

通过网口修改载具机架实验

1. 实验目的

本实验旨在通过网络接口修改PX4飞控的载具机架类型，实现在不使用地面站软件的情况下动态切换无人机机型的功能。通过本实验，学习者将掌握以下技能：

1. 理解PX4飞控中SYS_AUTOSTART参数的作用及其与不同机架类型的对应关系；
2. 掌握通过MAVLink协议向飞控发送NSH(NuttX Shell)命令的方法；
3. 学习如何使用Python脚本自动化配置飞控参数并实现飞控重启；
4. 熟悉硬件在环(Hardware-In-The-Loop, HITL)仿真环境下不同类型无人机的配置与运行方式；
5. 提高对无人机控制系统参数配置和自动化操作的实际应用能力。

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\1.RflySimIntro\2.AdvExps\e9.ConfigTools\e3.FCUEth_ModifyParam](#)

- [NetModifyAirframeHIL.py](#)：通过网络接口修改PX4飞控载具机架类型的Python脚本。
- [HITLRunPX4Net_FW.bat](#)：启动固定翼硬件在环仿真的批处理脚本。
- [HITLRunPX4Net_MQ.bat](#)：启动多旋翼硬件在环仿真的批处理脚本。
- [Python38Run.bat](#)：启动Python 3.8环境的批处理脚本。
- [FixWingPW.plan](#)：固定翼无人机航点飞行任务计划文件。

4. 实验内容或步骤

本实验的主要内容是通过运行Python脚本，将PX4飞控的SYS_AUTOSTART参数从默认的四旋翼配置修改为固定翼配置(2100)，并通过硬件在环仿真验证配置效果。

4.1 步骤1：实验前准备

本实验是基于

[RflySim安装目录]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e9.ConfigTools\e1.FCUETH_NetConfig

实验进行，在进行本实现之前请务必按照该实验连接飞控等硬件，按照实验步骤完成该实验。

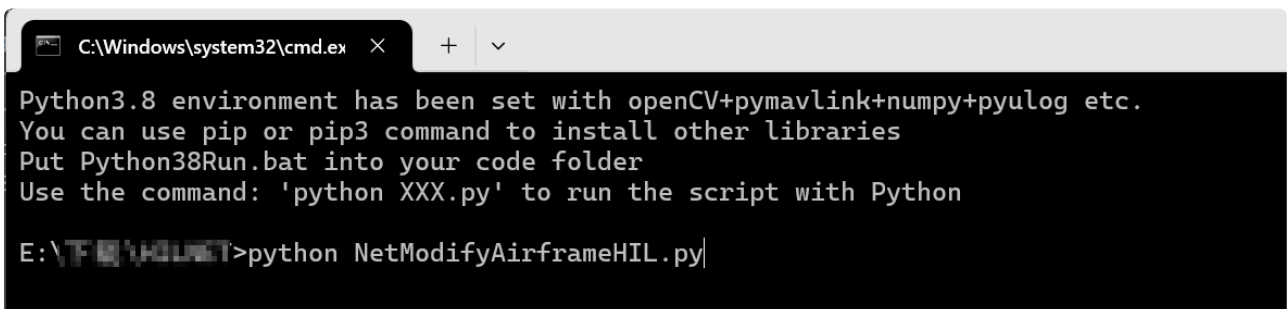
4.2 步骤2：机架替换

按照

[RflySim安装目录]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e9.ConfigTools\e1.FCUETH_NetConfig

实验中的配置，可直接双击运行HITLRunPX4Net_MQ.bat脚本启动仿真，开启四旋翼在环仿真。

双击打开Python38Run.bat脚本运行 `python NetModifyAirframeHIL.py`。



```
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:\下\PROJECT>python NetModifyAirframeHIL.py
```

等待运行完成后，会自动重启飞控并返回修改的机架类型，如下图所示。

```

心跳收到, 系统 ID=1
nsh> param set SYS_AUTOSTART 2100
param set SYS_AUTOSTART 2100
nsh> 发送指令: param set SYS_AUTOSTART 2100
nsh> reboot
param set SYS_AUTOSTART 2100
nsh> reboot 已发送, 飞控即将重启
⌚ 等待飞控重新上线...
✅ 心跳恢复 sysid=1 time=6.5s
⌚ 重新进入 nsh 控制台...
nsh> param show SYS_AUTOSTART

NuttShell (NSH) NuttX-11.0.0
nsh> param show SYS_AUTOSTART
Symbol
NuttShell (NSH) NuttX-11.0.0
nsh> param show SYS_AUTOSTART
Symbols: x = used, + = saved, * = unsaved
x + SYS_AUTOSTART [640,1403] : 2100 x = used, + = saved, * = unsaved
x + SYS_AUTOSTART [640,1403] : 2100

674/1744 parameters used.
nsh>

```

上图所示的2100，在PX4中对应的即为固定翼无人机机架。

4.3 步骤3：运行仿真

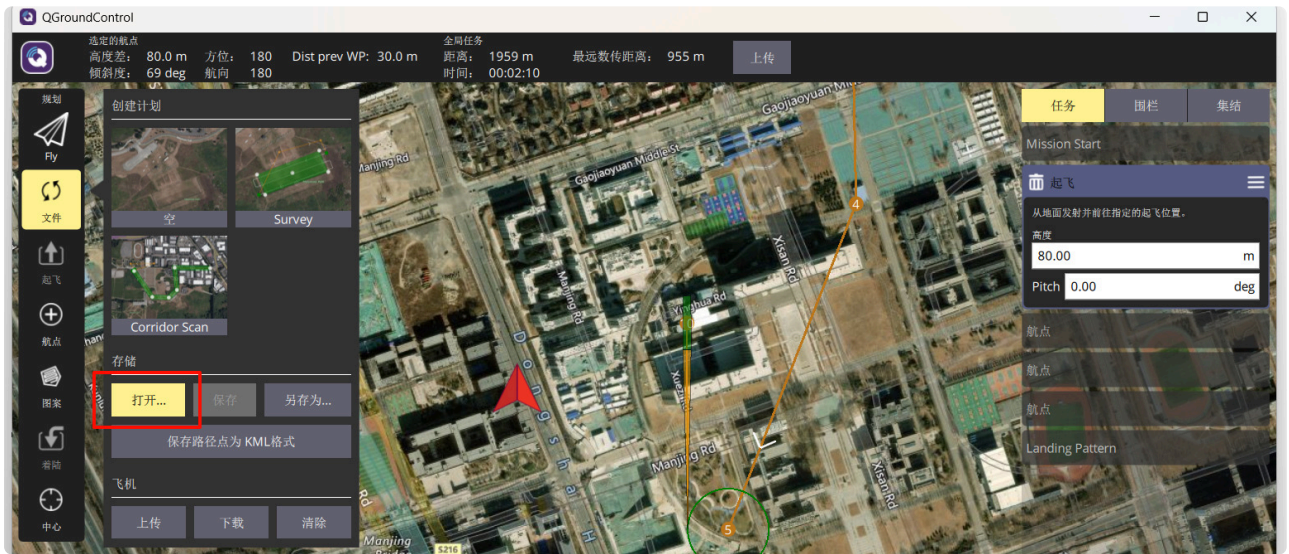
运行 `HITLRunPX4Net_MQ.bat` 脚本启动仿真，开启固定翼在环仿真。输入飞机的数量：
1。即可开启1架固定翼无人机基于网口的硬件在环仿真。

```
Please input UAV swarm number:1_
```

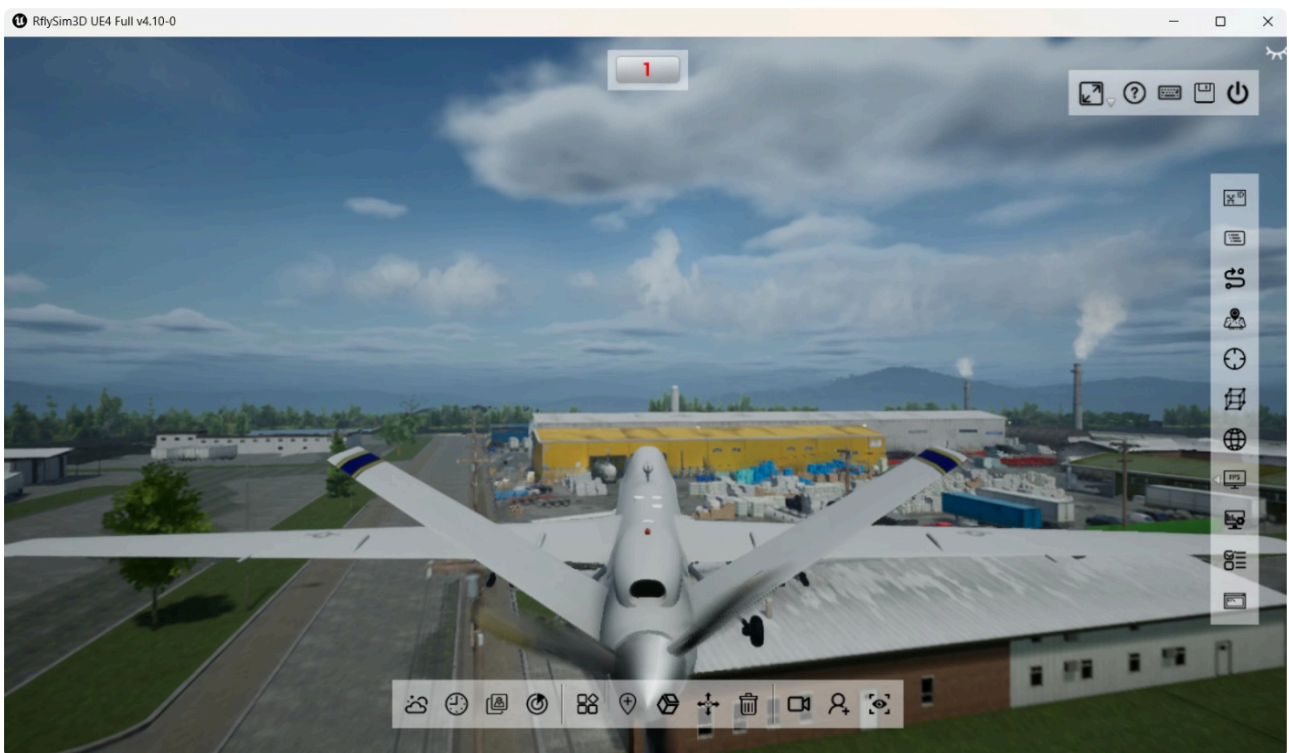
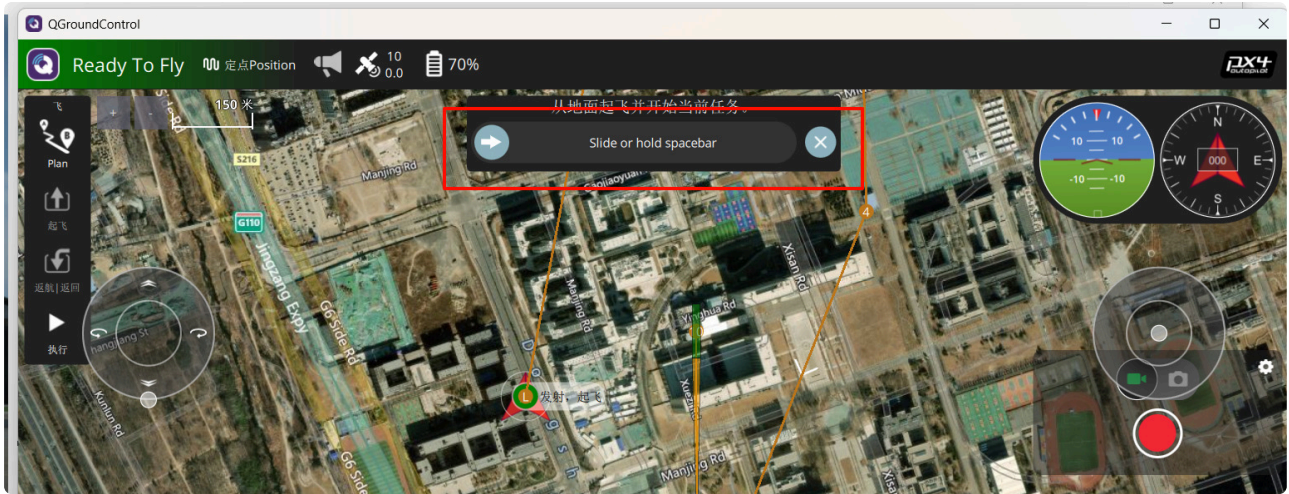
RflySim自动启动RflySim3D、CopterSim、QGroundControl软件，等待RflySim3D左上角显示：CopterSim/PX4 EKF 3DFixed:1/1。或者初始化CopterSim左下角对话框中显示：PX4: GPS 3D fixed & EKF initialized.则表示初始化成功。



依次点击左上角的"Plan"->"文件"按钮，点击"打开"按钮，加载本实验路径下的"FixWingPW.plan"航点文件，点击上侧的"上传"按钮即可上传航点信息。



返回后开始执行任务即可飞行。



5. 关键知识点

关键知识点1：MAVLink协议与飞控通信

MAVLink是一种轻量级的消息传输协议，专为资源受限的无人系统设计。在本实验中，我们通过MAVLink协议与PX4飞控建立UDP网络连接，并发送特定命令来修改飞控参数。

[NetModifyAirframeHIL.py](#) 脚本使用pymavlink库创建MAVLink连接，通过serial_control_send方法发送NSH命令到飞控，从而实现远程配置。该方法避免了传统通过QGroundControl等地面站软件修改参数的方式，实现了参数配置的自动化。

关键知识点2：PX4 SYS_AUTOSTART参数机制

SYS_AUTOSTART参数定义了PX4飞控启动时加载的机架配置文件编号。不同的编号对应不同的无人机类型和配置，例如四旋翼、固定翼、rover等。在本实验中，我们将该参数设置为2100，对应固定翼(Fixed Wing)机型。通过修改此参数并重启飞控，我们可以实现无人机类型的动态切换。这种方式在多机型无人机开发和测试中具有重要应用价值。

关键知识点3：Python自动化脚本实现原理

[NetModifyAirframeHIL.py](#)脚本主要包含以下几个核心功能模块：

1. 使用mavutil.mavlink_connection建立与飞控的UDP连接
2. 通过nsh_write函数封装NSH命令发送逻辑
3. 利用SERIAL_CONTROL消息类型实现命令传输和回显接收
4. 发送reboot命令重启飞控使配置生效
5. 等待飞控重启后验证参数修改结果

脚本通过模拟人工在NSH命令行中输入命令的方式，实现了对飞控参数的修改和验证，提高了操作效率和准确性。

6.参考资料

1. [MAVLink协议官方文档](#)
2. [PX4用户指南](#)
3. [pymavlink库文档](#)
4. [RflySim官方文档](#)

7.常见问题

Q1：飞控未上线或无法连接

A1：检查飞控是否正确连接并开启，确认网络连接正常。确保UDP端口6001未被其他程序占用，检查防火墙设置是否阻止了通信。

Q2: 参数修改后未生效

A2: 确认脚本执行过程中是否成功发送reboot命令，飞控是否完成重启。重启后可以手动检查SYS_AUTOSTART参数值是否正确设置为2100。

Q3: 固定翼仿真无法正常启动

A3: 检查 `HITLRunPX4Net_MQ.bat` 脚本中的配置是否正确，确认飞控已正确加载固定翼配置。查看RflySim3D和CopterSim日志以获取更多错误信息。

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