

# 1. 实验名称及目的

## 1.1 实验名称

基于最大模板的负载模块故障注入。

## 1.2 实验目的

了解基于最大模板的负载模块故障注入。

## 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最大模块中的负载故障注入模块，以此实现负载故障注入功能。编译simulink最大模块生成DLL文件，将DLL文件导入CopterSim后，成功注入负载故障并完成仿真。

负载故障在故障信号inSIL28d的识别ID为：123455，123456，123457，分别对应下方表格。其故障输入参数为：重量泄露比（0~1），x,y,z的泄露因子（0~1），泄露因子(0~1)。若未注入故障，默认无人机重量保持不变。

**注释：重量泄露比（Weight Leak Ratio）和泄露因子（Leakage Factor）：**  
是描述系统中质量或重量随时间减少的比例的术语，  
通常用于表示由于某种形式的泄露或损失导致的重量减少。重量泄露比：  
这是一个介于0到1之间的数值，表示由于泄露导致的重量损失占初始重量的比例。  
例如，如果一个容器的重量泄露比是0.2，  
那么意味着容器的重量由于泄露减少了20%。泄露因子：  
泄露因子通常用于更具体的上下文中，比如x、y、  
z轴的泄露因子可能表示在三个空间维度上的质量损失比例。  
每个轴的泄露因子都是一个独立的值，用于描述沿该轴向的泄露对总质量的影响。  
泄露因子的应用：在航空航天领域，泄露因子可能用于描述由于环境因素  
（如大气阻力）或结构损伤（如舱室破裂）导致的飞行器质量损失。

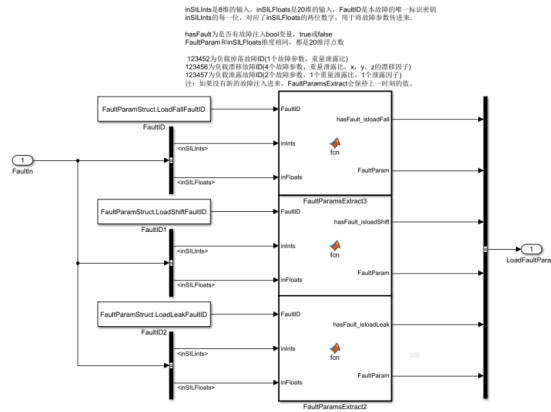
故障ID及对应参数见<../..../RflySimSDK/html/md\_phm\_2md\_2Faultinject.html>

故障ID	故障类型	输入参数
123455	负载故障	重量泄露比 (0~1)
123456	负载漂移故障	重量泄露比+x,y,z的泄露因子 (0~1)
123457	负载泄露故障	重量泄露比+泄露因子(0~1)

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

该模块用于提取故障参数。模块输入FaultIn，即inSIL28d，为包含8维的inSILInts，20维的inSILFloats的故障参数。

功能	负载故障模块 用于表示负载故障类型以及对应的故障参数。(如：LoadFall负载掉落故障、LoadShift负载漂移故障、LoadLeak负载泄露故障) inSILInts的每一位，对应了inSILFloats的两位数字，用于将故障参数传进来。
	123452为负载掉落故障ID(1个故障参数，重量泄露比)；123456为负载漂移故障ID(4个故障参数，重量泄露比，x，y，z的漂移因子)；123457为负载泄露故障ID(2个故障参数，1个重量泄露比，1个泄露因子)；注:如果没有新的故障注入进来，FaultParamsExtract会保持上一时刻的值。
输入	
FaultIn	故障输入，包括故障类型和故障参数。
LoadFallFaultID	负载掉落故障的唯一标识密钥
LoadShiftFaultID	负载漂移故障的唯一标识密钥
LoadLeakFaultID	负载泄露故障的唯一标识密钥
输出	
LoadFaultParam	将负载掉落，负载漂移，负载泄露故障参数 (hasFault, FaultParam ) 封装在总线中。



内部模块Model Failure结构：

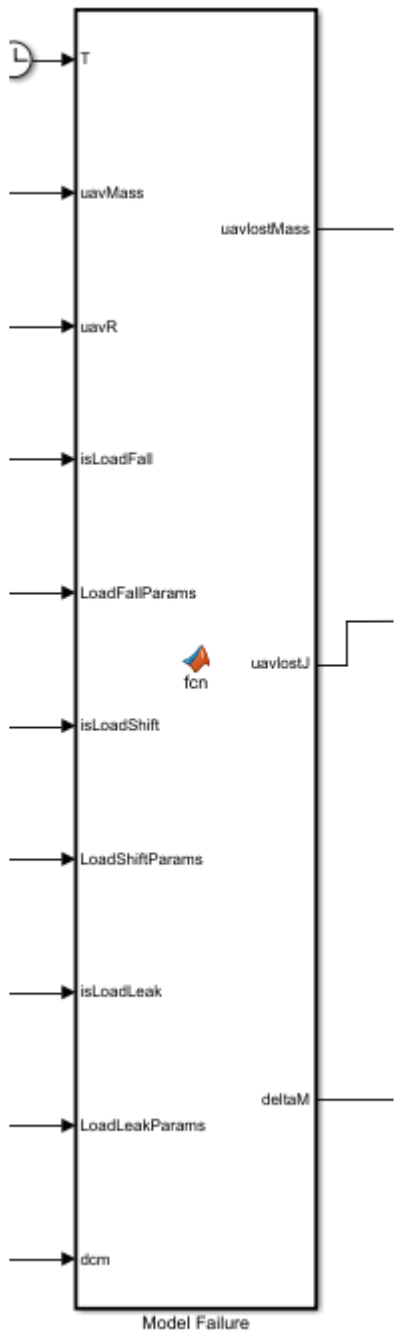
```
function [uavlostMass,uavlostJ,deltaM] = fcn(T,uavMass,uavR,isLoadFall,LoadFallParams,isLoadShift,LoadShiftParams,isLoadLeak,LoadLeakPara
% This function aims to calculate the variation of mass and moment of
% inertia caused by model failure including LoadFall, LoadShift, and
% LoadLeak.

persistent tOld;
if isempty(tOld)
    tOld=0;
end

deltaM=[0;0;0];

if isLoadFall
    if LoadFallParams(1)>0.95
        LoadFallParams(1)=0.95;
    elseif LoadFallParams(1)<0
        LoadFallParams(1)=0;
    end
    uavlostMass=LoadFallParams(1)*uavMass;
    uavlostJ=zeros(3,3);
elseif isLoadShift
    if LoadShiftParams(1)>0.95
        LoadShiftParams(1)=0.95;
    elseif LoadShiftParams(1)<0
        LoadShiftParams(1)=0;
    end
end

uavlostMass= 0;
```



功能	该模块包含了负载故障模块以及故障模型模块。负载故障模块 用于表示负载故障类型以及对应的故障参数（如：LoadFall 负载掉落故障、LoadShift 负载漂移故障、LoadLeak 负载泄露故障）；故障模型模块 用于计算由于模型故障(包括 LoadFall 负载掉落故障、LoadShift 负载漂移故障、LoadLeak 负载泄露故障)引起的质量和惯性矩的变化。
参数	
uavMass	无人机的总质量
uavR	无人机的某个特定参考半径或距离，用于计算移动质量的偏移。
isLoadFall	bool布尔值，指示是否发生负载掉落故障。
LoadFallParams	数组，包含负载掉落故障的参数。
isLoadShift	bool布尔值，指示是否发生负载漂移故障。
LoadShiftParams	数组，包含负载漂移故障的参数。
isLoadLeak	bool布尔值，指示是否发生负载泄露故障。
LoadLeakParams	数组，包含负载泄露故障的参数。
dcm	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。
uavlostMass	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。
uavlostJ	无人机由于载荷掉落、移动或泄漏损失的惯性矩矩阵（3x3矩阵）。
deltaM	由于载荷移动引起的力矩变化（3x1向量）。
输入	
6DOF Bus1	将简易环境模型和刚体六自由度模型的输出封装在总线中。
ModelParam.uavMass	无人机的总质量。
ModelParam.uavR	无人机的某个特定参考半径或距离，用于计算移动质量的偏移。
FaultIn	故障输入，包括故障类型和故障参数。
输出	
Fail Model Bus1	将uavMass, uavJ, deltaM, uavR封装在总线中。

主要要求掌握：

负载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LoadFault模块	<a href="PX4PSP/RflySimSDK/html/md_phm_2md_2FailModel.html">PX4PSP/RflySimSDK/html/md_phm_2md_2FailModel.html</a>
<a href="#">FaultInjectAPITest.py</a>	<a href="PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\xe4_FaultInjectAPITest_py\readme.pdf">PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\xe4_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\6\\_LoadFault](#)

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。

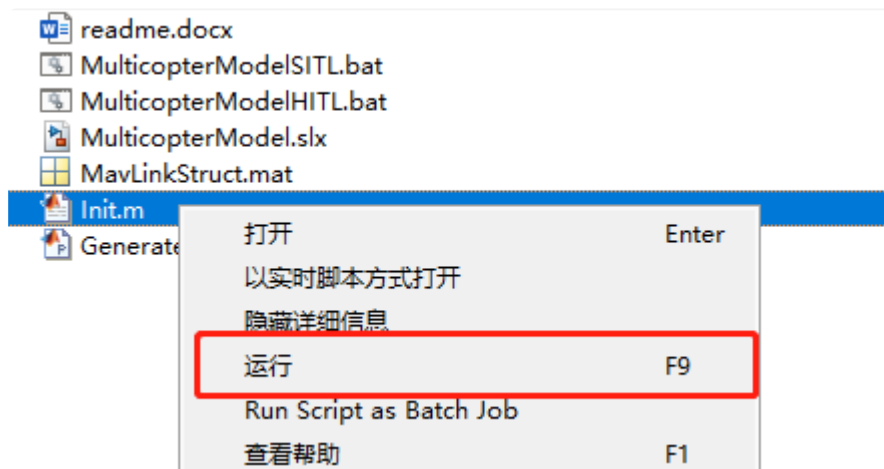
①：推荐配置请见：<https://rflysim.com/>

## 5. 实验步骤

### 5.1. 必做实验

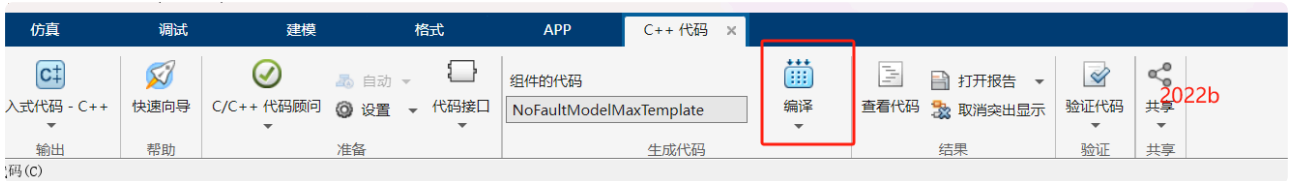
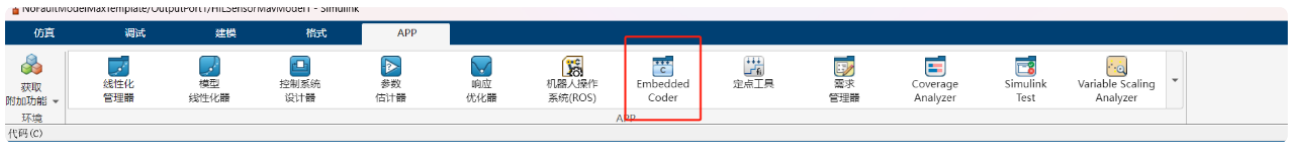
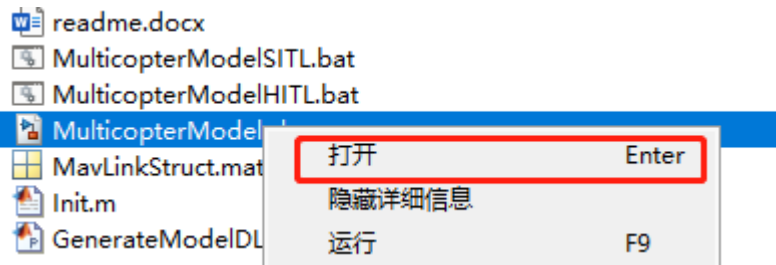
#### Step 1: 初始化数据

打开“Init.m”文件并运行。



#### Step 2: 编译文件

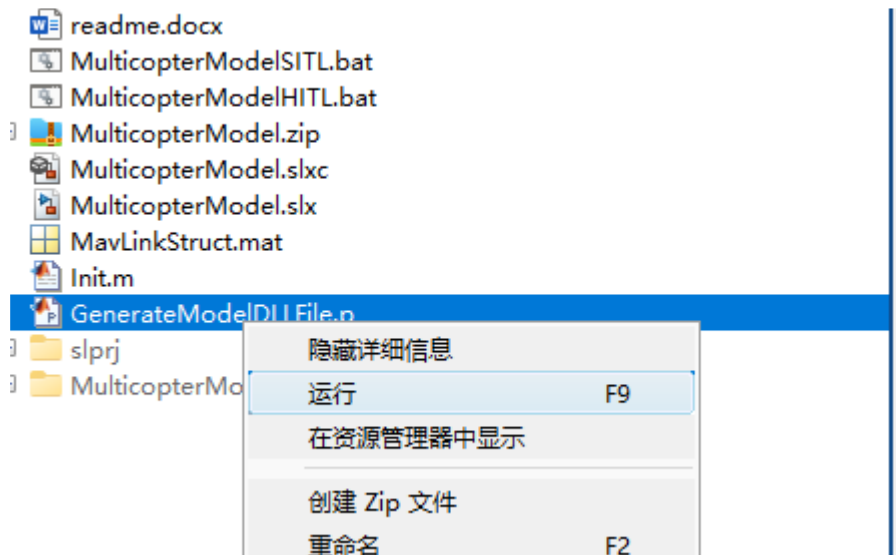
打开“MulticopterModel.slx” Simulink 文件，点击Build Model 按钮生成代码。



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。

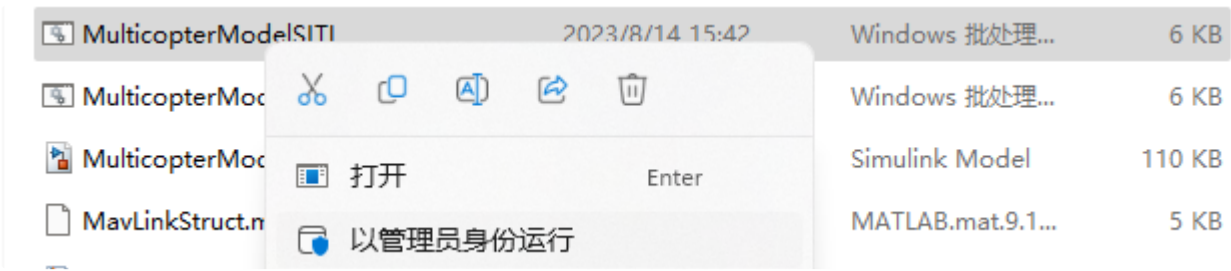
### Step 3: 生成dll文件

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负载模块  
故障（负载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。（具体修改方法  
可以参考e4\_FaultInjectAPITest\_py文件夹中的readme）

```
silInt[0:2]=[123456,123456]  
silFloat[0:4]=[1,2,3,4]
```

```
# silInt[0:1]=[123540]  
# silFloat[0:2]=[15,0]
```

注：负载模块故障包括负载故障、负载漂移故障、负载泄露故障三种

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\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，可以看到飞机随时间越来越不平衡



## 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\3.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: \*\*\*