333s (30Hz)
22
23 # 圆形轨迹
24 n = 30
25 r = 400
26 missionPoints=[]

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE Python Debug Console

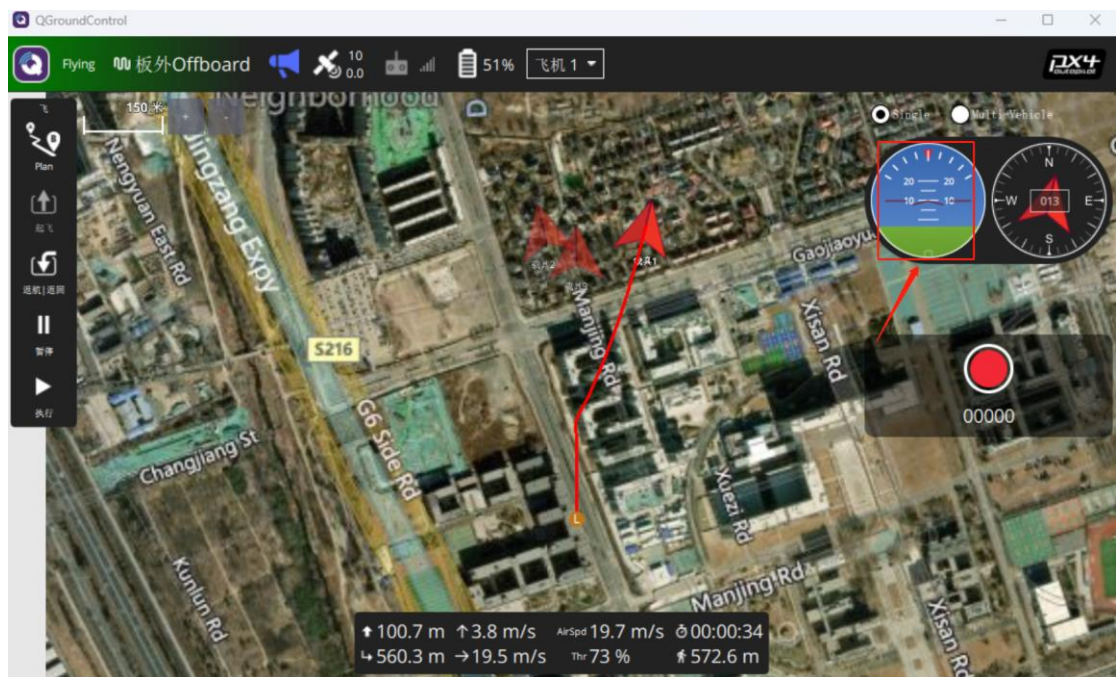
```

0.6986083984375
0.6986083984375
0.6986083984375
0.6986083984375
0.6986083984375
PS D:\OneDrive\桌面\Python-固定角> z: cd 'z:\工程师\2022-叶颖鑫-模型组\4-项目文件\5-例程整理\童俊豪\PythonCtrlAPI\固定翼航向角控制'; & 'E:\python38\python.exe' 'c:\Users\admin\python\python-2023.12.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '55185' '-.' 'Z:\工程师\2022-叶颖鑫-模型组\4-项目文件\5-例程整理\童俊豪\PythonCtrlAPI\固定翼航向角控制\AircraftMathworksAttCtrl3.py'
55, Arm the drone
开始起飞

```




可在 QGC 上方切换飞机,从 QGC 罗盘处看到每一架固定翼按期望指令以 10° 俯仰角前飞。



或观察每个 CopterSim 右下角看出固定翼按期望指令以 10° 俯仰角前飞。

X	251.216	Y	-190.854	Z	93.031
Vx	16.696	Vy	-2.263	Vz	3.159
ϕ	0.01	θ	10.808	ψ	-7.576

6.3. 硬件在环仿真

Step 1:

按下图所示将飞控与计算机链接。



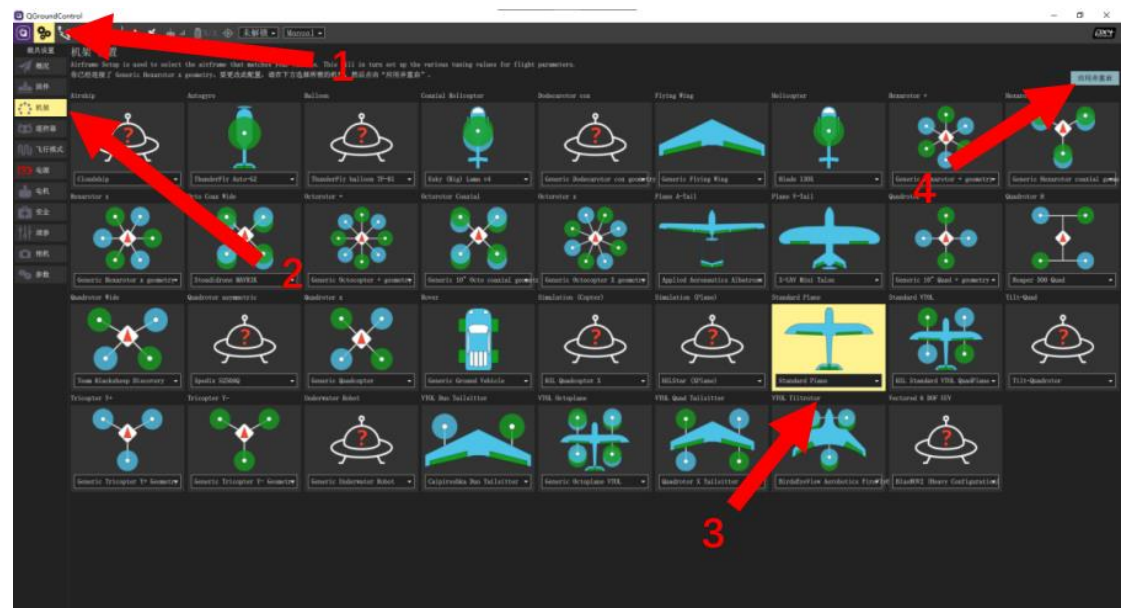
Step 2:

在 Rflytools 文件夹中打开 QGC 地面站。

3DDisplay	2023/7/27 15:02	快捷方式	1 KB
CopterSim	2023/7/27 15:02	快捷方式	1 KB
FlightGear-F450	2023/7/27 15:02	快捷方式	2 KB
HITLRun	2023/7/27 15:02	快捷方式	2 KB
Python38Env	2023/7/27 15:02	快捷方式	2 KB
QGroundControl	2023/7/27 15:02	快捷方式	1 KB
RflySim3D	2023/7/27 15:02	快捷方式	1 KB
RflySimAPIs	2023/7/27 15:02	快捷方式	1 KB
RflySimUE5	2023/7/27 15:02	快捷方式	1 KB
SITLRun	2023/7/27 15:02	快捷方式	2 KB
Win10WSL	2023/7/27 15:02	快捷方式	2 KB

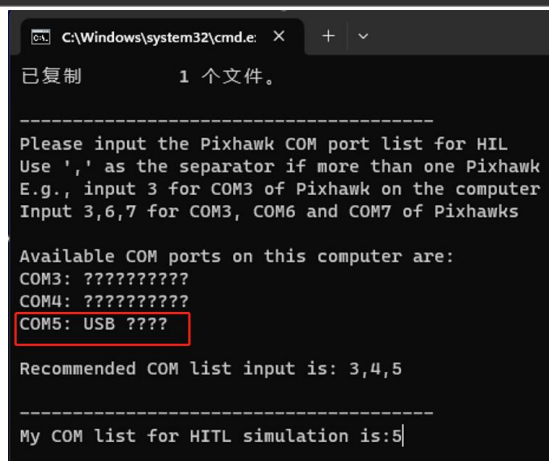
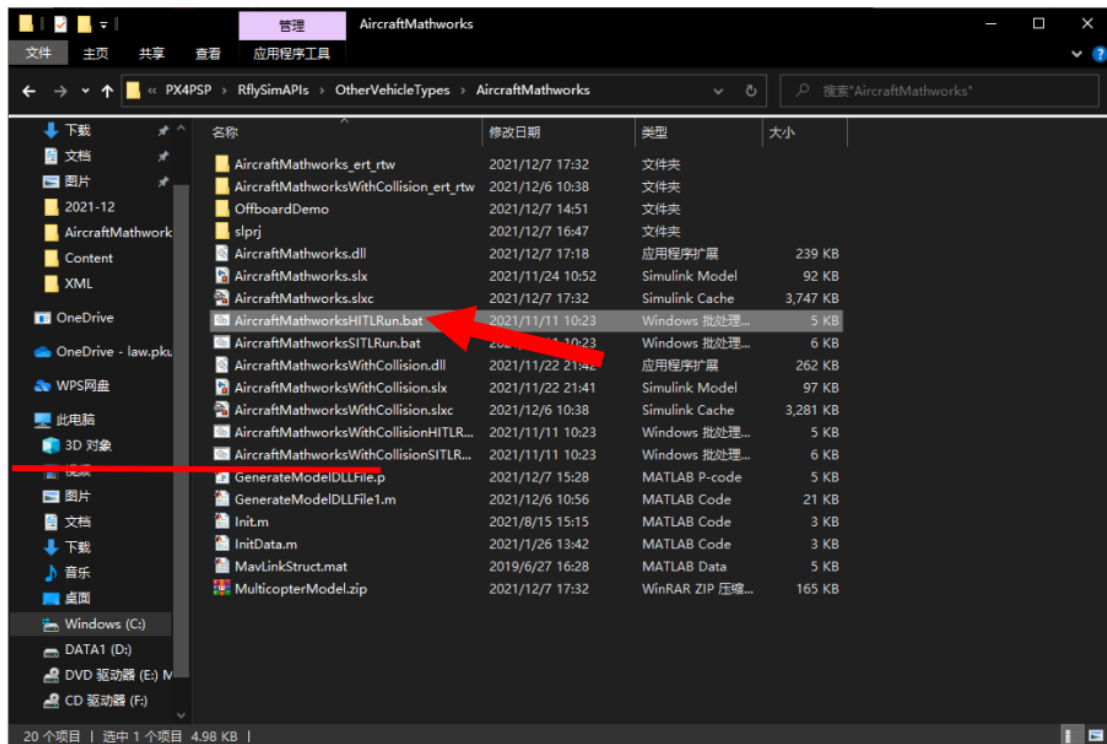
Step 3:

在机架界面设置机架型号为“Standard Plane”，设置完毕后点击右侧“应用并重启”。



Step 4:

在“安全”界面，选择“HITL enabled”启动硬件在环仿真，之后在概況界面中确认配置完成后，重新插拔飞控完成设置。



Step 6:

之后测试步骤与 5.1.1 中 Step2 到 Step3 与以及 5.1.2 中的 Step2 到 Step3 相同,运行后可在 QGC 与 RflySim3D 中观测运行状态与角度。

注意事项：在固定翼的 offboard 控制中，用到如下控制接口：

- 1) SendMavTakeOff: 起飞指令。
- 2) SendAttPX4: 设置姿态角。

7. 参考资料

- [1]. DLL/SO 模型与通信接口 [..\..\API.pdf](#)
- [2]. 外部控制接口 [..\..\API.pdf](#)
- [3].

8. 常见问题

Q1.

A1.