

健康评估基础实验

1. 实验目的

收集数据方便对训练模型进行训练及验证。通过仿真对无人机运行过程中的数据进行收集，掌握健康评估算法的实现方法。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链^[1]；Visual Studio Code。
- 硬件要求：笔记本/台式电脑1台^[2]。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4_fmu-v6x_default，推荐PX4固件版本为：1.12.3。其他配套飞控及编译命令请见：<https://rflsim.com/doc/zh/1/Hardware.html>

3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\7.RflySimPHM\0.ApiExps\e12_health_basic](#)

- [health_basic.py](#)：故障判断文件，包含健康评估算法和故障注入功能
- [Off_ctrl_connect_copter.py](#)：软硬件在环切换文件，用于选择仿真模式
- [Python38Run.bat](#)：Python运行环境启动脚本
- [bat\FaultModelV5HITL.bat](#)：硬件在环仿真环境启动脚本
- [bat\FaultModelV5SITL.bat](#)：软件在环仿真环境启动脚本

4. 实验内容或步骤

4.1 步骤1：修改程序路径

bat文件夹下的bat文件。（PX4PSP默认在c盘，如果装在别的盘，需要改此文件：自动测试平台V5\bat\FaultModelV5SITL.bat（用vscode或记事本打开），将C（见下图）改为自己的盘）

```
6 SET PSP_PATH=C:\PX4PSP
7 SET PSP_PATH_LINUX=/mnt/C/PX4PSP
8 C:
-
```

自动测试平台V5\bat\FaultModelV5HITL.bat将C（见下图）改为自己的盘。

```
6 SET PSP_PATH=C:\PX4PSP
7 C:
```

4.2 步骤2：选择软\硬件仿真

修改Off_ctrl_connect_copter.py。

如果要插入真实飞控进行仿真测试，将Off_ctrl_connect_copter.py的11行改为

```
11 cmdStr = '{}/bat//FaultModelV5HITL.bat'.format(path)
```

如果不插入飞控，进入软件在环，将Off_ctrl_connect_copter.py的11行改为

```
11 cmdStr = '{}/bat//FaultModelV5SITL.bat'.format(path)
```

4.3 步骤3：运行python程序

在文件夹下，双击Python38Run.bat，打开集成好的python环境，在该环境下运行health_basic.py文件，输入python health_basic.py，运行health_basic.py。

此为基础版判断故障，即简单通过特定时间内位移的变化，有一定的误诊断。

直接运行即可。

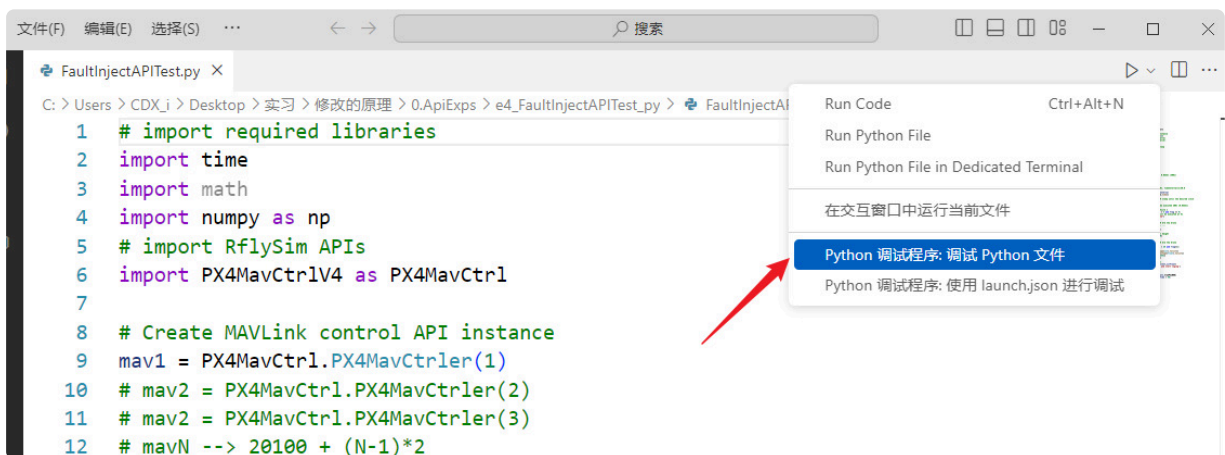
4.4 步骤4：VS Code调试运行（选作实验）

准备工作：

- 先确保已经按 [RflySimAPIs\1.RflySimIntro\2.AdvExps\e3.PythonConfig\Readme.pdf](#) 步骤，正确配置VS Code环境。或者配置了自己的Pycharm等自定义Python环境。
- 其他步骤与上文相同，在运行 [health_basic.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [health_basic.py](#) 文件，并阅读代码，修改代码，调试执行等。

扩展实验：

- 请自行使用VS Code阅读 [health_basic.py](#) 源码，通过程序跳转，了解每条代码的执行原理；再通过调试工具，验证每条指令的执行效果。



5. 关键知识点

5.1 关键知识点1：导入模块

PX4MavCtrlV4：用于与PX4飞行控制器进行通信。

MavDataRec：用于记录无人机的飞行数据。

time, math, os, random, numpy：Python标准库，用于各种辅助功能。

5.2 关键知识点2：初始化无人机和数据记录

创建PX4MavCtrlr实例mav1与无人机建立通信。

使用Off_ctrl_connect_copter.HIL_connect(mav1)将无人机连接到硬件在环仿真环境。

启动数据记录器data1，记录LOCAL_POSITION_NED消息。

5.3 关键知识点3：飞行模式控制

脚本设定了不同的飞行模式和对应的时间条件。

ModeFlag标志位用于控制飞行模式的状态。

在特定时间后，通过发送命令改变无人机的飞行状态（例如起飞、悬停、移动）。

5.4 关键知识点4：健康评估算法

使用PHMAlgFlag控制健康评估算法的执行。

算法目的是监测无人机的飞行数据并进行故障诊断。

5.5 关键知识点5：故障注入

使用InJectFlag标志位控制故障注入的执行。

在满足一定条件后，注入故障以测试无人机的容错能力。

故障注入通过发送特定消息到无人机的仿真环境来实现。

5.6 关键知识点6：主循环

主循环使用while True无限循环，通过计算睡眠时间来控制执行频率。

根据预设的时间条件和飞行模式，执行相应的控制命令和数据记录。

5.7 关键知识点7：数据处理和故障检测

脚本中预留了数据处理和故障检测的代码位置，但实际的健康评估算法尚未实现。

使用LOCAL_POSITION_NED消息中的数据进行简单故障检测示例，当检测到异常高度变化时，打印"故障"。

6. 参考资料

1. [RflySim官方文档](#)
2. [PX4开发者指南](#)
3. [MAVLink协议规范](#)

7. 常见问题

Q1: 运行程序时出现连接错误怎么办?

A1: 检查CopterSim是否已启动且已完成初始化（确认已显示"GPS 3D fixed & EKF initialization finished"）。同时检查 [Off_ctrl_connect_copter.py](#) 中的仿真模式设置是否正确。

Q2: 如何区分软件在环(SITL)和硬件在环(HITL)仿真?

A2: SITL是纯软件仿真，不依赖实际飞控硬件；HITL需要连接真实的飞控硬件。可以通过修改 [Off_ctrl_connect_copter.py](#) 中的IS_HIL参数来切换两种模式。

Q3: 健康评估算法的判断依据是什么?

A3: 目前的基础版本通过分析LOCAL_POSITION_NED消息中的位置和速度数据，检测是否存在异常变化来判断可能的故障情况。更复杂的算法可以根据传感器数据设计专门的故障检测算法。

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