

# 1. 实验名称及目的

## 1.1 实验名称

基于最大模板的电机模块故障注入的原理。

## 1.2 实验目的

了解最大模板的电机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最大模块中的电机故障注入模块，以此实现电机故障注入功能。编译simulink最大模块生成DLL文件，将DLL文件导入CopterSim后，成功注入电机故障并完成仿真。

电机故障在故障信号inSIL28d的识别ID为：123450；其故障输入参数按照排列顺序依次为：#1~#x号电机执行效率比（0~1）。若未注入故障，电机执行效率比默认为100%。

**注释：电机执行效率比：电机实际输出的机械功率与输入的电功率之间的比率。**

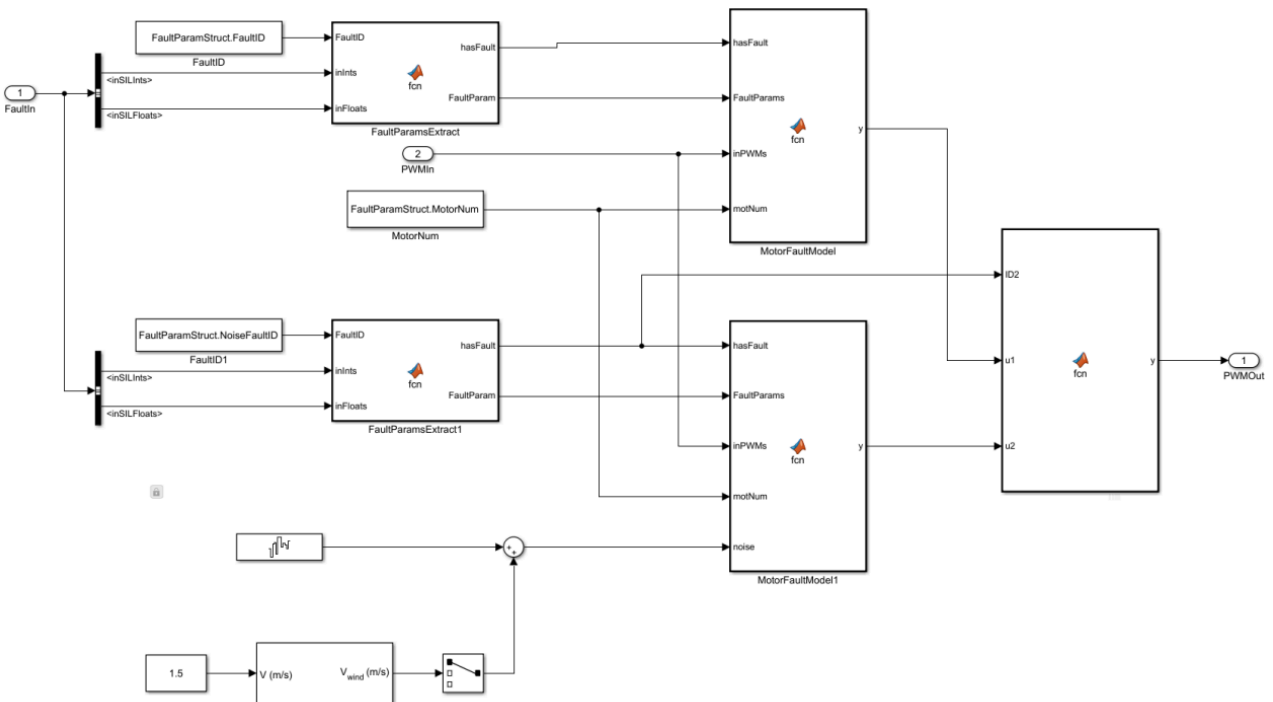
$$\text{电机执行效率}(\eta) = \frac{\text{输出的机械功率}(P_{\text{out}})}{\text{输入的电功率}(P_{\text{in}})} \times 100\%$$

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

<b>功能</b>	<b>包含FaultParamsExtract噪声参数提取模块， MotorFaultModel电机模型模块。本模块用于从 无噪声电机执行效率故障 和 有噪声电机执行效率故障 选择电机故障类型，并输出注入故障后电机转速。</b>
	无噪声电机执行效率故障 和 有噪声电机执行效率故障 这两个故障模块的实现机制不同，但都同属于电机故障。通过用对应的故障ID可以触发对应的故障。123450为电机故障ID
<b>输入</b>	
FaultIn	故障输入，包括故障类型和故障参数。
FaultID	故障的唯一标识密钥
NoiseFaultID	噪声故障的唯一标识密钥
MotorNum	无人机的故障电机数量
<b>输出</b>	
PWMOut	注入故障后电机输出的转速

该模块用于提取故障参数并输出注入故障后的电机转速。模块输入FaultIn，即inSIL28d，为包含8维的inSILInts，20维的inSILFloats的故障参数。其中有两个MotorFaultModel（图中上方的未叠加噪声因素，下方的叠加了噪声因素），这两个故障模块的实现机制不同，但都同属于电机故障。通过用对应的故障ID可以触发对应的故障。后续接入的MATLAB Function用于对两个MotorFaultModel二选一并输出。

inSILInts是8维的输入，inSILFloats是20维的输入，FaultID是本故障的唯一标识密钥  
inSILInts的每一位，对应了inSILFloats的两位数字，用于将故障参数传进来。  
hasFault为是否有故障注入bool变量，true或false  
FaultParam和inSILFloats维度相同，都是20维浮点数  
123450为电机故障ID  
注：如果没有新的故障注入进来，FaultParamsExtract会保持上一时刻的值。



主要要求掌握：

电机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MotorFualt模块	<a href="PX4PSP/RflySimSDK/html/md_phm_2md_2Motor_01FaultModel.html">PX4PSP/RflySimSDK/html/md_phm_2md_2Motor_01FaultModel.html</a>
<a href="#">FaultInjectAPITest.py</a>	<a href="PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py\readme.pdf">PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py\readme.pdf</a>
PX4MavCtrlV4库	<a href="PX4PSP/RflySimSDK/html/PX4MavCtrlV4_8py.html">PX4PSP/RflySimSDK/html/PX4MavCtrlV4_8py.html</a>

## 2. 实验效果

## 3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\7.RflySimPHM\1.BasicExps\e3\\_MotorFault](#)

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。
<a href="#">FaultInjectAPITest.py</a>	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。

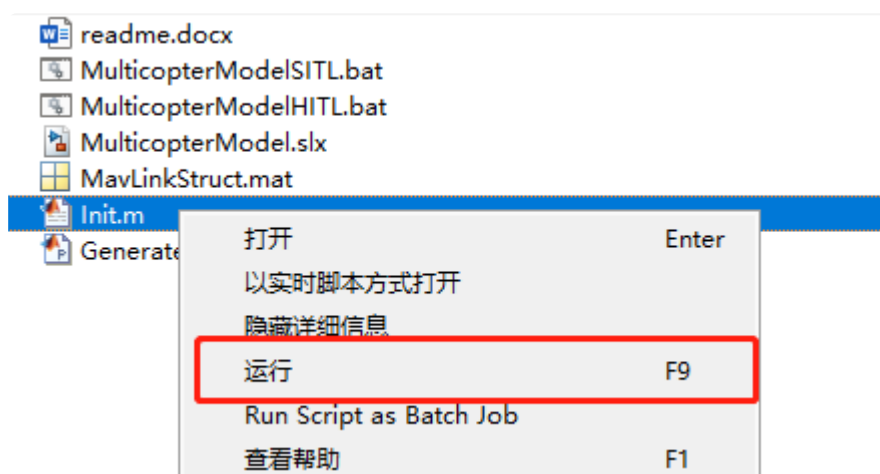
①：推荐配置请见：<https://rflysim.com/>

## 5. 实验步骤

### 5.1. 必做实验

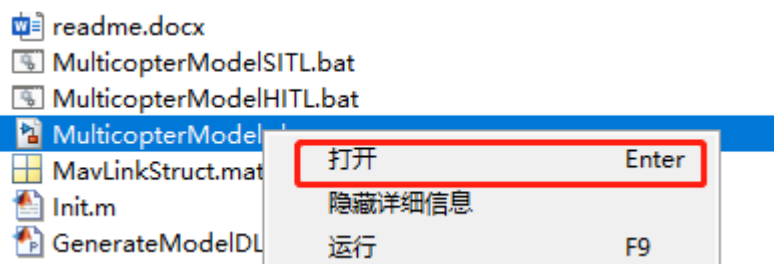
#### Step 1: 初始化数据

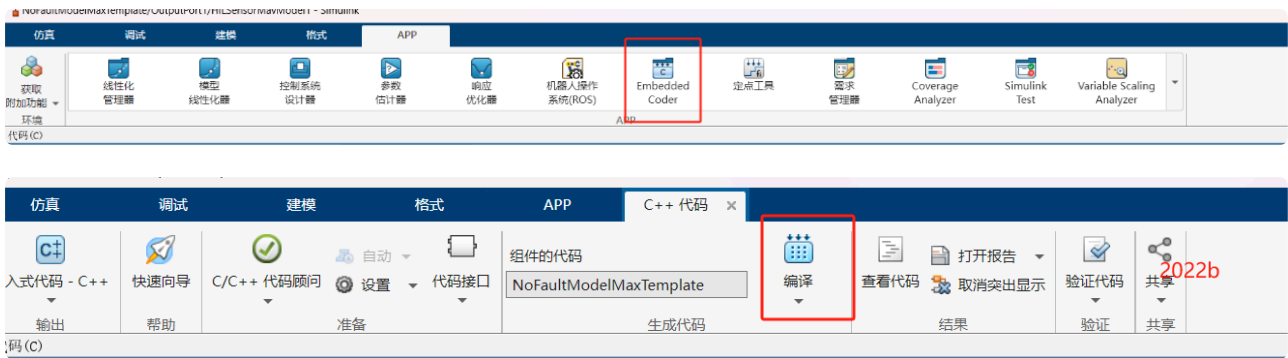
打开“Init.m”文件并运行。



#### Step 2: 编译文件

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

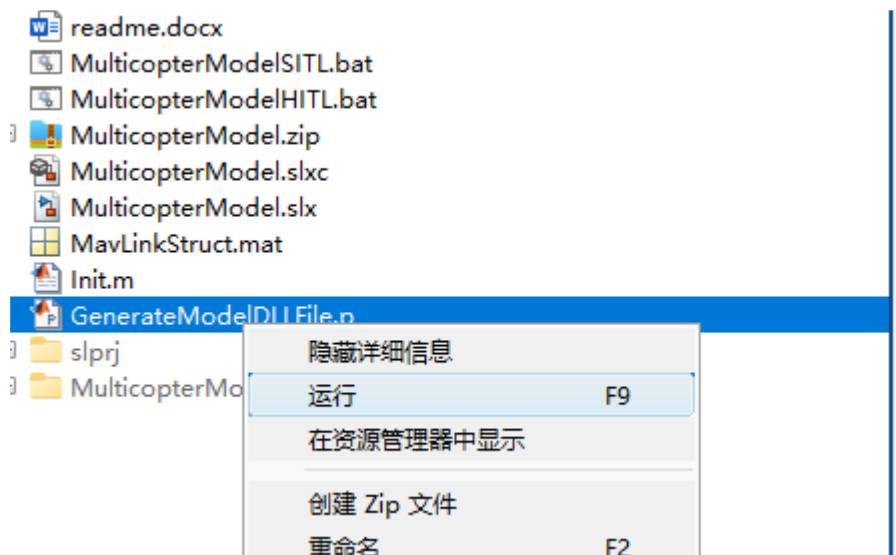




注：如果故障模块版本错误，无法编译，需要从故障模块库中选择对应的模块进行替换。

### 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电机故障（电机故障注入代码可以查看参考文献），并对故障参数进行修改。（具体修改方法可以参考e4\_FaultInjectAPITest\_py文件夹中的readme）

```
60     if time.time() - starttime > 20 and flag==1:
61         #np.zeros()
62         silInt=np.zeros(8).astype(int).tolist()
63         silFloat=np.zeros(20).astype(float).tolist()
64         silInt[0:2]=[123450,123450]
65         silFloat[0:4]=[0.6,0,0,0]
66         # silInt[0:1]=[123542]
67         # silFloat[0:2]=[15,0]
68         mav1.sendSILIntFloat(silInt,silFloat)
```

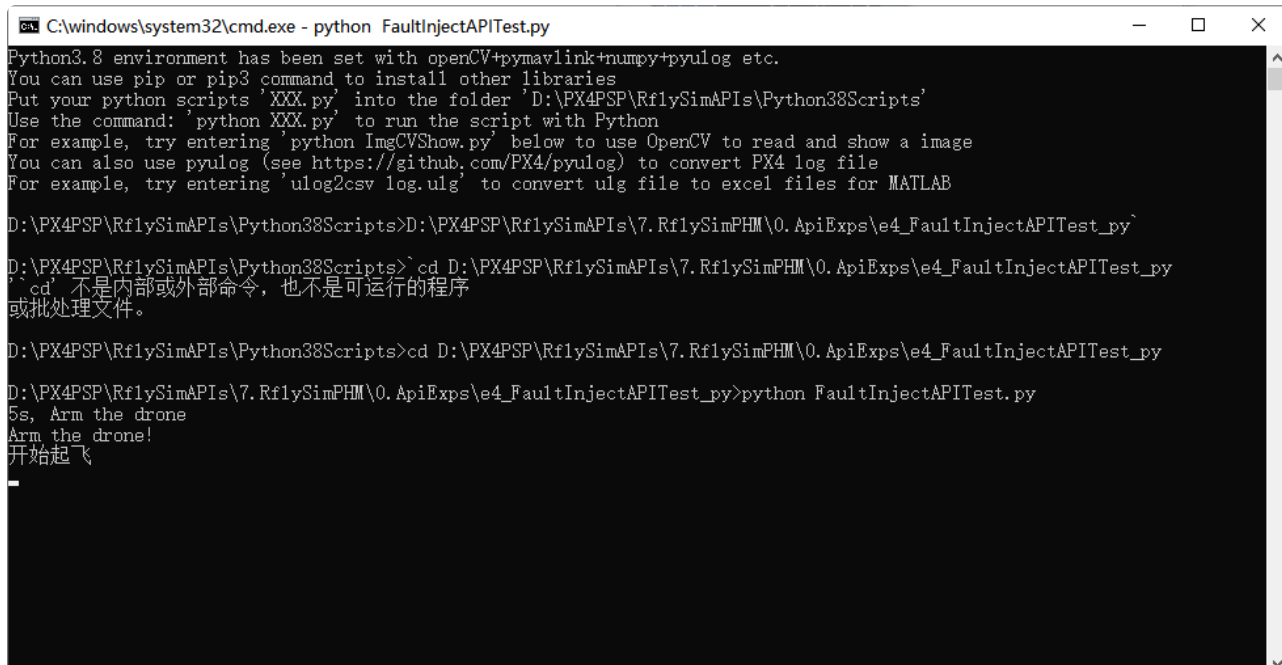
注：文件中的silFloat已经修改好，可以直接运行注入例程对应故障。

## 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\RflySimAPIs\7.RflySimPHM\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\RflySimAPIs\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\RflySimAPIs\Python38Scripts>D:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py
D:\PX4PSP\RflySimAPIs\Python38Scripts>`cd D:\PX4PSP\RflySimAPIs\7.RflySimPHM\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\RflySimAPIs\Python38Scripts>cd D:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py
D:\PX4PSP\RflySimAPIs\7.RflySimPHM\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，可以看到飞机直接坠毁。



再次发送起飞指令，可以看到只有一个电机旋转



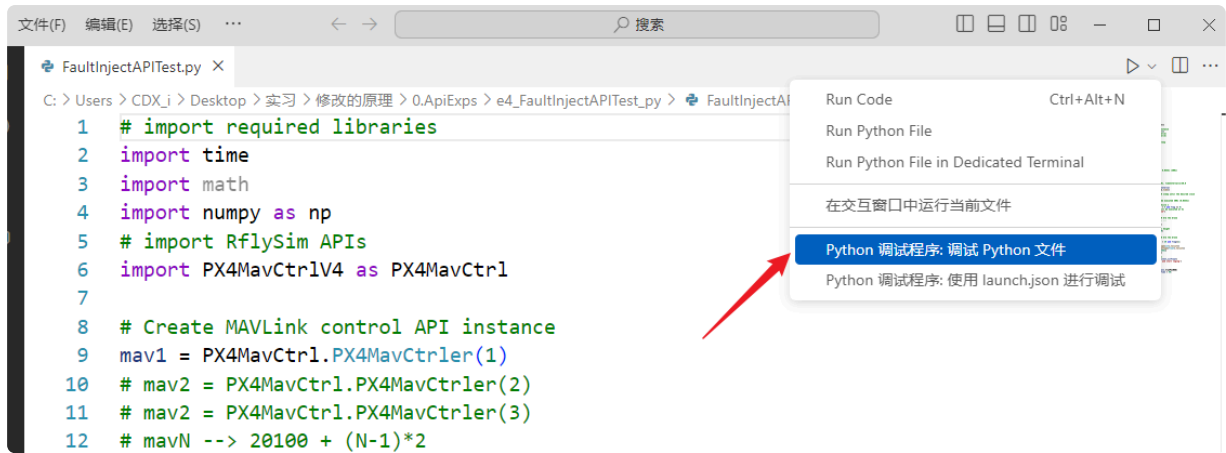
## 5.2. 选作实验（VS Code调试运行）

### 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。



```
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.PX4MavCtrl(1)
10 # mav2 = PX4MavCtrl.PX4MavCtrl(2)
11 # mav2 = PX4MavCtrl.PX4MavCtrl(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: \*\*\*