

1. 实验名称及目的

1.1 实验名称

基于最大模板的传感器模块故障注入的原理。

1.2 实验目的

了解基于最大模板的传感器模块故障注入的原理。

1.3 关键知识点

原理：FaultInjectAPITest.py通过

PX4MavCtrlV4库的sendInDoubCtrls/sendSILIntDouble/sendSILIntFloat 系列接口，将故障信号inSIL28d（或inSILInts/inSILFloats）在UDP端口中传输到simulink最大模块中的传感器故障注入模块，以此实现传感器故障注入功能。编译simulink最大模块生成DLL文件，将DLL文件导入CopterSim后，成功注入传感器故障并完成仿真。

传感器故障在故障信号inSIL28d的识别ID为：123542，123543，123544，123545；分别对应下方表格。其故障输入参数为：噪声增益。若未注入故障，传感器的噪声=默认噪声*0.2，若注入故障，传感器的噪声=默认噪声*（0.2+0.8*噪声增益）。噪声注入方式为原信号与噪声信号叠加。

注释：噪声增益 (Noise Gain)：噪声增益是指在传感器信号上人为添加的噪声强度，用来模拟传感器信号受到干扰或噪声影响的情况。增益的值越大，模拟的噪声强度越高，对传感器信号的准确性影响也越大。

故障ID及对应参数见<../..../RflySimSDK/html/md_phm_2md_2Faultinject.html>

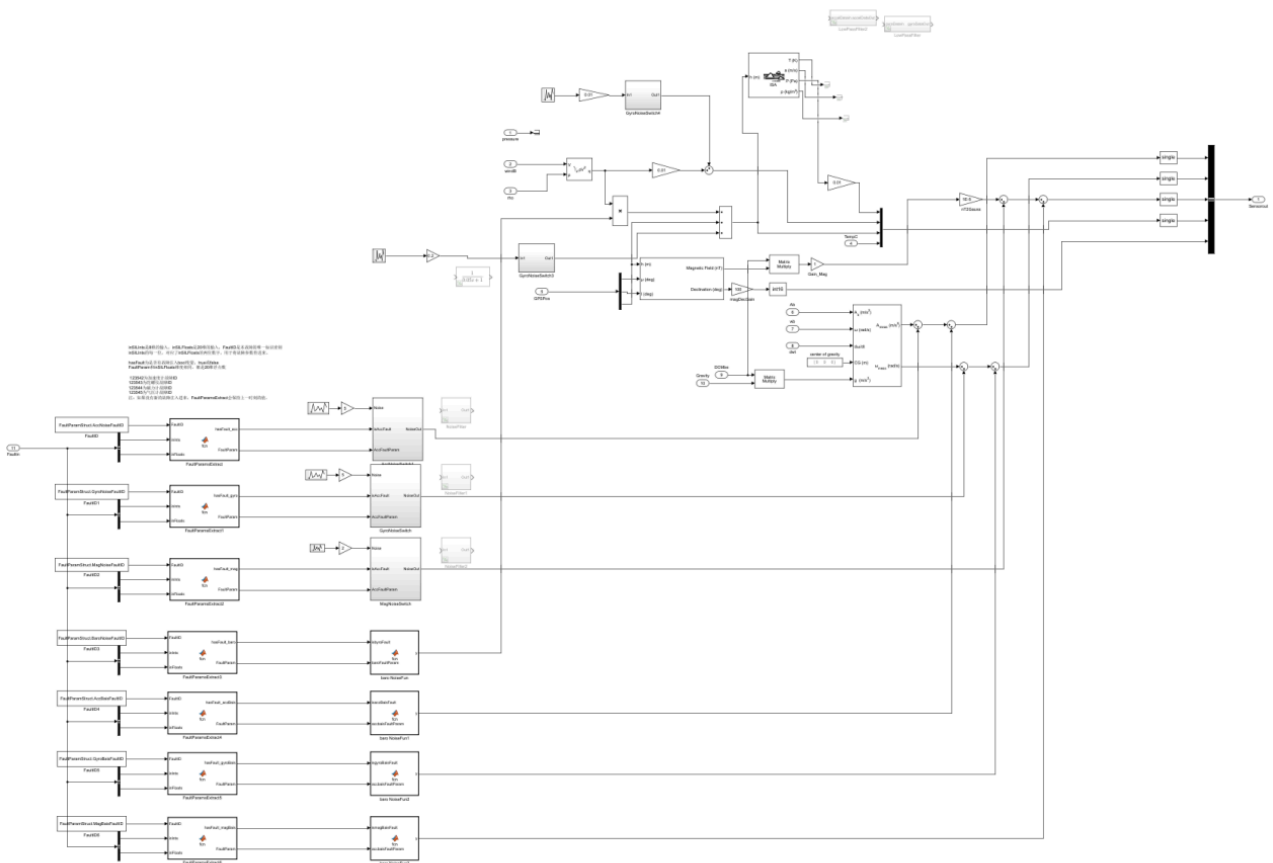
故障ID	故障类型	输入参数
123542	加速度计噪声干扰	噪声增益
123543	陀螺仪噪声干扰	噪声增益
123544	磁力计噪声干扰	噪声增益

故障ID	故障类型	输入参数
123545	气压计噪声干扰	噪声增益

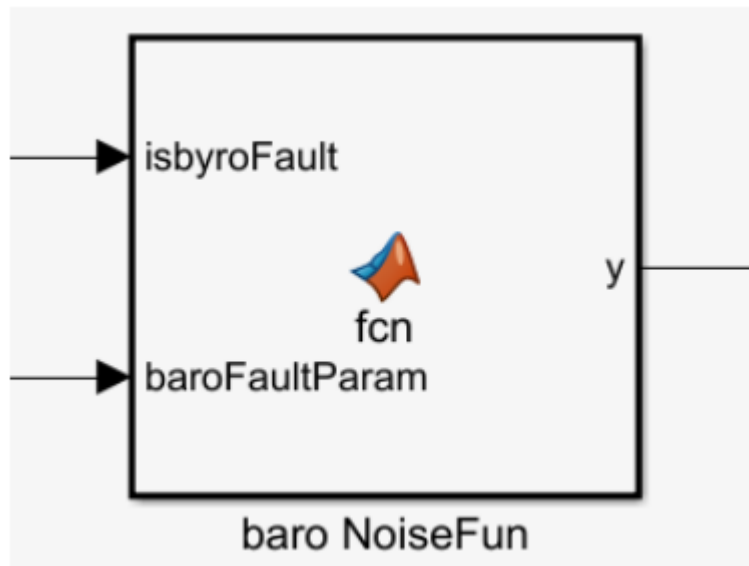
注入故障所用到的simulink模块SensorFault结构图 及其 输入输出参数解释：

该模块用于提取故障参数和输出注入故障后的传感器数据。模块输入FaultIn，即inSIL28d，为包含8维的inSILInts，20维的inSILFloats的故障参数。输出为注入噪声后传感器采集的无人机加速度、角速度、磁场与维度乘积。其中带噪声的坐标是指无人机仿真准确数据叠加随机噪声，目的是更贴合真实情况。

功能	
输入	
pressure	大气压强
windB	速度，用三元素向量表示。
rho	空气密度，用标量显示
TempC	大气
GPSPos	定位
Ab	在固定物体轴上的实际加速度，用三元素矢量指定，以选定的单位表示。
wb	以物体固定轴表示的角速率，用三元素矢量表示，单位为弧度/秒。
dwt	固定物体轴的角加速度，用三元素矢量表示，单位为弧度/秒的平方。
DCMbe	方向余弦矩阵，用于将载荷移动引起的力转换到无人机坐标系中。
Gravity	重力
FaultIn	故障输入，包括故障类型和故障参数。
输出	
Sensorout	将无人机加速度、角速度、磁场与维度乘积、维度封装在bus总线中



注入故障所用到的simulink模块baro NoiseFun内部结构：



```

1  function y = fcn(isbyroFault, baroFaultParam)
2
3  y = 0;
4  if isbyroFault
5      y = baroFaultParam(1);
6  end

```

功能	根据是否存在气压计故障的标志来确定输出的值。
输入	
isbyroFault	指示是否存在气压计故障。如果为 true，表示存在气压计故障。
baroFaultParam	包含气压计故障参数的向量。在这个函数中，baroFaultParam 的第一个元素用于设置故障值。
输出	
y	函数的输出结果。默认情况下，y 被初始化为 0。如果 isbyroFault 为 true，则 y 会被设置为 baroFaultParam 的第一个元素。

主要要求掌握：

传感器故障注入模块的功能及其实现。

说明文档	说明文档链接以及地址
simulink中SensorFault模块	PX4PSP/RflySimSDK/html/md_phm_2md_2SensorFault.html
FaultInjectAPITest.py	PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py\readme.pdf
PX4MavCtrlV4库	PX4PSP/RflySimSDK/html/PX4MavCtrlV4_8py.html

2. 实验效果

3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\7.RflySimPHM\1.BasicExps\e4_SensorFault](#)

文件夹/文件名称	说明
MulticopterModel.slx	故障注入模块的最大模板模型文件。
MulticopterModelHITL.bat	硬件在环仿真批处理文件。
MulticopterModelSITL.bat	软件在环仿真批处理文件。
GenerateModelDLLFile.p	DLL格式转化文件。
Init.m	动力学模型相关参数。
Python38Run.bat	Python程序执行脚本。
FaultInjectAPITest.py	故障注入程序。

4. 运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；MATLAB 2022B及以上版本。

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://rflysim.com/doc/zh/1/Hardware.html>

4.2 硬件要求

笔记本/台式电脑① 1台。

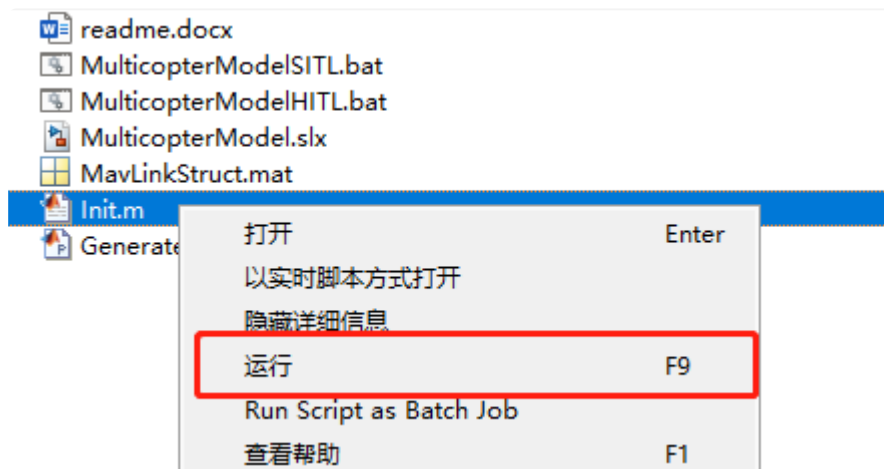
①：推荐配置请见：<https://rflysim.com/>

5. 实验步骤

5.1. 必做实验

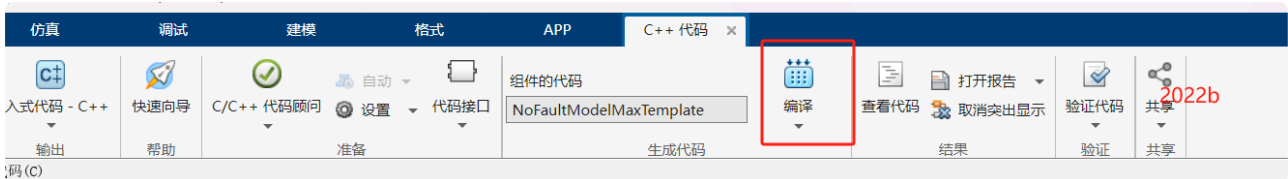
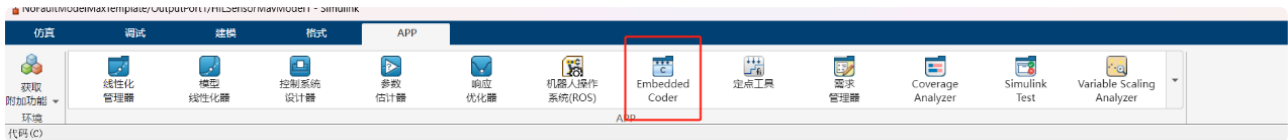
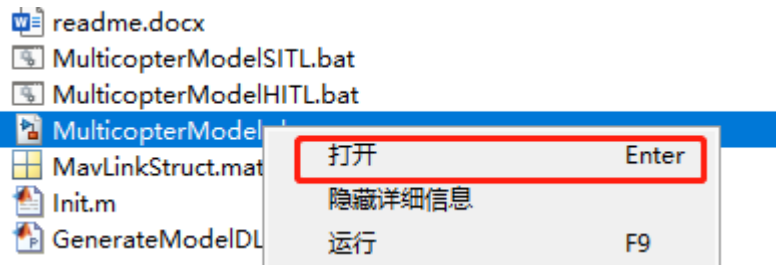
Step 1: 初始化数据

打开“Init.m”文件并运行。



Step 2: 编译文件

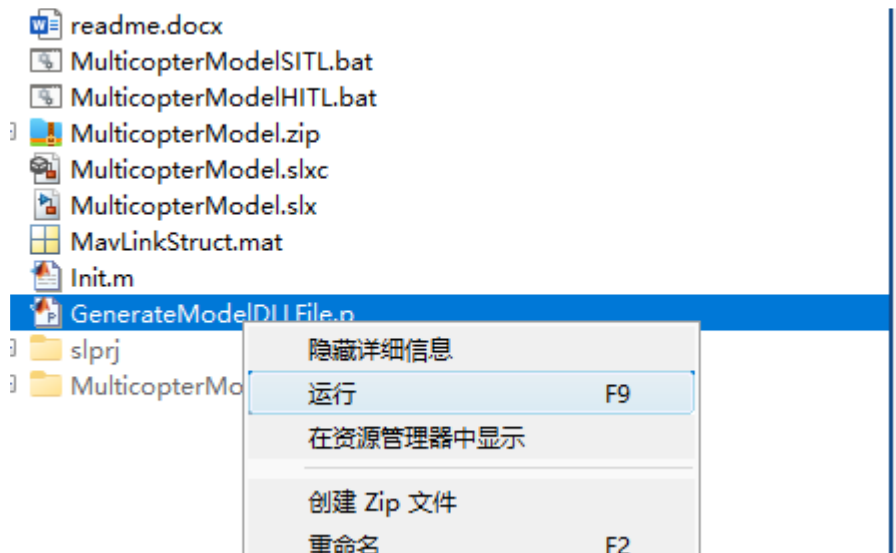
打开“MulticopterModel.slx” Simulink 文件，点击Build Model 按钮生成代码。



注：如果故障模块版本错误，无法编译，需要从故障模块库中选择对应的模块进行替换。

Step 3: 生成dll文件

代码生成完毕后，在 matlab 中右键“GenerateModelDLLFile.p”文件，点击运行，生成 DLL 文件。



Step 4: 运行软件在环仿真

以管理员身份运行软件在环脚本。



Step 5: 打开python文件，修改故障参数

打开Visual Studio

Code，选择打开文件夹，打开文件夹

RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py。



对FaultInjectAPITest.py其中的故障注入代码按照

RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py中的

FaultInjectAPITest_py中的故障注入代码更改为传感器模块故障（传感器模块故障注入代码可以查看参考文献），并对故障参数进行修改。如下加速度计噪声故障参数为30（具体修改方法可以参考e4_FaultInjectAPITest_py文件夹中的readme）

```
if time.time() - start_time > 20 and flag == 1:
    #np.zeros()
    silInt=np.zeros(8).astype(int).tolist()
    silFloat=np.zeros(20).astype(float).tolist()
    # silInt[0:2]=[123546,123546]
    # silFloat[0:4]=[0,0,0,0]
    silInt[0:1]=[123542]
    silFloat[0:2]=[30,0]
```

注：传感器故障包含加速度计、陀螺仪、磁力计、气压计四种故障

注：文件中的silFloat已经修改好，可以直接运行注入例程对应故障，如果想要同时注入多种故障，可以参考e4_FaultInjectAPITest_py文件夹中的readme。

Step 6: 运行python文件实现故障注入

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FaultInjectAPITest.py文件，输入python FaultInjectAPITest.py，运行FaultInjectAPITest.py。

```
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
D:\PX4PSP\RfIySimAPIs\7.RfIySimPHM\0.ApiExps\e4_FaultInjectAPITest.py>python FaultInjectAPITest.py
```

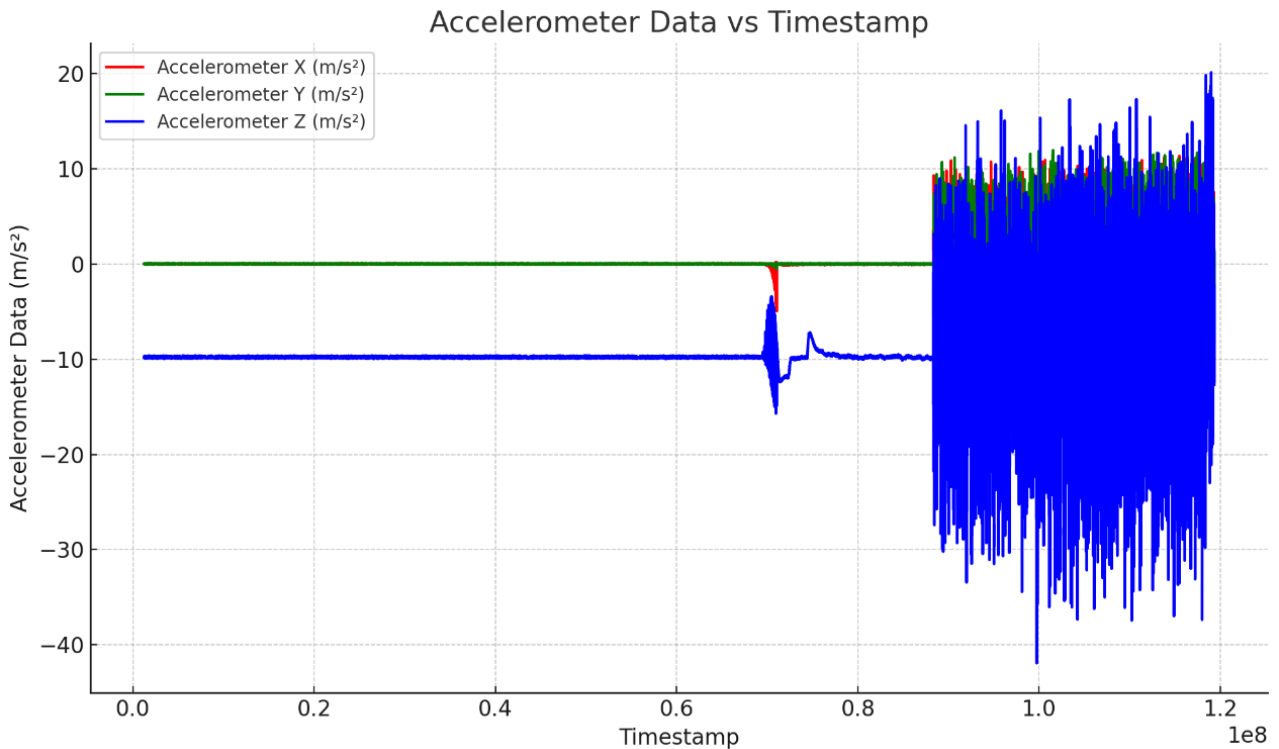
```
C:\windows\system32\cmd.exe - python FaultInjectAPITest.py
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put your python scripts 'XXX.py' into the folder 'D:\PX4PSP\RfIySimAPIs\Python38Scripts'
Use the command: 'python XXX.py' to run the script with Python
For example, try entering 'python ImgCVShow.py' below to use OpenCV to read and show a image
You can also use pyulog (see https://github.com/PX4/pyulog) to convert PX4 log file
For example, try entering 'ulog2csv log.ulg' to convert ulg file to excel files for MATLAB
D:\PX4PSP\RfIySimAPIs\Python38Scripts>D:\PX4PSP\RfIySimAPIs\7.RfIySimPHM\0.ApiExps\e4_FaultInjectAPITest_py`
D:\PX4PSP\RfIySimAPIs\Python38Scripts>`cd D:\PX4PSP\RfIySimAPIs\7.RfIySimPHM\0.ApiExps\e4_FaultInjectAPITest_py
'cd' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
D:\PX4PSP\RfIySimAPIs\Python38Scripts>cd D:\PX4PSP\RfIySimAPIs\7.RfIySimPHM\0.ApiExps\e4_FaultInjectAPITest_py
D:\PX4PSP\RfIySimAPIs\7.RfIySimPHM\0.ApiExps\e4_FaultInjectAPITest_py>python FaultInjectAPITest.py
5s, Arm the drone
Arm the drone!
开始起飞
```

然后可以看到无人机已经注入故障，可以看到飞机的姿态控制发生紊乱，无法定点悬停



Step7读取日志查看故障注入效果

以加速度计噪声干扰为例，首先解析对应的飞控日志，（软件在环日志位于 `C:\PX4PSP\Firmware\build\px4_sitl_default\instance_1\log`，硬件在环日志位于飞控SD卡中），将ulg文件转换为csv后，读取 `_sensor_combined_0.csv` 中的 `accelerometer_m_s2[0]~accelerometer_m_s2[2]`，可以看到可以看到注入故障后加速度计噪声增大



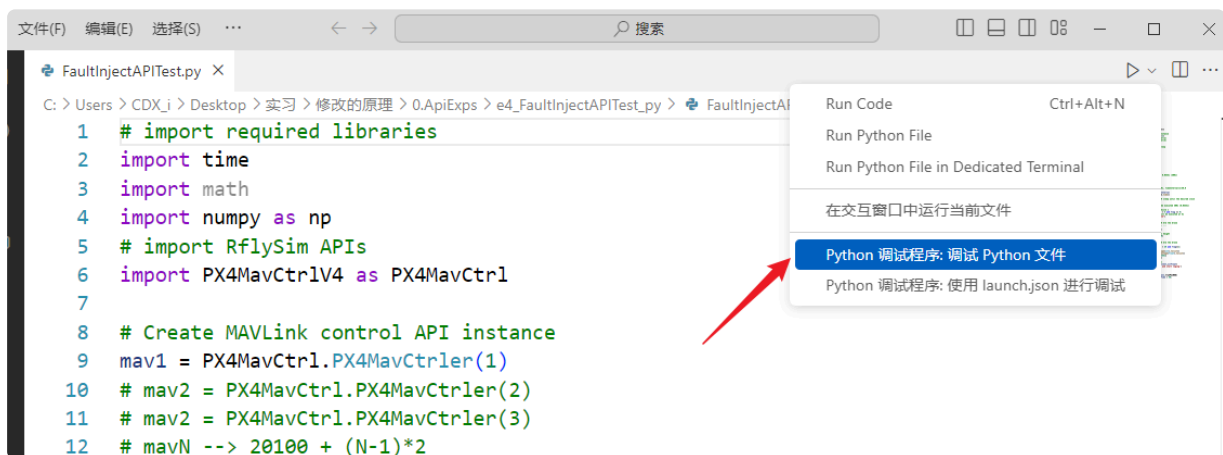
5.2. 选作实验（VS Code调试运行）

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FaultInjectAPITest.py时，可使用VS Code（或Pycharm等工具）来打开FaultInjectAPITest.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FaultInjectAPITest.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。



```
文件(F) 编辑(E) 选择(S) ... 搜索
FaultInjectAPITest.py X
C:\Users\CDX_i\Desktop\实习>修改的原理>0.ApiExps>e4_FaultInjectAPITest.py > FaultInjectA
1 # import required libraries
2 import time
3 import math
4 import numpy as np
5 # import RflySim APIs
6 import PX4MavCtrlV4 as PX4MavCtrl
7
8 # Create MAVLink control API instance
9 mav1 = PX4MavCtrl.PX4MavCtrler(1)
10 # mav2 = PX4MavCtrl.PX4MavCtrler(2)
11 # mav2 = PX4MavCtrl.PX4MavCtrler(3)
12 # mavN --> 20100 + (N-1)*2
```

Run Code Ctrl+Alt+N
Run Python File
Run Python File in Dedicated Terminal
在交互窗口中运行当前文件
Python 调试程序: 调试 Python 文件
Python 调试程序: 使用 launchjson 进行调试

6. 参考资料

故障ID及对应参数见<../..../RflySimSDK/html/md_phm_2md_2Faultinject.html>

7. 常见问题

Q1: ***

A1: ***