Development and practice of intelligent unmanned cluster system

Full stack development case based on RflySim toolchain

Talk 7 Safety test and health assessment

飞思实验室

Outline

1. Experimental platform configuration

1.1 Components to be installed

- Visual Studio 2017 (installation required for both the Experience and full versions)
- Configure the C++ compiler for MATLAB (both the experience and full versions need to be installed)
- Matlab 2023a* (Advanced Full version installation)

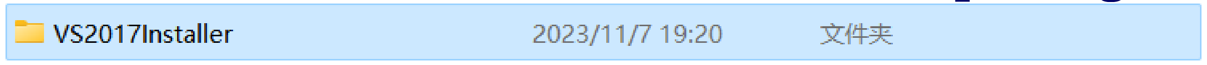
Here is how to install Visual Studio 2017 (requires networking) :

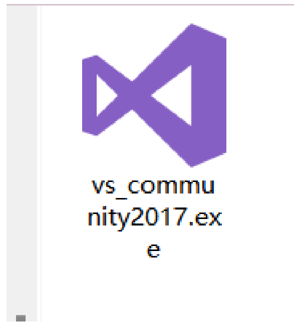
In this platform, the installation package of Visual Studio 2017 has been placed

1. Experimental platform configuration

1.2 How to install Visual Studio 2017

- First, we can open the location of platform installation, find *:\PX4PSP\ RflySimAPIs here, where are placed some routines in the platform and the installation package of the software
- After that, we can open the content of chapter 4 and find the basic version of the routine, 4.rflysimmodel \1.BasicExps, where we can find the folder named VS2017Installer, which is the installation package of Visual Studio 2017.


VS2017Installer 2023/11/7 19:20 文件夹

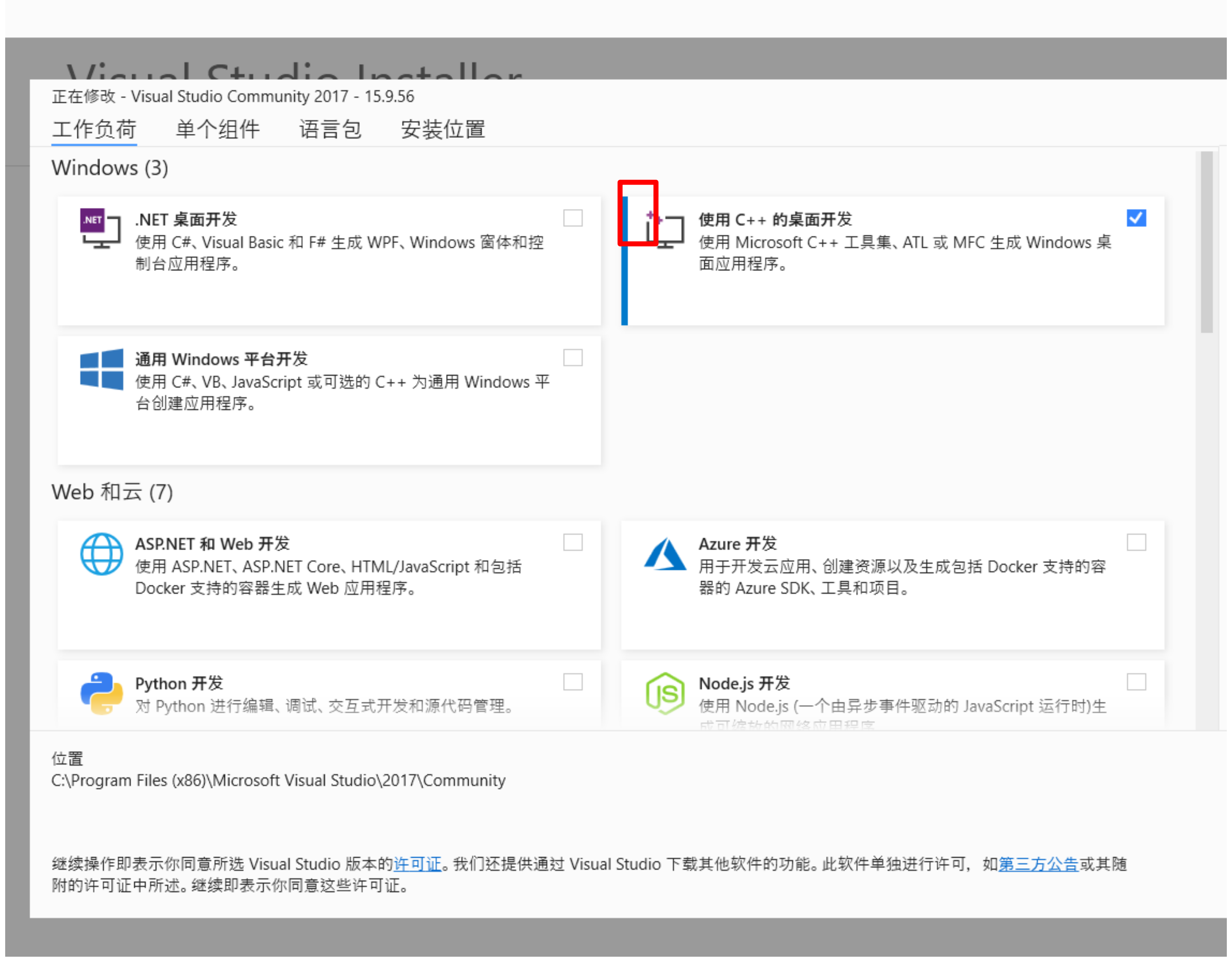


The online installation steps (Internet required) are as follows: double-click PX4PSP\ RflySimAPIs \4.RflySimModel\1.BasicExps\VS2017Installer\vs _community2017.exe

1. Experimental platform configuration

1.2 How to install Visual Studio 2017

- Install Visual Studio 2017 (you can also use other versions, MATLAB can recognize it).
- The Visual Studio compiler is needed in many places in the following courses, such as the use of MATLAB S-Function Builder module, and the automatic generation of C/C++ model code in Simulink
- For this course, just check "C++ desktop development" in the right picture.



Visual Studio Installer

正在修改 - Visual Studio Community 2017 - 15.9.56

工作负荷　单个组件　语言包　安装位置

Windows (3)

☐ .NET 桌面开发
使用 C#、Visual Basic 和 F# 生成 WPF、Windows 窗体和控制台应用程序。

☑ 使用 C++ 的桌面开发
使用 Microsoft C++ 工具集、ATL 或 MFC 生成 Windows 桌面应用程序。

☐ 通用 Windows 平台开发
使用 C#、VB、JavaScript 或可选的 C++ 为通用 Windows 平台创建应用程序。

Web 和云 (7)

☐ ASP.NET 和 Web 开发
使用 ASP.NET、ASP.NET Core、HTML/JavaScript 和包括 Docker 支持的容器生成 Web 应用程序。

☐ Azure 开发
用于开发云应用、创建资源以及生成包括 Docker 支持的容器的 Azure SDK、工具和项目。

☐ Python 开发
对 Python 进行编辑、调试、交互式开发和源代码管理。

☐ Node.js 开发
使用 Node.js (一个由异步事件驱动的 JavaScript 运行时)生成可缩放的网络应用程序

位置
C:\Program Files (x86)\Microsoft Visual Studio\2017\Community

继续操作即表示你同意所选 Visual Studio 版本的许可证。我们还提供通过 Visual Studio 下载其他软件的功能。此软件单独进行许可，如第三方公告或其随附的许可证中所述。继续即表示你同意这些许可证。

飞思实验室

1. Experimental platform configuration

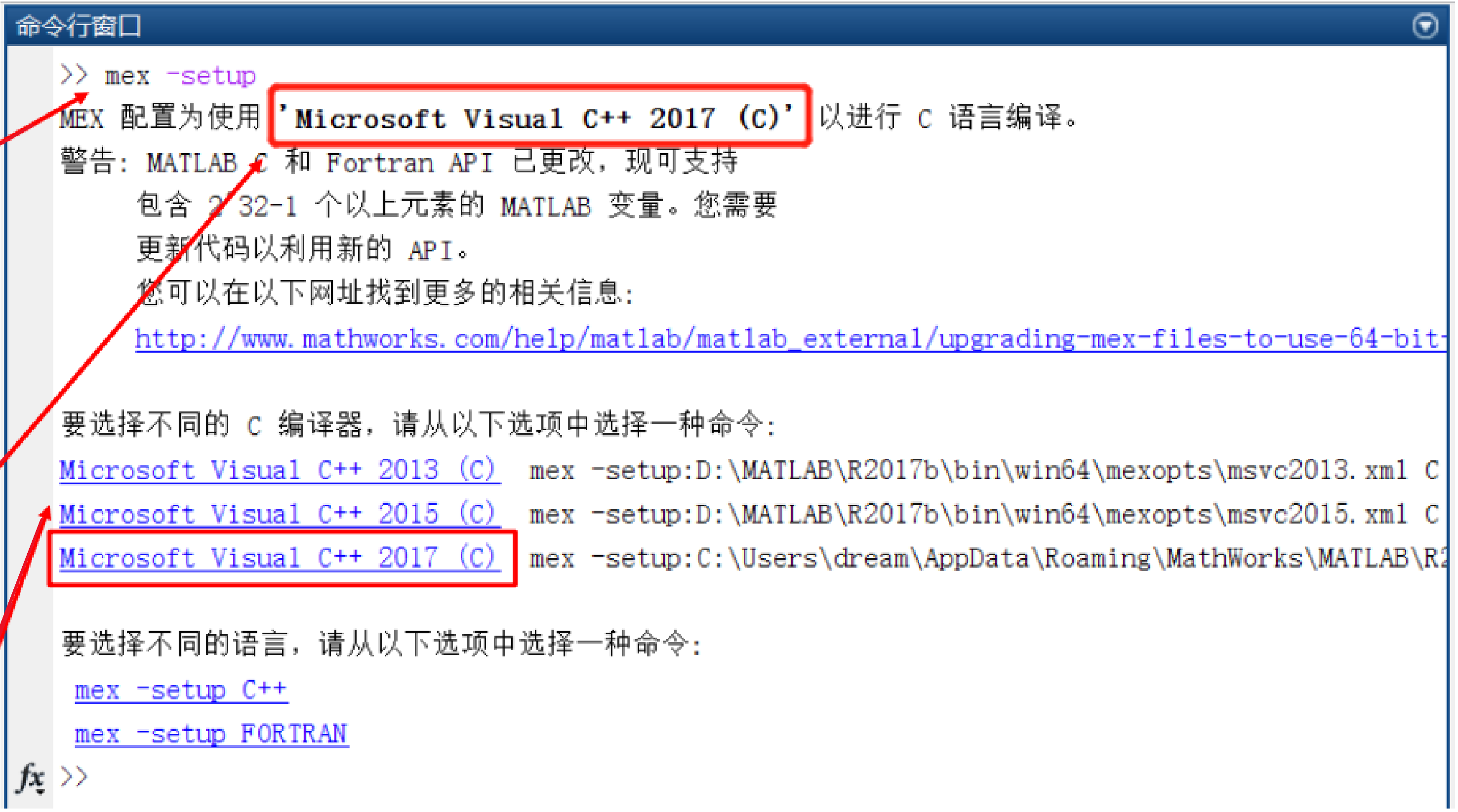
1.2 How to install Visual Studio 2017

- Note: VS2019 can also be installed on the advanced version of MATLAB, but MATLAB can only recognize the version of Visual Studio that is lower than its own, so MATLAB 2017b has no way to recognize VS2019.
- Note: Please do not change the default installation directory of VS (for example, to disk D), which will cause MATLAB to fail to recognize.
- Cannot use the Mingw compiler, requires VS

1. Experimental platform configuration

- **1.3 Configure the C++ compiler for MATLAB**

- **Enter the instruction "mex-setup" in the command line window of MATLAB**
- **In general, the VS 2017 compiler is automatically identified and installed, as shown in the image on the right, "MEX configuration uses' Microsoft Visual C++ 2017 'for compilation" indicates that the installation is correct**
- **If there are other compilers, this page can also be toggled to select another compiler such as VS 2013/2015**

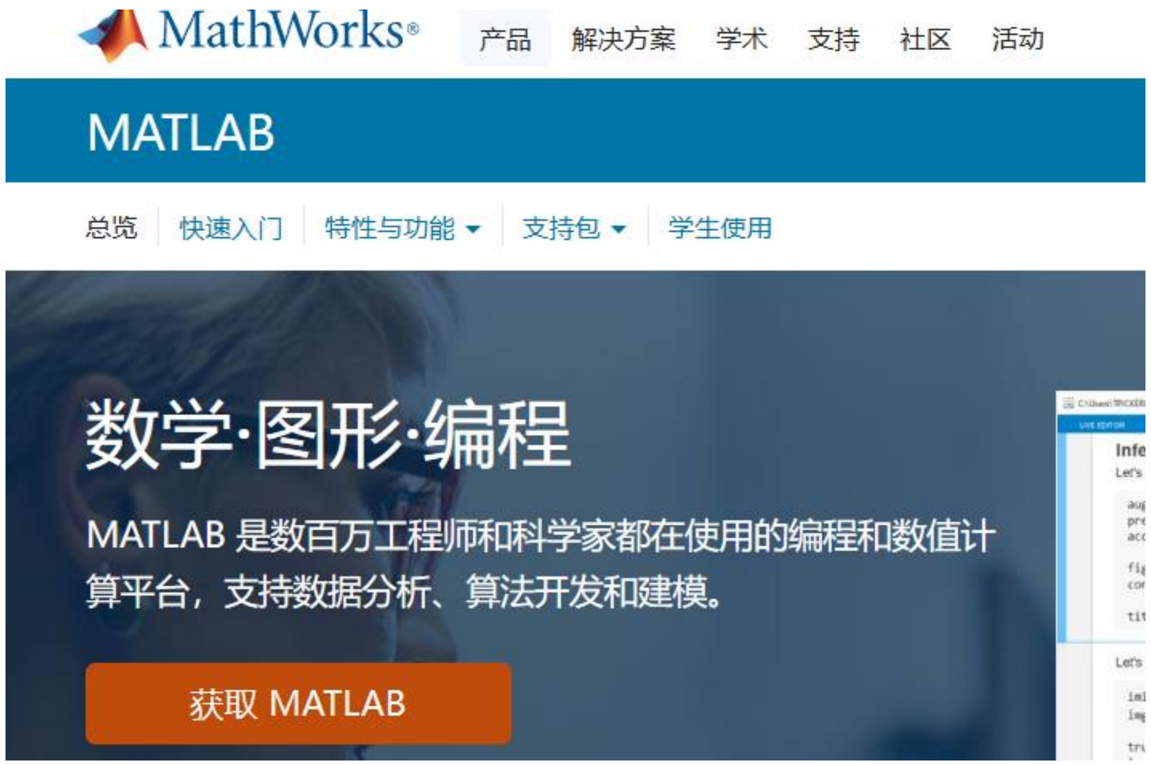


```
>> mex -setup
MEX 配置为使用 'Microsoft Visual C++ 2017 (C)' 以进行 C 语言编译。
警告: MATLAB C 和 Fortran API 已更改，现可支持
      包含 2^32-1 个以上元素的 MATLAB 变量。您需要
      更新代码以利用新的 API。
      您可以在以下网址找到更多的相关信息:
      http://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit

要选择不同的 C 编译器，请从以下选项中选择一种命令:
Microsoft Visual C++ 2013 (C)  mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2013.xml C
Microsoft Visual C++ 2015 (C)  mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2015.xml C
Microsoft Visual C++ 2017 (C)  mex -setup:C:\Users\dream\AppData\Roaming\MathWorks\MATLAB\R

要选择不同的语言，请从以下选项中选择一种命令:
 mex -setup C++
 mex -setup FORTRAN
fx >>
```

飞思实验室

1. Experimental platform configuration

- **1.4 Installation method of Matlab 2023a**
- **MATLAB installation package download path:**
- **https://ww2.mathworks.cn/products/matlab.html**



数学·图形·编程

MATLAB 是数百万工程师和科学家都在使用的编程和数值计算平台，支持数据分析、算法开发和建模。

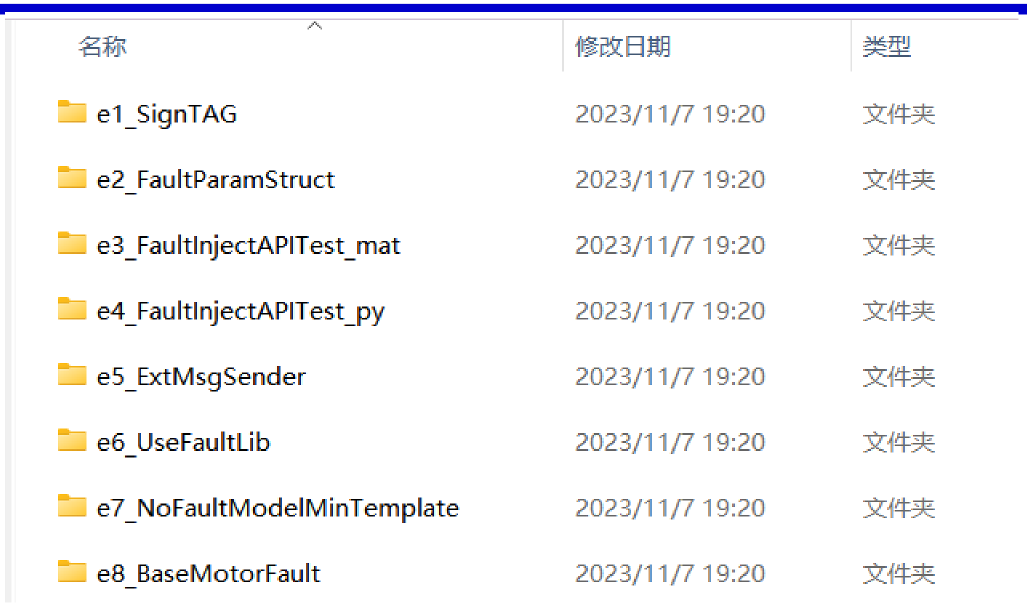
Outline

2. Key interface introduction

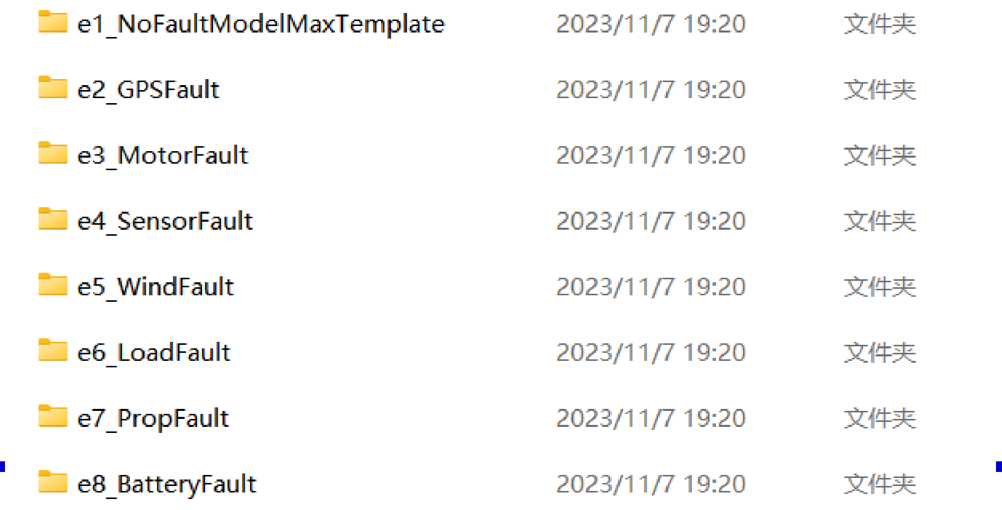
2.0 Basic experiment overview

Includes the basic functional interface "RflySimAPIs /7.RflySimPHM/0.ApiExps" and the basic routine "RflySimAPIs \7.RflySimPHM\1.BasicExps"

See API.pdf and Readme.pdf for details



名称	修改日期	类型
e1_SignTAG	2023/11/7 19:20	文件夹
e2_FaultParamStruct	2023/11/7 19:20	文件夹
e3_FaultInjectAPITest_mat	2023/11/7 19:20	文件夹
e4_FaultInjectAPITest_py	2023/11/7 19:20	文件夹
e5_ExtMsgSender	2023/11/7 19:20	文件夹
e6_UseFaultLib	2023/11/7 19:20	文件夹
e7_NoFaultModelMinTemplate	2023/11/7 19:20	文件夹
e8_BaseMotorFault	2023/11/7 19:20	文件夹

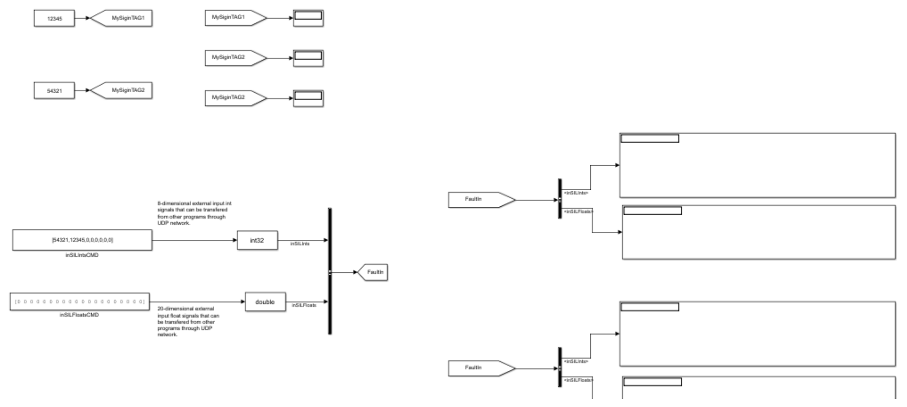


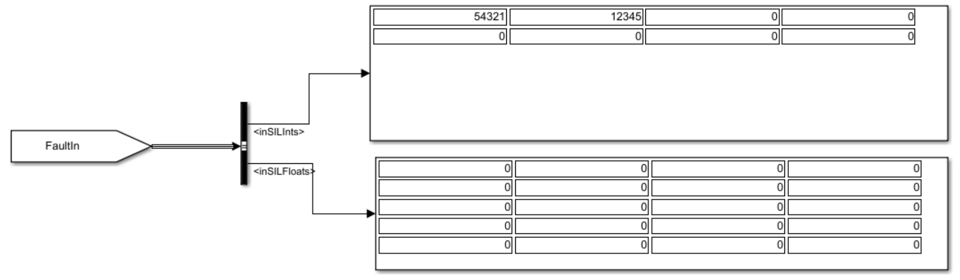
e1_NoFaultModelMaxTemplate	2023/11/7 19:20	文件夹
e2_GPSFault	2023/11/7 19:20	文件夹
e3_MotorFault	2023/11/7 19:20	文件夹
e4_SensorFault	2023/11/7 19:20	文件夹
e5_WindFault	2023/11/7 19:20	文件夹
e6_LoadFault	2023/11/7 19:20	文件夹
e7_PropFault	2023/11/7 19:20	文件夹
e8_BatteryFault	2023/11/7 19:20	文件夹

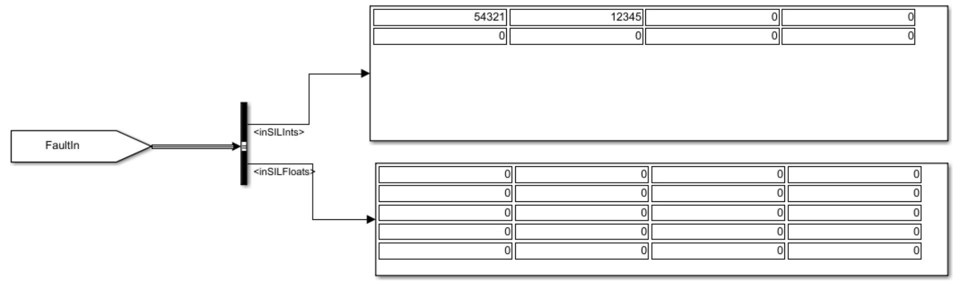
2. Key interface introduction

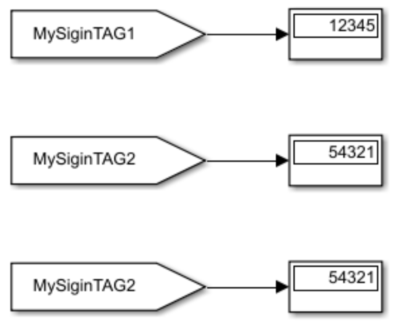
2.1 Study and use of signal label module

Learn the use of Goto and From modules through this routine. See 0.ApiExps\e1_SignTAG\readme.pdf for detailed operation and experimental results







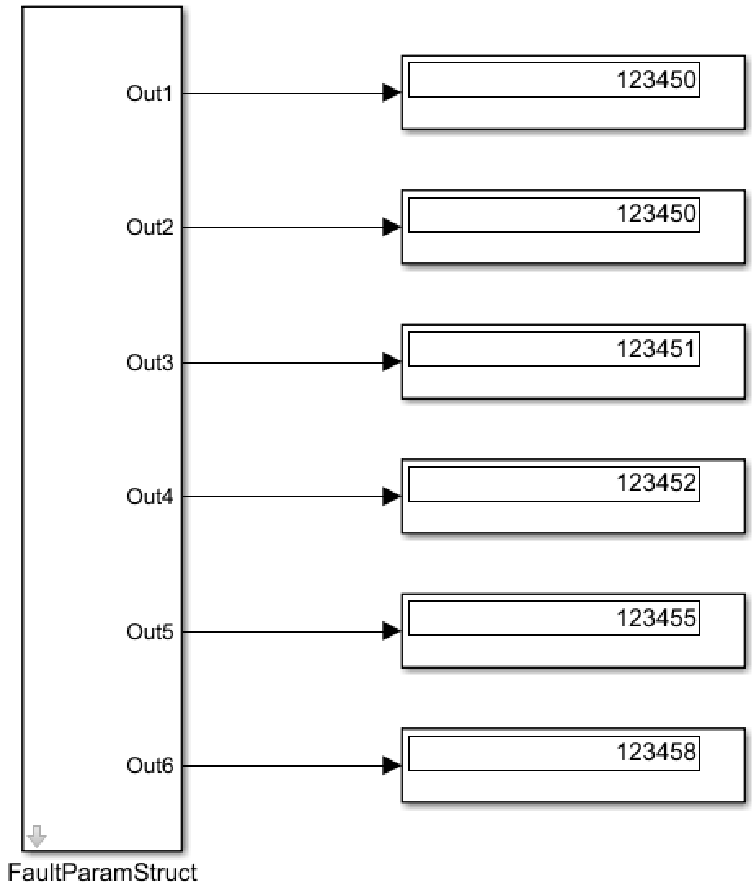


2. Key interface introduction

2.2 Learning and using fault parameters and module encapsulation parameter reference

Learn to read the required fault parameters from the workspace by creating the encapsulation parameters.

See 0.ApiExps\e2_FaultParamStruct\readme.pdf for details and experimental results



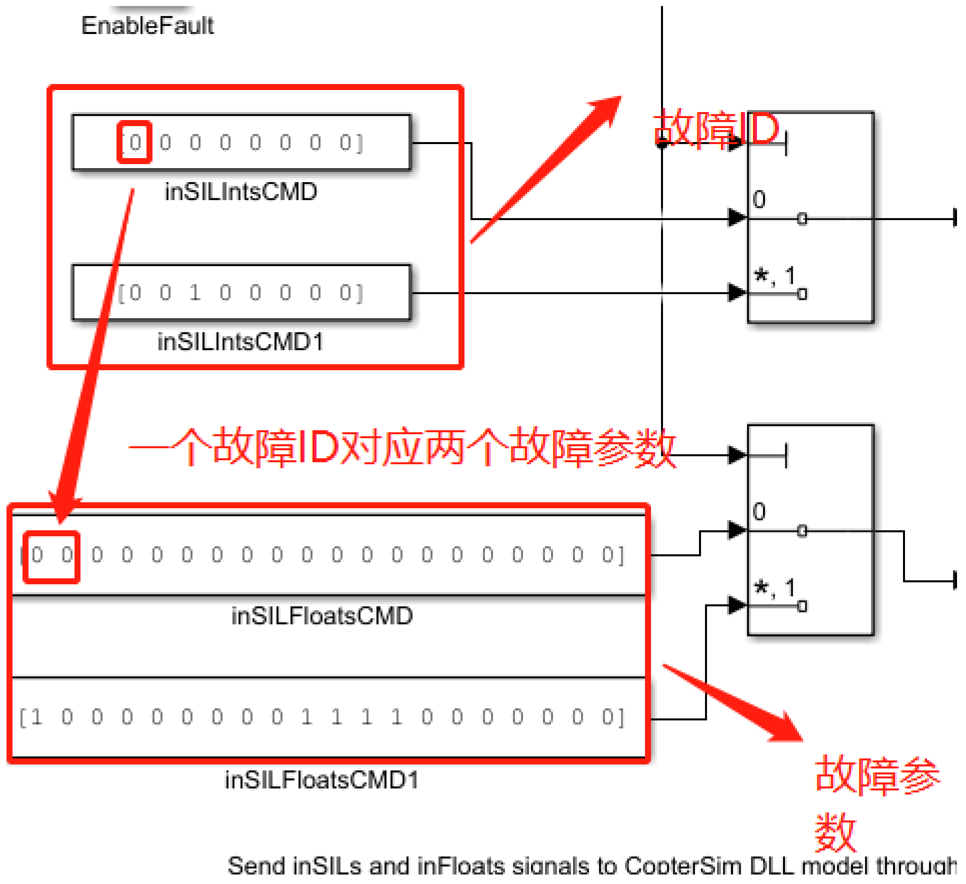
飞思实验室

2. Key interface introduction

2.3 Learning and using the fault injection parameter module in UDP mode

Learn how to send fault injection parameter code using UDP mode through this routine.

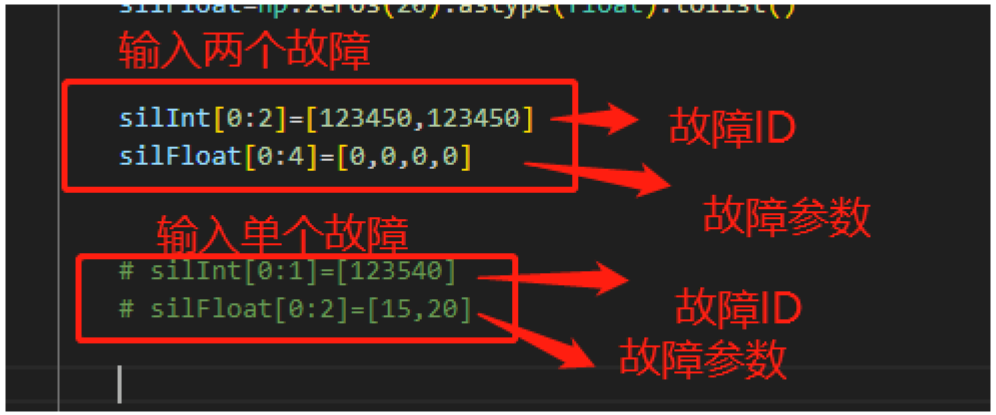
See 0.ApiExps\e3_FaultInjectAPITest_mat\readme.pdf for specific experiment operation and effect



2. Key interface introduction

2.4 Learning and using the fault injection parameter module in UDP mode

Learn how to use UDP mode to send fault injection parameter code through this routine. Part of the experimental effect as the figure on the right, detailed operation and experimental effect see 0.ApiExps\e4_FaultInjectAPITest_py\readme.pdf



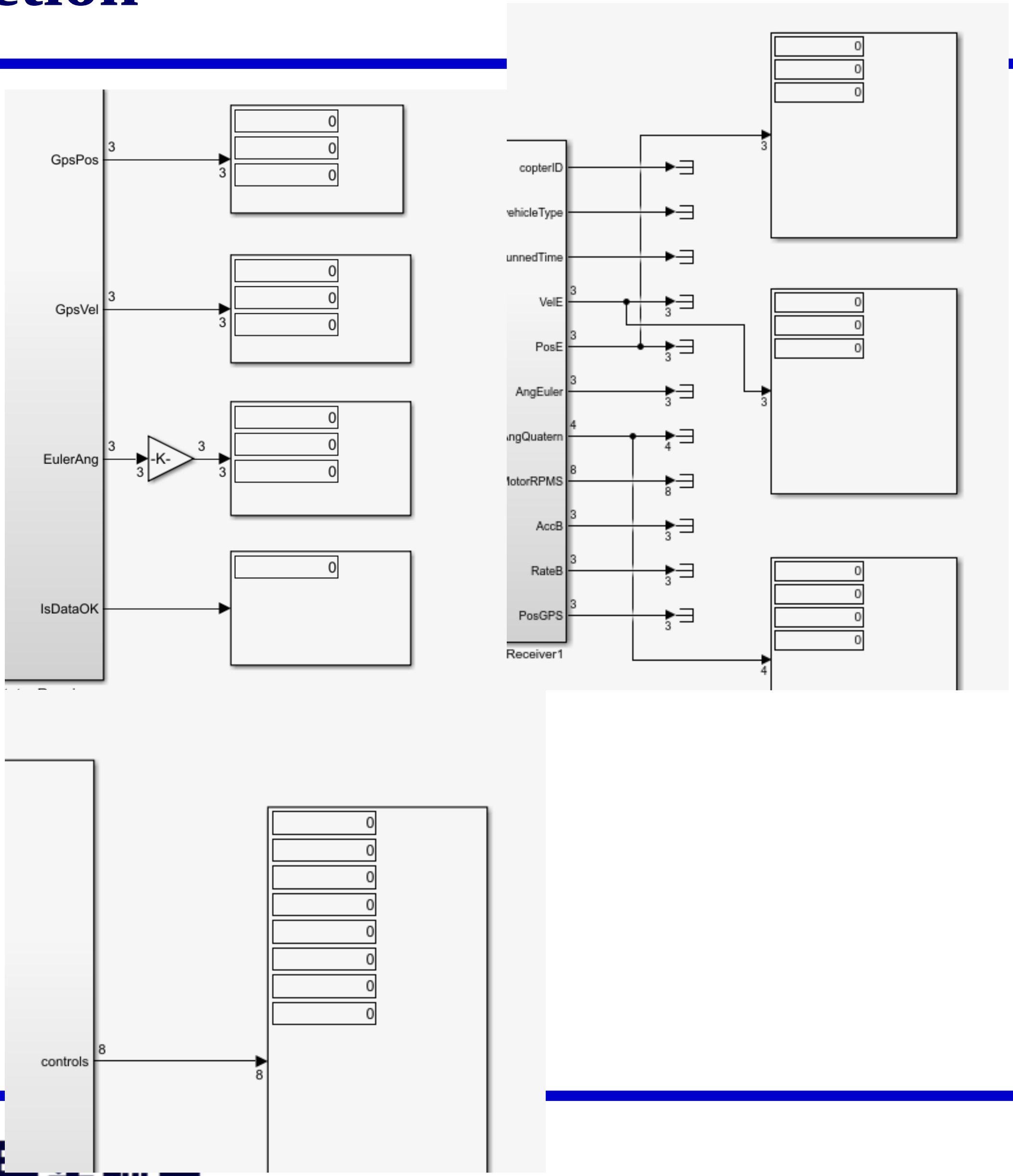
```
silFloat=np.zeros(20).astype(float).tolist()

silInt[0:2]=[123450,123450]
silFloat[0:4]=[0,0,0,0]

# silInt[0:1]=[123540]
# silFloat[0:2]=[15,20]
```

2. Key interface introduction

2.5 Learning and use of PX4 external message sending and receiving module

Learn how to send external messages to the interface of PX4 and receive PX4 status information through this routine.

See 0.ApiExps\e5_ExtMsgSender\readme.pdf for the specific experiment operation and effect

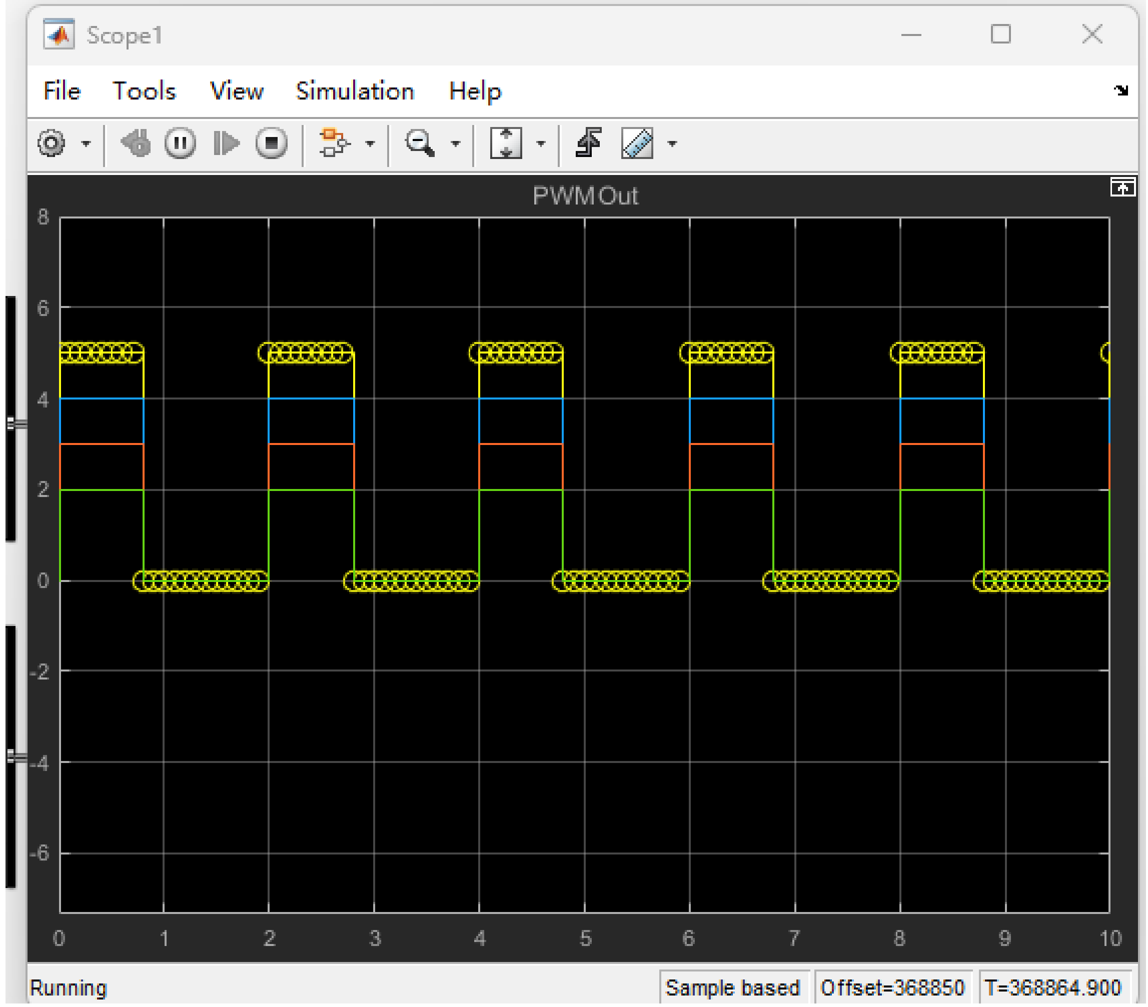


2. Key interface introduction

2.6 Learning and application of motor fault modeling principle

Through this routine learning from 0 to 1 using a motor fault injection module learning and use.

See file 0.ApiExps\e6_UseFaultLib\readme.pdf for specific experiment operation

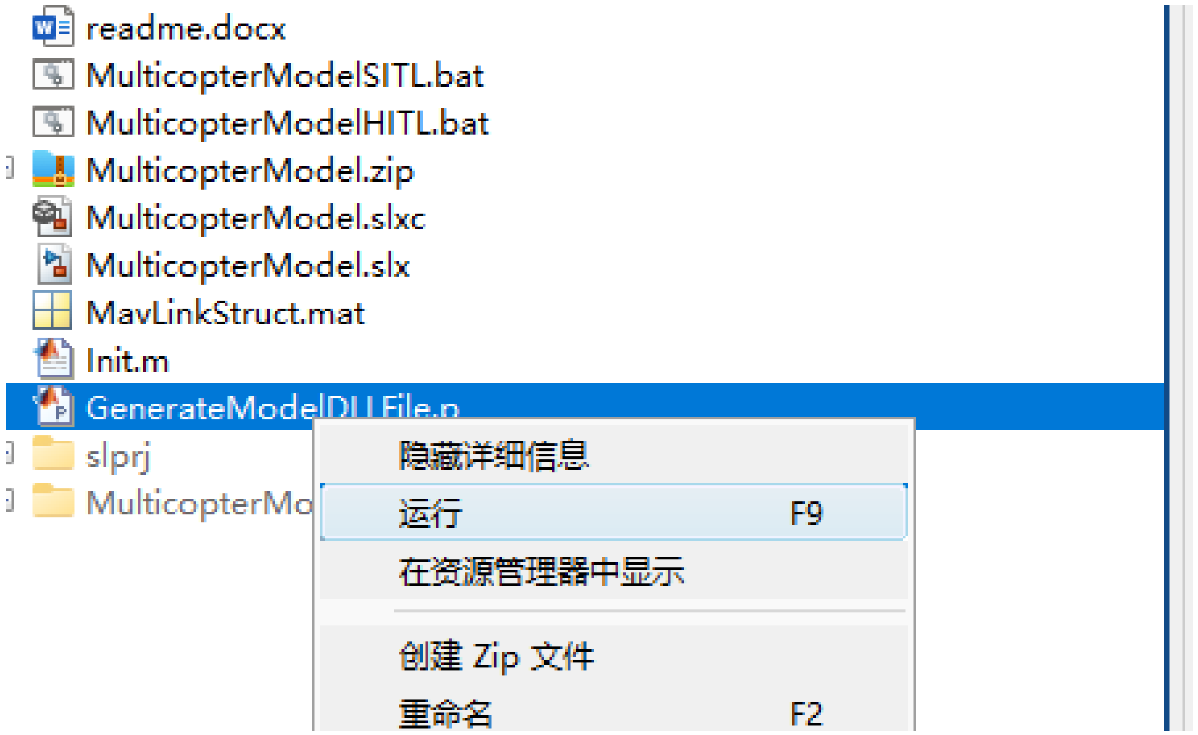


2. Key interface introduction

2.7 Minimum template of the fault injection module

Learn how to generate DLL files by learning the use of the fault injection minimum template with this routine.

See file [0.ApiExps\e7_NoFaultModelMinTemplate\readme.pdf] for specific experiment operation

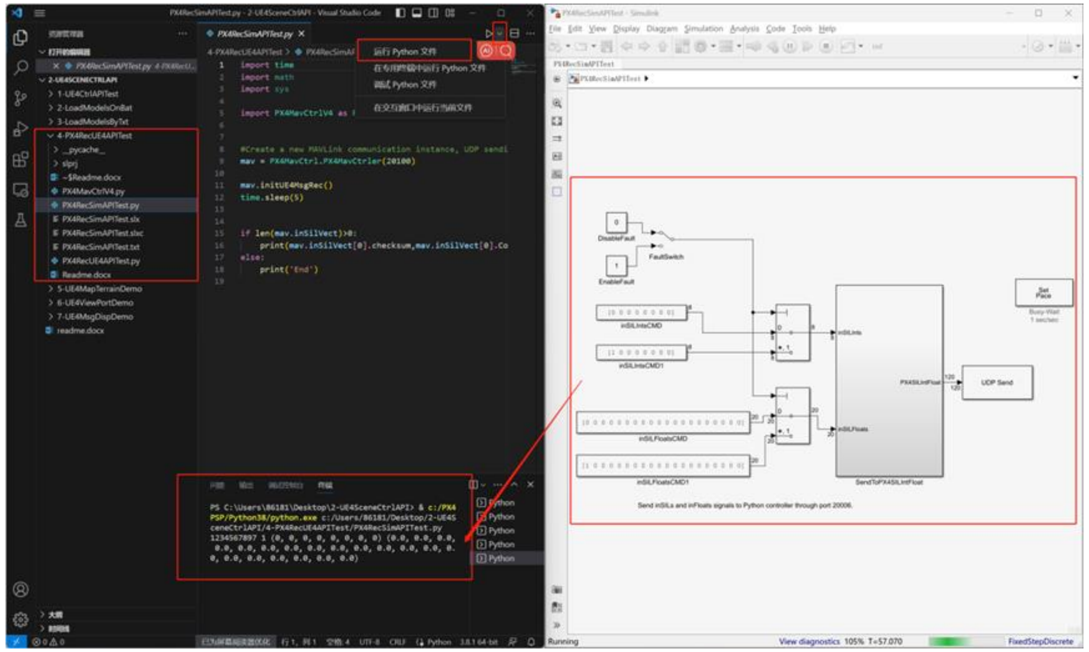


2. Key interface introduction

2.8 Learning and application of motor fault injection based on minimum template

Learn to use minimum template-based motor fault injection through this routine.

See file [0.ApiExps\e8_BaseMotorFault\readme.pdf] for specific experimental operations

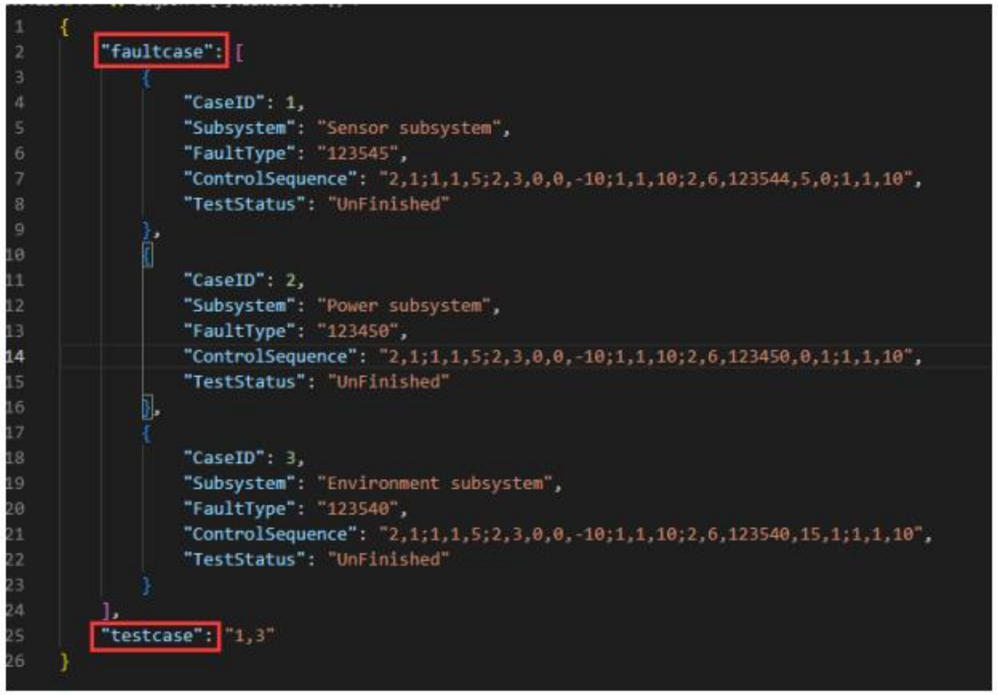


```
import time
import math
import sys

import PX4MavCtrlV4 as P


#Create a new MAVLink communication instance, UDP sendi
mav = PX4MavCtrl.PX4MavCtrler(20100)

mav.initUE4MsgRec()
time.sleep(5)


if len(mav.inSilVect)>0:
    print(mav.inSilVect[0].checksum,mav.inSilVect[0].Co
else:
    print('End')
```

```
PS C:\Users\86181\Desktop\2-UE4SceneCtrlAPI> & c:/PX4PSP/Python38/python.exe c:/Users/86181/Desktop/2-UE4SceneCtrlAPI/4-PX4RecUE4APITest/PX4RecSimAPITest.py
1234567897 1 (0, 0, 0, 0, 0, 0, 0, 0) (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
```

2. Key interface introduction

- **2.9 Data Collection**
- **Data collection facilitates training and verification of the training model.**
- **See file 0.ApiExps\e9_data_collect\readme.pdf for specific experimental operation**

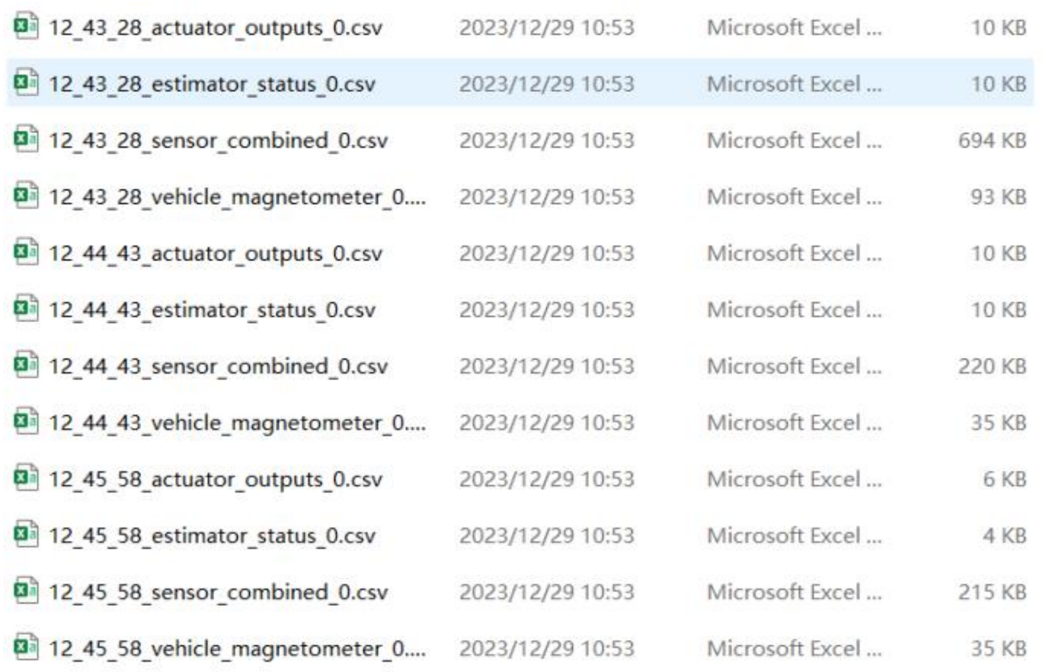


```
{
    "faultcase": [
        {
            "CaseID": 1,
            "Subsystem": "Sensor subsystem",
            "FaultType": "123545",
            "ControlSequence": "2,1;1,1,5;2,3,0,0,-10;1,1,10;2,6,123544,5,0;1,1,10",
            "TestStatus": "UnFinished"
        },
        {
            "CaseID": 2,
            "Subsystem": "Power subsystem",
            "FaultType": "123450",
            "ControlSequence": "2,1;1,1,5;2,3,0,0,-10;1,1,10;2,6,123450,0,1;1,1,10",
            "TestStatus": "UnFinished"
        },
        {
            "CaseID": 3,
            "Subsystem": "Environment subsystem",
            "FaultType": "123540",
            "ControlSequence": "2,1;1,1,5;2,3,0,0,-10;1,1,10;2,6,123540,15,1;1,1,10",
            "TestStatus": "UnFinished"
        }
    ],
    "testcase": "1,3"
}
```

飞思实验室

2. Key interface introduction

- **2.10 Data Processing**
- **Process the collected data and generate Excel tables.**
- **See file 0.ApiExps\e10_data_process\readme.pdf for specific experimental operation**

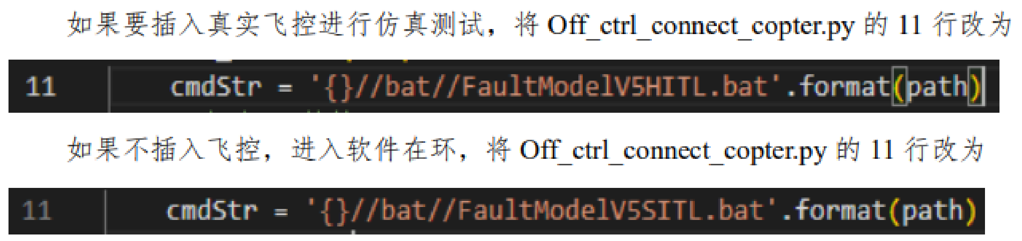


12_43_28_actuator_outputs_0.csv	2023/12/29 10:53	Microsoft Excel ...	10 KB
12_43_28_estimator_status_0.csv	2023/12/29 10:53	Microsoft Excel ...	10 KB
12_43_28_sensor_combined_0.csv	2023/12/29 10:53	Microsoft Excel ...	694 KB
12_43_28_vehicle_magnetometer_0....	2023/12/29 10:53	Microsoft Excel ...	93 KB
12_44_43_actuator_outputs_0.csv	2023/12/29 10:53	Microsoft Excel ...	10 KB
12_44_43_estimator_status_0.csv	2023/12/29 10:53	Microsoft Excel ...	10 KB
12_44_43_sensor_combined_0.csv	2023/12/29 10:53	Microsoft Excel ...	220 KB
12_44_43_vehicle_magnetometer_0....	2023/12/29 10:53	Microsoft Excel ...	35 KB
12_45_58_actuator_outputs_0.csv	2023/12/29 10:53	Microsoft Excel ...	6 KB
12_45_58_estimator_status_0.csv	2023/12/29 10:53	Microsoft Excel ...	4 KB
12_45_58_sensor_combined_0.csv	2023/12/29 10:53	Microsoft Excel ...	215 KB
12_45_58_vehicle_magnetometer_0....	2023/12/29 10:53	Microsoft Excel ...	35 KB

飞思实验室

2. Key interface introduction

- **2.11 Model verification**
- **Validation of the training model.**
- **See file [0.ApiExps\e11_moder_ver\readme.pdf](0.ApiExps\e11_moder_ver\readme.pdf) for specific experimental operation**


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

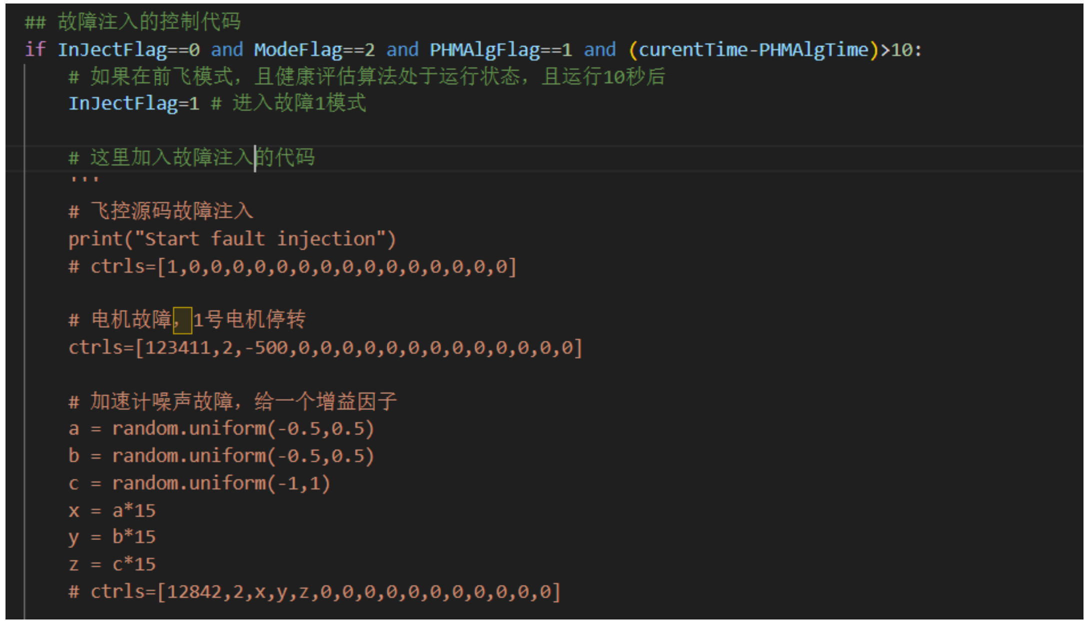
```
11        cmdStr = '{}//bat//FaultModelV5HITL.bat'.format(path)
```

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)
```

2. Key interface introduction

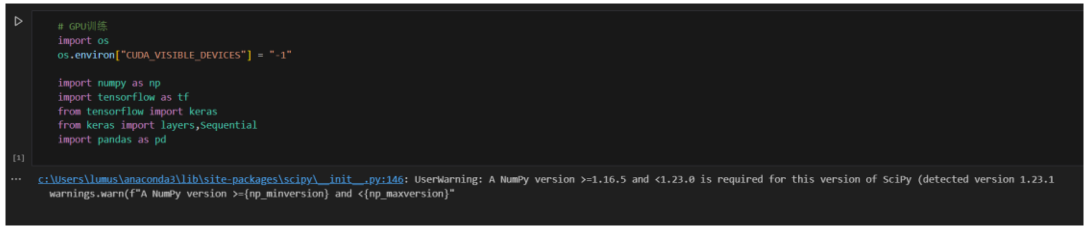
- **2.12 Fault Diagnosis**
- **Through the hardware and software in the loop, according to the change of the position of the aircraft fault judgment.**
- **See file 0.ApiExps\e12_health_basic\readme.pdf for specific experimental operation**



```
## 故障注入的控制代码
if InJectFlag==0 and ModeFlag==2 and PHMAlgFlag==1 and (curentTime-PHMAlgTime)>10:
    # 如果在前飞模式，且健康评估算法处于运行状态，且运行10秒后
    InJectFlag=1 # 进入故障1模式

    # 这里加入故障注入的代码
    '''
    # 飞控源码故障注入
    print("Start fault injection")
    # ctrls=[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

    # 电机故障，1号电机停转
    ctrls=[123411,2,-500,0,0,0,0,0,0,0,0,0,0,0,0,0]

    # 加速计噪声故障，给一个增益因子
    a = random.uniform(-0.5,0.5)
    b = random.uniform(-0.5,0.5)
    c = random.uniform(-1,1)
    x = a*15
    y = b*15
    z = c*15
    # ctrls=[12842,2,x,y,z,0,0,0,0,0,0,0,0,0,0,0]
```

2. Key interface introduction

- **2.13 Neural network training model**
- **In order to verify the health of the model**
- **See file C:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e13_model_train\readme.pdf for detailed experimental operation**



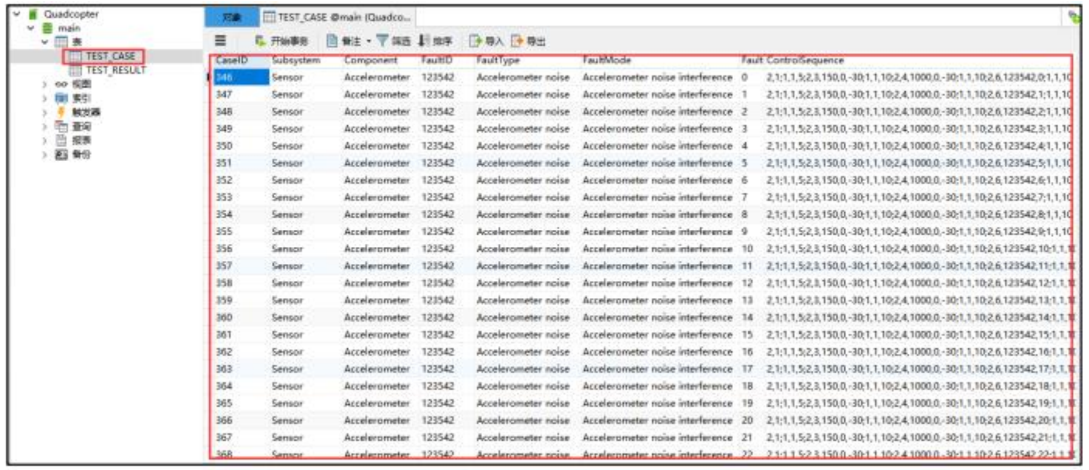
```
# GPU训练
import os
os.environ["CUDA_VISIBLE_DEVICES"] = "-1"

import numpy as np
import tensorflow as tf
from tensorflow import keras
from keras import layers,Sequential
import pandas as pd
```

```
[1]
c:\Users\lumus\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

飞思实验室

2. Key interface introduction

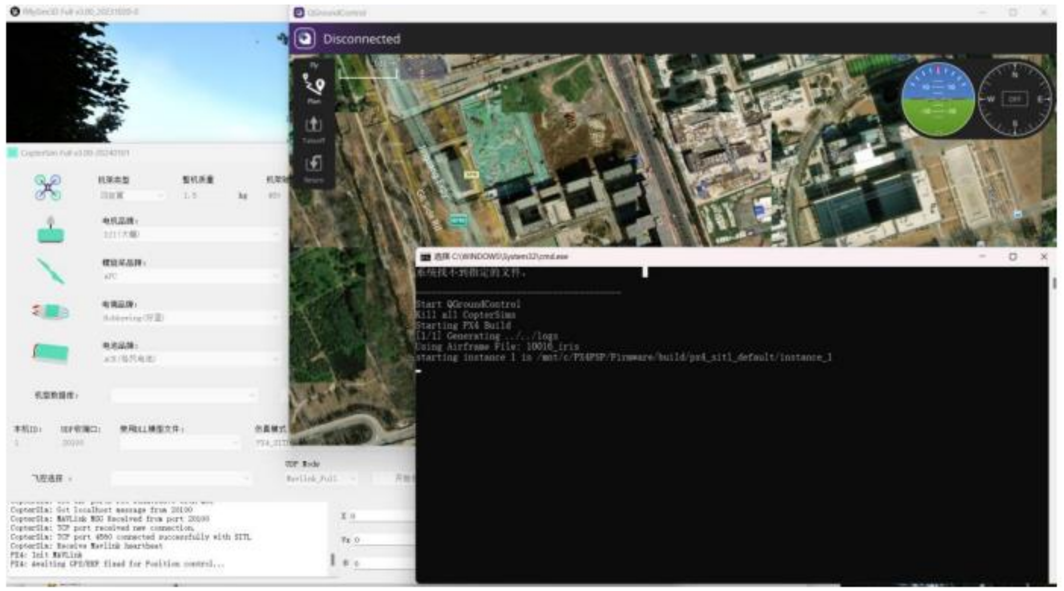
- **2.14 Automatic test case generation and import**
- **Master the basic use and configuration of test cases.**
- **See file [0.ApiExps\e14_DBExp\readme.pdf] for specific experiment operation**



CaseID	Subsystem	Component	FaultID	FaultType	FaultMode	Fault	ControlSequence
346	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	0	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,0;1,1,10
347	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	1	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,1;1,1,10
348	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	2	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,2;1,1,10
349	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	3	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,3;1,1,10
350	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	4	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,4;1,1,10
351	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	5	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,5;1,1,10
352	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	6	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,6;1,1,10
353	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	7	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,7;1,1,10
354	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	8	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,8;1,1,10
355	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	9	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,9;1,1,10
356	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	10	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,10;1,1,1
357	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	11	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,11;1,1,1
358	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	12	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,12;1,1,1
359	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	13	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,13;1,1,1
360	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	14	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,14;1,1,1
361	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	15	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,15;1,1,1
362	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	16	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,16;1,1,1
363	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	17	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,17;1,1,1
364	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	18	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,18;1,1,1
365	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	19	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,19;1,1,1
366	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	20	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,20;1,1,1
367	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	21	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,21;1,1,1
368	Sensor	Accelerometer	123542	Accelerometer noise	Accelerometer noise interference	22	2,1;1,1,5;2,3,150,0,-30;1,1,10;2,4,1000,0,-30;1,1,10;2,6,123542,22;1,1,1

2. Key interface introduction

- **2.15 Custom control sequence configuration and use**
- **Master the basic use and configuration of custom control sequences.**
- **See file [0.ApiExps\e15_CmdExp\readme.pdf](0.ApiExps\e15_CmdExp\readme.pdf) for specific experiment operation**

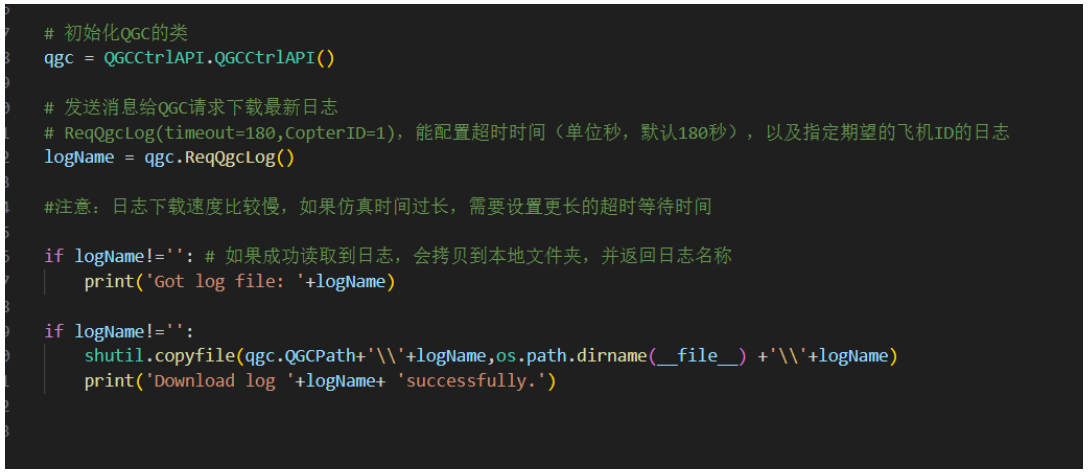


```
系统找不到指定的文件。
----------------------------------------
Start QGroundControl
Kill all CopterSims
Starting PX4 Build
[1/1] Generating ../../logs
Using Airframe File: 10016_iris
starting instance 1 in /mnt/c/PX4PSP/Firmware/build/px4_sitl_default/instance_1
```

```
CopterSim: Got localhost message from 20100
CopterSim: MAVLink MSG Received from port 20100
CopterSim: TCP port received new connection.
CopterSim: TCP port 4560 connected successfully with SITL
CopterSim: Receive Mavlink heartbeat
PX4: Init MAVLink
PX4: Awaiting GPS/EKF fixed for Position control...
```

2. Key interface introduction

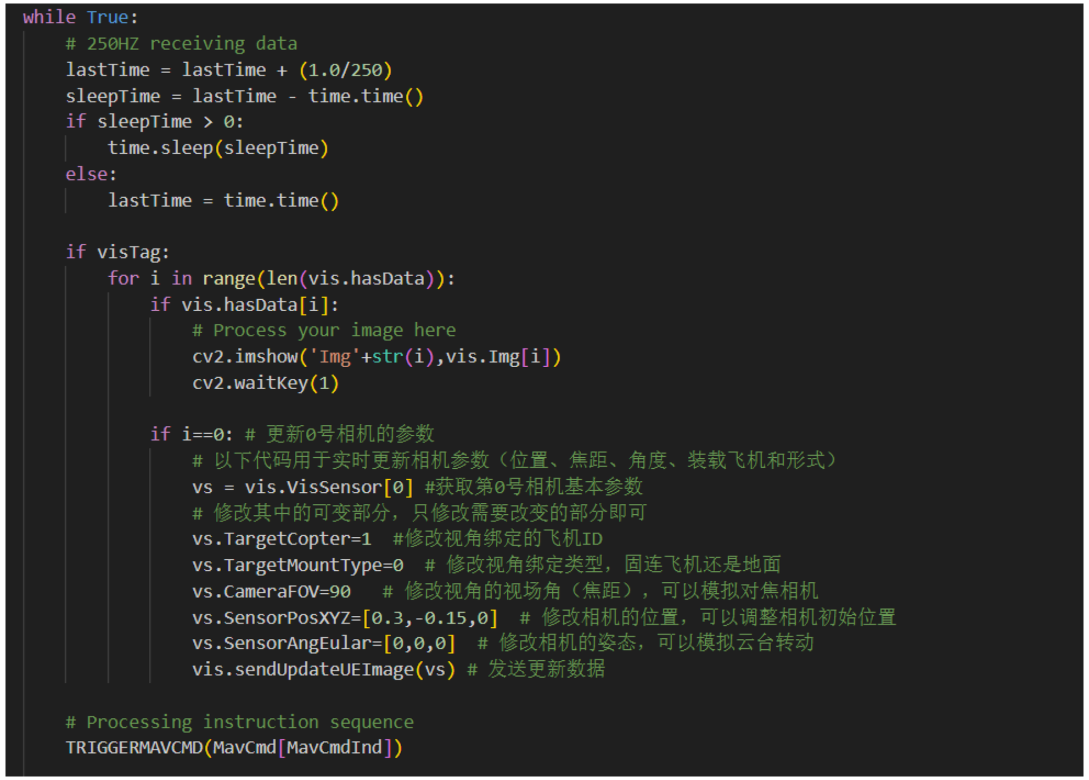
- **2.16 Use of automatic log download**
- **Master automatic log download and configuration.**
- **See file [0.ApiExps\e16_QGCLoadExp\readme.pdf](0.ApiExps\e16_QGCLoadExp\readme.pdf) for specific experiment operation**



```
# 初始化QGC的类
qgc = QGCCtrlAPI.QGCCtrlAPI()

# 发送消息给QGC请求下载最新日志
# ReqQgcLog(timeout=180,CopterID=1)，能配置超时时间（单位秒，默认180秒），以及指定期望的飞机ID的日志
logName = qgc.ReqQgcLog()

#注意：日志下载速度比较慢，如果仿真时间过长，需要设置更长的超时等待时间

if logName!='': # 如果成功读取到日志，会拷贝到本地文件夹，并返回日志名称
    print('Got log file: '+logName)

if logName!='':
    shutil.copyfile(qgc.QGCPath+'\\'+logName,os.path.dirname(__file__) +'\\'+logName)
    print('Download log '+logName+ 'successfully.')
```

2. Key interface introduction

- **2.17 Preparation and use of custom control sequence with vision**
- **Master the formulation and use of visual custom control sequences.**
- **See file 0.ApiExps\e17_VisCmdExp\readme.pdf for specific experiment operation**



```
while True:
    # 250HZ receiving data
    lastTime = lastTime + (1.0/250)
    sleepTime = lastTime - time.time()
    if sleepTime > 0:
        time.sleep(sleepTime)
    else:
        lastTime = time.time()

    if visTag:
        for i in range(len(vis.hasData)):
            if vis.hasData[i]:
                # Process your image here
                cv2.imshow('Img'+str(i),vis.Img[i])
                cv2.waitKey(1)

            if i==0: # 更新0号相机的参数
                # 以下代码用于实时更新相机参数（位置、焦距、角度、装载飞机和形式）
                vs = vis.VisSensor[0] #获取第0号相机基本参数
                # 修改其中的可变部分，只修改需要改变的部分即可
                vs.TargetCopter=1  #修改视角绑定的飞机ID
                vs.TargetMountType=0  # 修改视角绑定类型，固连飞机还是地面
                vs.CameraFOV=90   # 修改视角的视场角（焦距），可以模拟对焦相机
                vs.SensorPosXYZ=[0.3,-0.15,0]  # 修改相机的位置，可以调整相机初始位置
                vs.SensorAngEular=[0,0,0]  # 修改相机的姿态，可以模拟云台转动
                vis.sendUpdateUEImage(vs) # 发送更新数据

    # Processing instruction sequence
    TRIGGERMAVCMD(MavCmd[MavCmdInd])
```

飞思实验室

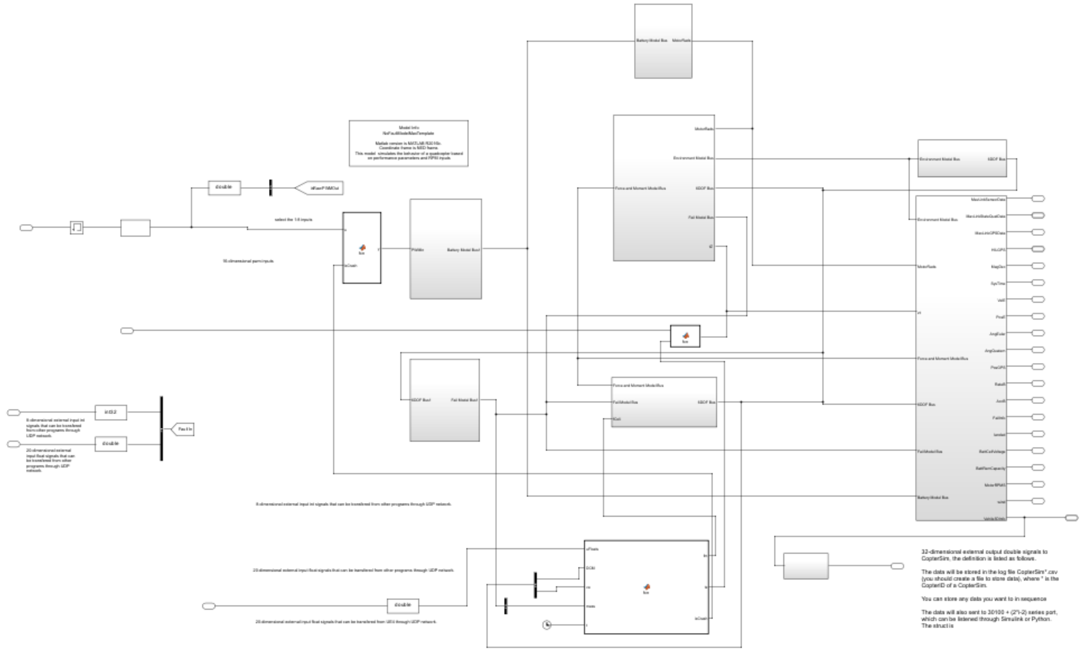
Outline

3. Basic experiment case

3.1 Maximum template of fault injection module

This experiment did not produce any fault effect, just a maximum te mplate without any fault that can be replaced by any fault injection mod ule. The difference between the mini mum template and the maximum te mplate is that the minimum templat e has no external fault injection inte rface, and the maximum template h as an external fault injection interfa ce.

See file 1.BasicExps\e1_NoFault ModelMaxTemplate\readme.pdf for details

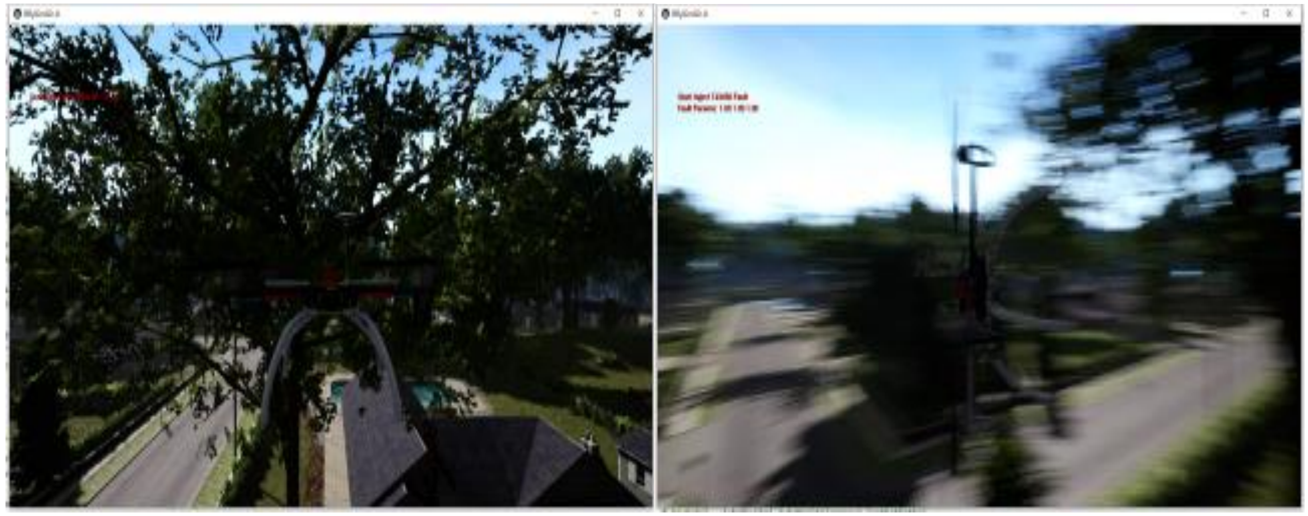


3. Basic experiment case

3.2 Principle of GPS module fault injection based on maximum template

The fault modeling of GPS module is carried out based on the maximum template. The fault modeling model is exported as DLL file, and then the DLL file is loaded through CopterSim. Finally, the fault code is injected through udp mode for fault injection simulation.

See file 1.BasicExps\e2_GPSFault\Readme.pdf for specific experimental operation

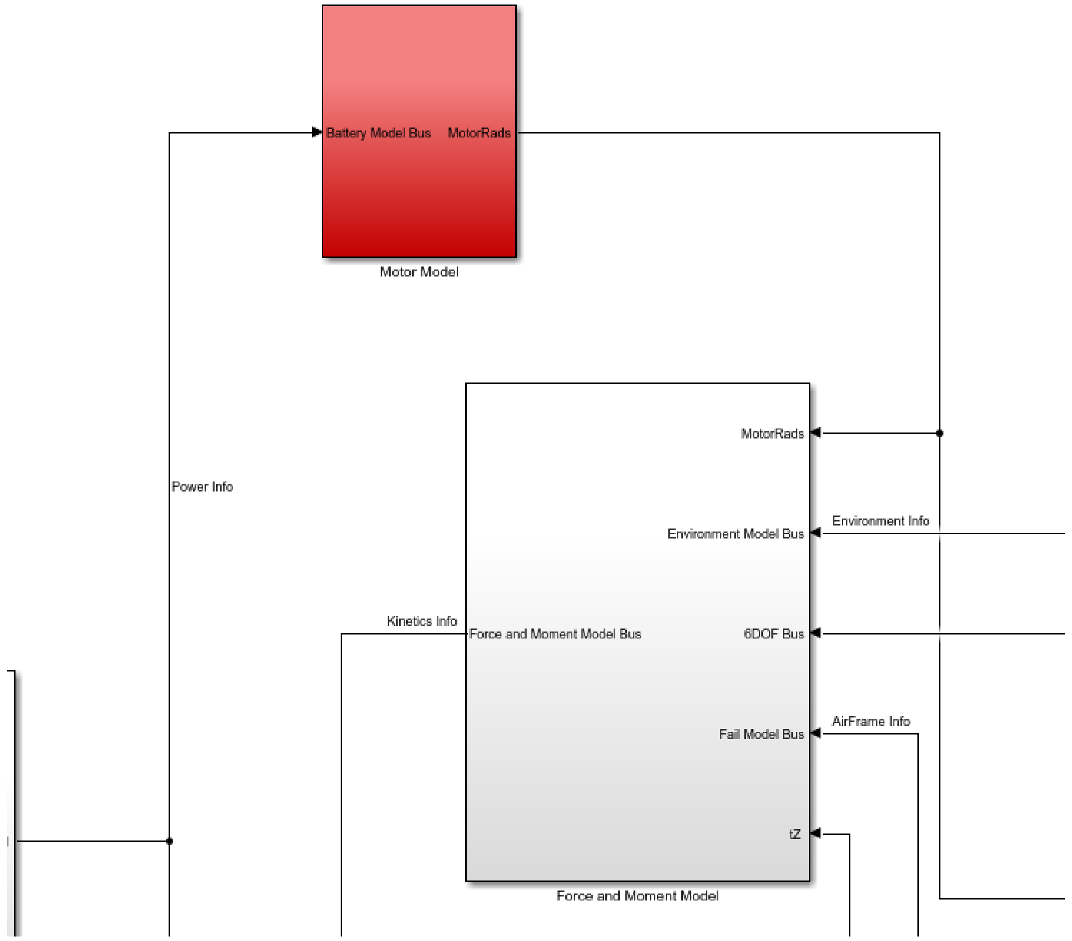


3. Basic experimental cases

3.3 Principle of motor module fault injection based on maximum template

The motor module fault modeling is carried out based on the maximum template, and the fault modeling model is exported as DLL file, and then the DLL file is loaded through CopterSim. Finally, the fault code is injected through udp mode for fault injection simulation.

See file 1.BasicExps\e3_MotorFault\Readme.pdf for specific experimental operation

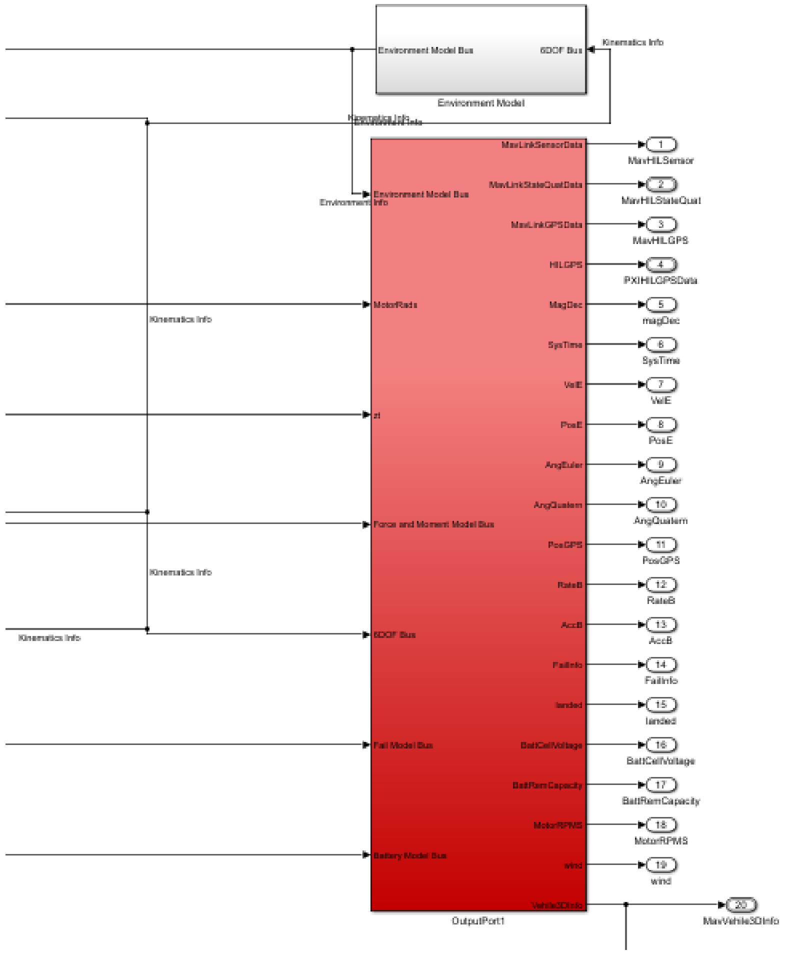


3. Basic experiment case

3.4 Principle of sensor module fault injection based on maximum template

For the fault modeling of the sensor module of the maximum template, the fault modeling model is exported as a DLL file, and then the DLL file is loaded through CopterSim. Finally, the fault code is injected through udp mode (python/ matlab form) for fault injection simulation.

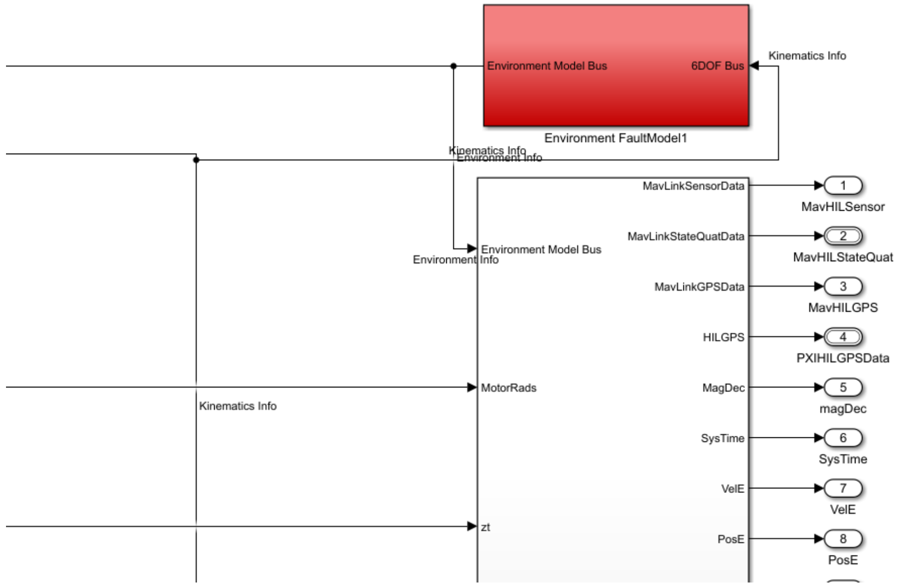
See file 1.BasicExps\e4_SensorFault\Readme.pdf for specific experimental operation



3. Basic experiment case

3.5 Principle of fault injection of ambient air module based on maximum template

Modeling the fault of the environ mental wind module of the maximum template, exporting the fault modelin g model as a DLL file, loading the DL L file through CopterSim, and finally injecting fault code through udp mod e (python/ matlab form) for fault inje ction simulation. Specific experiment al operation see file [1.BasicExps\e5_ WindFault\Readme.pdf]

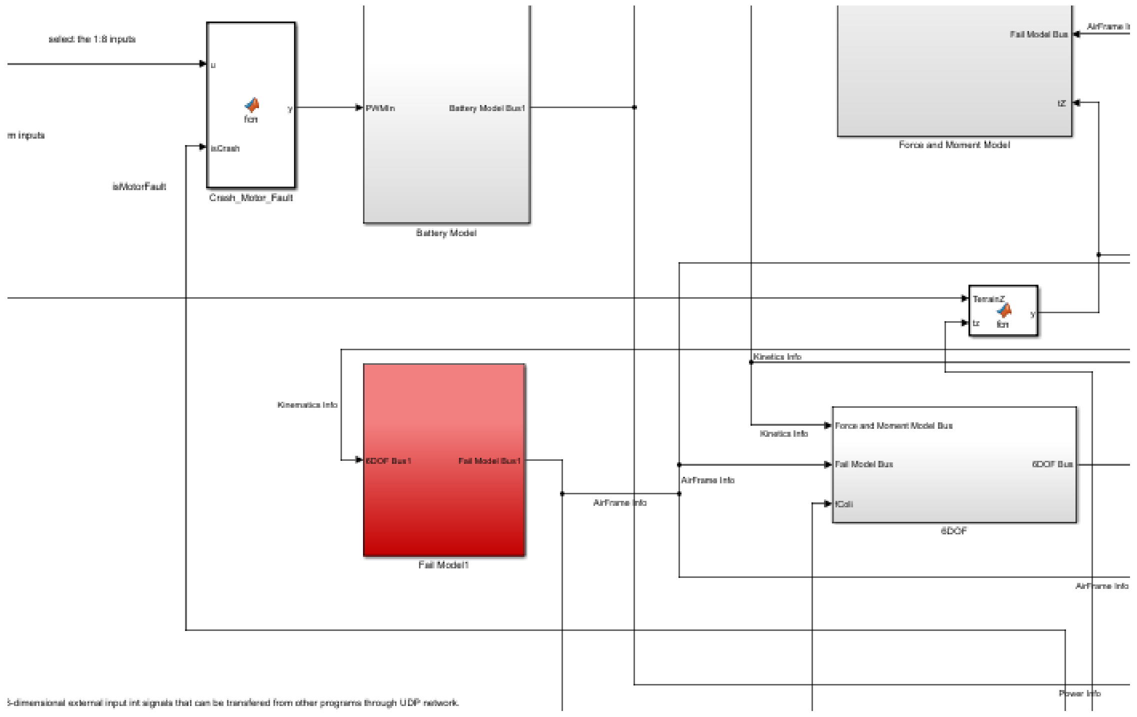


3. Basic experiment case

3.6 Principle of fault injection of load module based on maximum template

The fault model of the load module of the maximum template was exported as a DLL file, and the DLL file was loaded through CopterSim. At last, the fault code was injected through udp mode (python/ matlab form) for fault injection simulation.

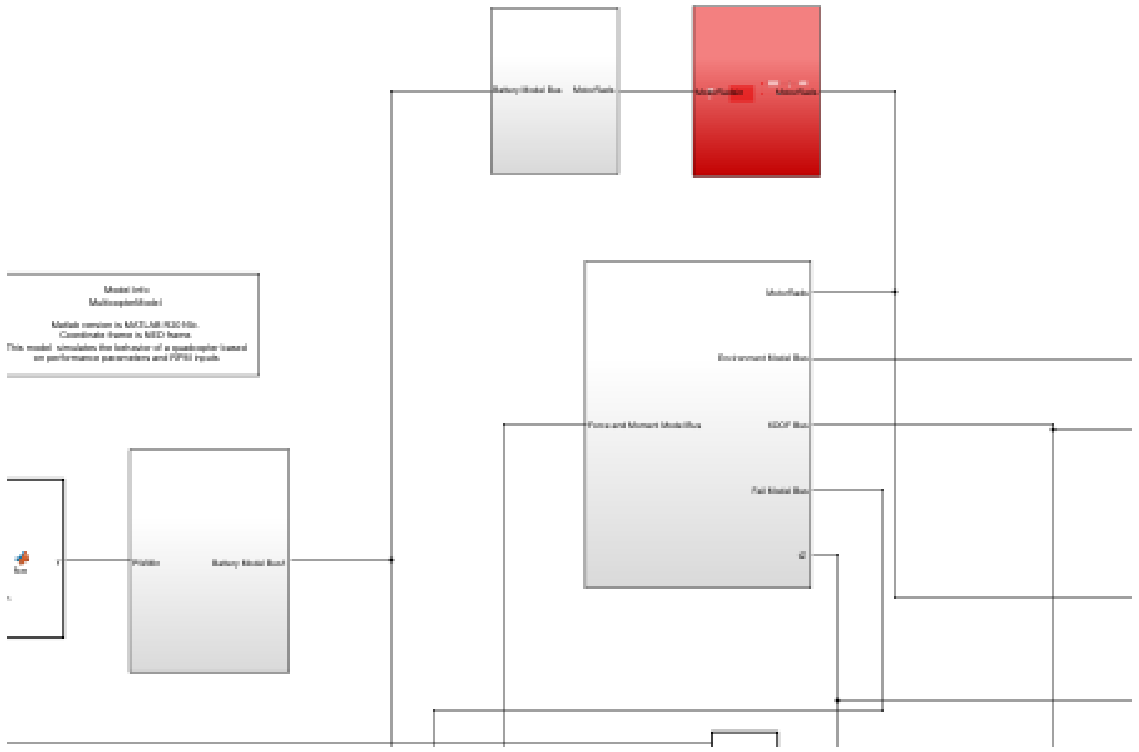
Specific experimental operation see file 1.BasicExps\e6_LoadFault\Readme.pdf



3. Basic experiment case

3.7 Principle of propeller module fault injection based on maximum template

For the propeller module fault mo deling of the maximum template, the fault modeling model is exported as D LL file, and then the DLL file is loade d through CopterSim. Finally, the fau lt code is injected through udp mode (python/ matlab form) for fault injecti on simulation. See file 1.BasicExps\e7 _PropFault\Readme.pdf for specific e xperimental operation

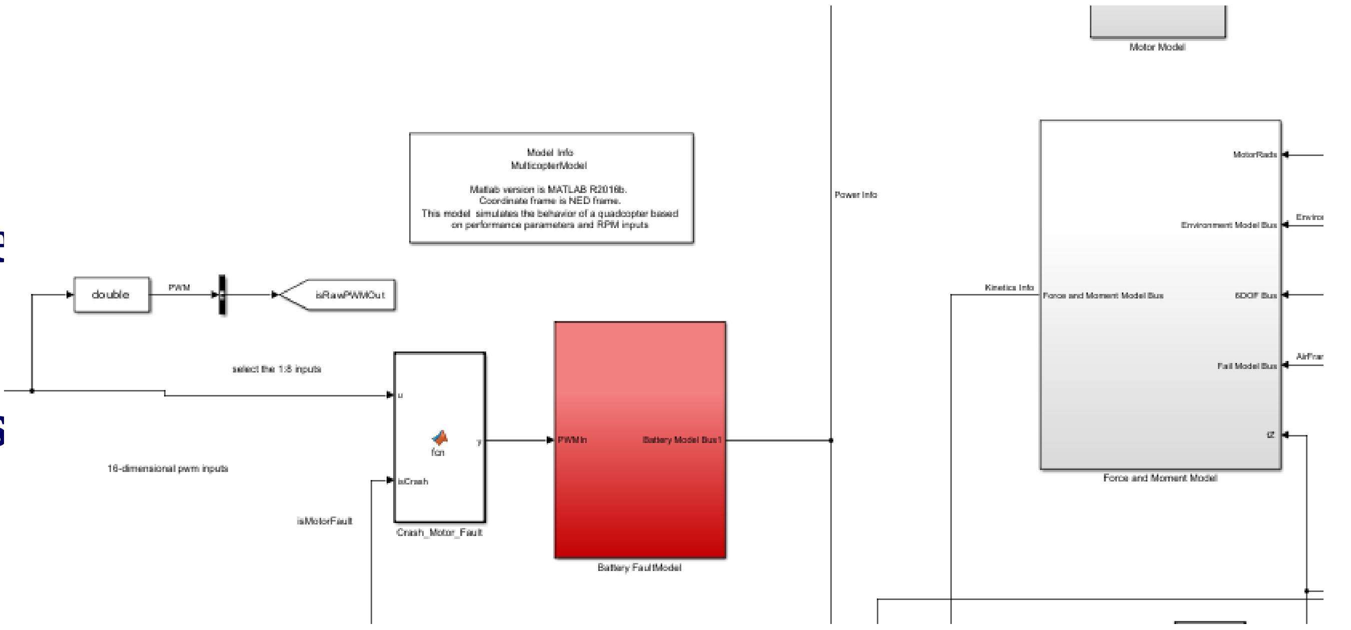

Model Info
MulticopterModel
Matlab version is MATLAB R20 HBx.
Coordinate frame is NED frame.
This model simulates the behavior of a quadcopter based on performance parameters and PWM inputs.
PWMIn
Battery Model Bus
Battery Model Bus
MotorRads
MotorRads
MotorRads
Environment Model Bus
Force and Moment Model Bus
6DOF Bus
Fail Model Bus

3. Basic experiment case

3.8 Principle of battery module fault injection based on maximum template

For the fault modeling of the battery module of the maximum template, the fault modeling model is exported as a DLL file, and then the DLL file is loaded through CopterSim. Finally, the fault code is injected through udp mode (python/ matlab form) for fault injection simulation.

For specific experimental operation, see file [1.BasicExps\e8_BatteryFault\Readme.pdf]


Model Info
MulticopterModel

Matlab version is MATLAB R2016b.
Coordinate frame is NED frame.
This model simulates the behavior of a quadcopter based on performance parameters and RPM inputs

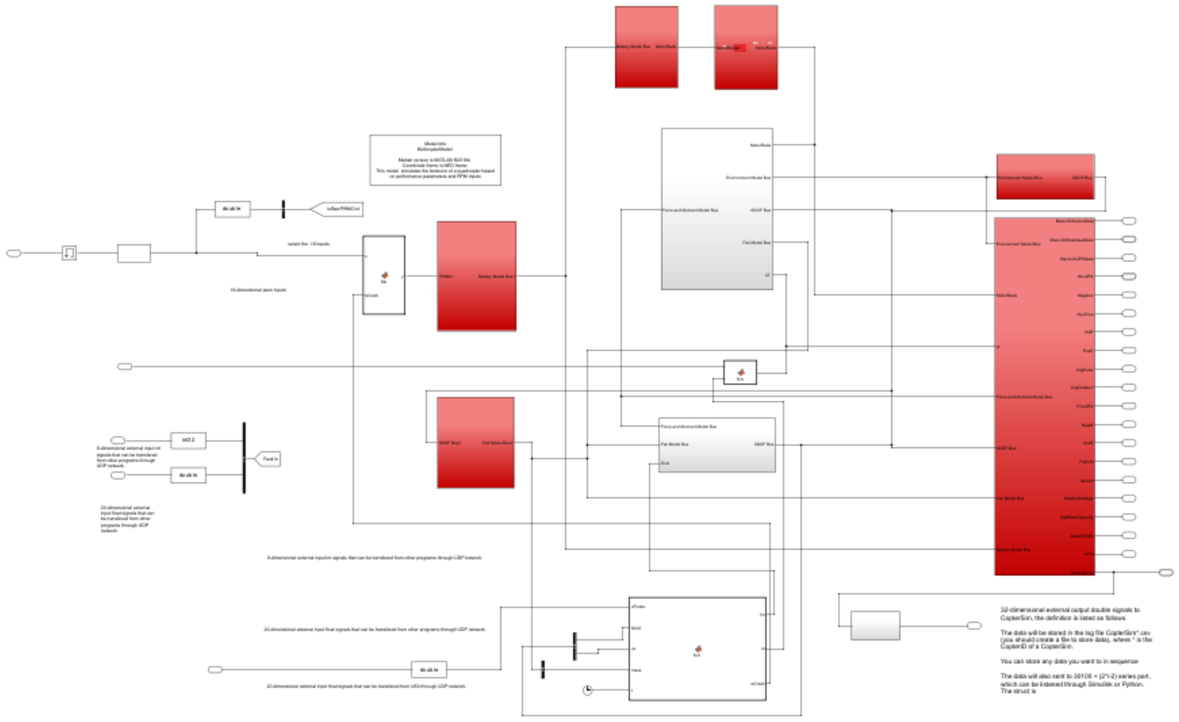
Outline

4. Advanced case experiment

4.1 Principle of total fault module injection based on maximum template

For the fault modeling of the total fault module of the maximum templat e, the fault modeling model is exporte d as a DLL file, and then the DLL file is loaded through CopterSim, and fin ally the fault code is injected through udp mode (python/ matlab form) for f ault injection simulation.

Specific experimental operation se e file 2.AdvExps\e1_FullFaultModelT emplate\readme.pdf

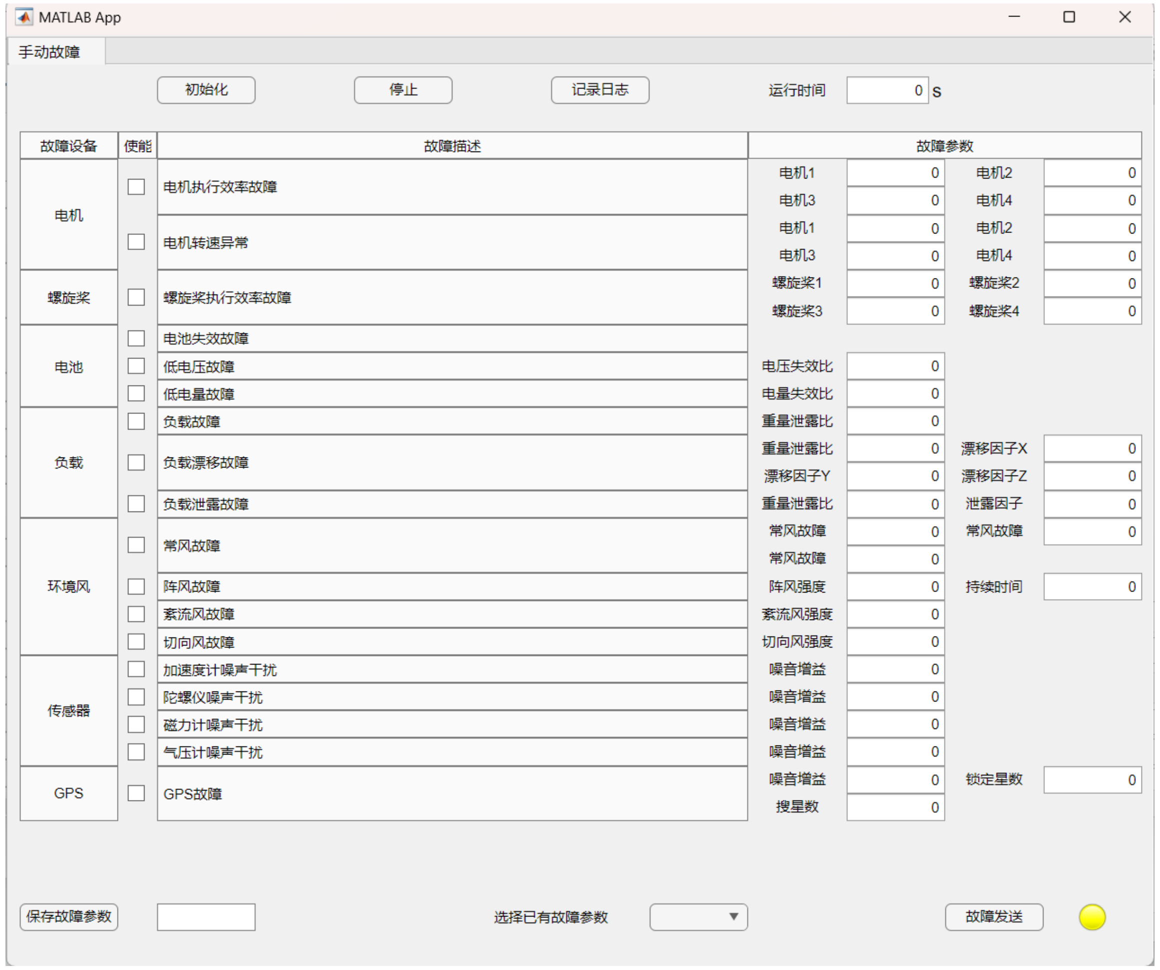


4. Advanced case experiment

4.2 Fault generation injection interface application

Based on the maximum template to carry out a variety of fault injecti on, through the MATLAB APP desi gn can be injected into the model of various fault parameters of the APP, through this method can obviously s ee the injected fault, and the generat ed fault injection into the software i n the loop, to achieve the effect of fa ult injection.

For specific experimental operat ion, see file 2.AdvExps\e2_FailureG enerator GUI APP\readme.pdf


MATLAB App

手动故障

运行时间 0 s

故障设备	使能	故障描述	故障参数			
电机	☐	电机执行效率故障	电机1	0	电机2	0
			电机3	0	电机4	0
	☐	电机转速异常	电机1	0	电机2	0
			电机3	0	电机4	0
螺旋桨	☐	螺旋桨执行效率故障	螺旋桨1	0	螺旋桨2	0
			螺旋桨3	0	螺旋桨4	0
电池	☐	电池失效故障				
	☐	低电压故障	电压失效比	0		
	☐	低电量故障	电量失效比	0		
负载	☐	负载故障	重量泄露比	0		
	☐	负载漂移故障	重量泄露比	0	漂移因子X	0
			漂移因子Y	0	漂移因子Z	0
	☐	负载泄露故障	重量泄露比	0	泄露因子	0
环境风	☐	常风故障	常风故障	0	常风故障	0
			常风故障	0		
	☐	阵风故障	阵风强度	0	持续时间	0
	☐	紊流风故障	紊流风强度	0		
	☐	切向风故障	切向风强度	0		
传感器	☐	加速度计噪声干扰	噪音增益	0		
	☐	陀螺仪噪声干扰	噪音增益	0		
	☐	磁力计噪声干扰	噪音增益	0		
	☐	气压计噪声干扰	噪音增益	0		
GPS	☐	GPS故障	噪音增益	0	锁定星数	0
			搜星数	0		

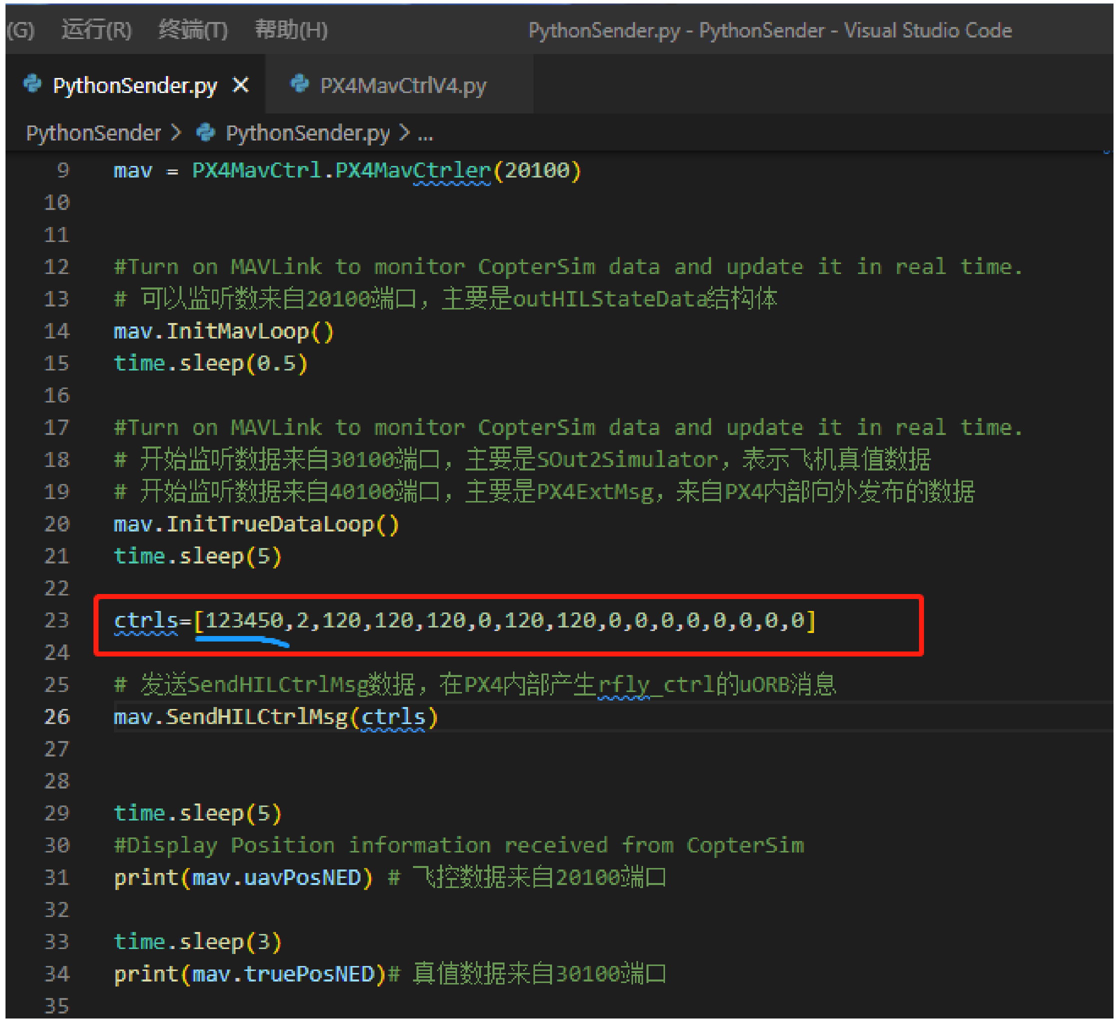
飞思实验室

4. Advanced case experiment

4.3 flight control source code fault injection experiment

This experiment provides a set of scenarios through this experiment do not rely on automatically generated code for fault injection experiment, but directly modify the source code, so as to achieve the effect of fault injection.

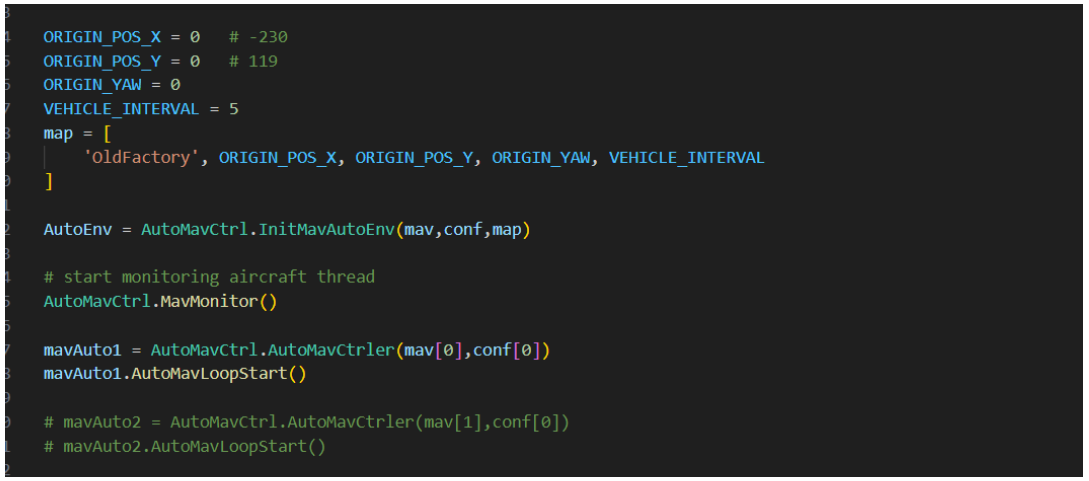
See file 2.AdvExps\e3_PX4 FailureGenerator\readme.pdf for specific experimental operations


(G) 运行(R) 终端(T) 帮助(H)
PythonSender.py - PythonSender - Visual Studio Code
PythonSender.py ×
PX4MavCtrlV4.py
PythonSender > PythonSender.py > ...

```python
mav = PX4MavCtrl.PX4MavCtrler(20100)


#Turn on MAVLink to monitor CopterSim data and update it in real time.
# 可以监听数来自20100端口，主要是outHILStateData结构体
mav.InitMavLoop()
time.sleep(0.5)

#Turn on MAVLink to monitor CopterSim data and update it in real time.
# 开始监听数据来自30100端口，主要是SOut2Simulator，表示飞机真值数据
# 开始监听数据来自40100端口，主要是PX4ExtMsg，来自PX4内部向外发布的数据
mav.InitTrueDataLoop()
time.sleep(5)

ctrls=[123450,2,120,120,120,0,120,120,0,0,0,0,0,0,0,0]

# 发送SendHILCtrlMsg数据，在PX4内部产生rfly_ctrl的uORB消息
mav.SendHILCtrlMsg(ctrls)


time.sleep(5)
#Display Position information received from CopterSim
print(mav.uavPosNED) # 飞控数据来自20100端口

time.sleep(3)
print(mav.truePosNED)# 真值数据来自30100端口
```

4. Advanced case experiment

- **4.4 Automatic test platform Use single machine single instance automatic test**
- **Master the basic structure and use process of automatic test platform.**
- **See file 2.AdvExps\e4_SingFramSingleInsExp\readme.pdf for specific experimental operation**

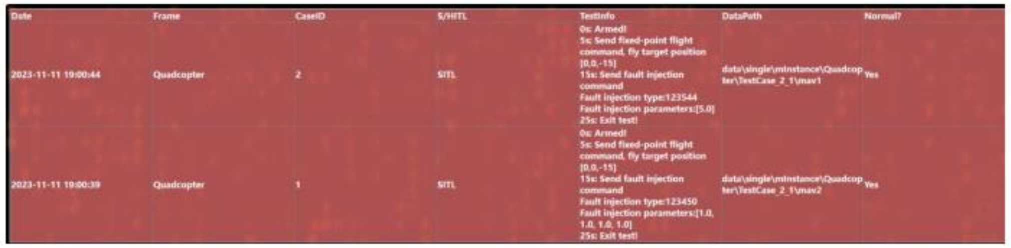


```
ORIGIN_POS_X = 0   # -230
ORIGIN_POS_Y = 0   # 119
ORIGIN_YAW = 0
VEHICLE_INTERVAL = 5
map = [
    'OldFactory', ORIGIN_POS_X, ORIGIN_POS_Y, ORIGIN_YAW, VEHICLE_INTERVAL
]

AutoEnv = AutoMavCtrl.InitMavAutoEnv(mav,conf,map)

# start monitoring aircraft thread
AutoMavCtrl.MavMonitor()

mavAuto1 = AutoMavCtrl.AutoMavCtrler(mav[0],conf[0])
mavAuto1.AutoMavLoopStart()

# mavAuto2 = AutoMavCtrl.AutoMavCtrler(mav[1],conf[0])
# mavAuto2.AutoMavLoopStart()
```

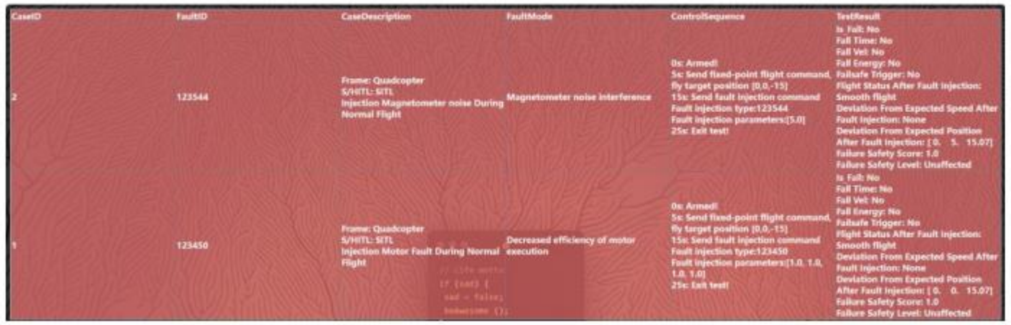
飞思实验室

4. Advanced case experiment

- **4.5 Automatic test platform uses single-machine multi-instance automated testing**
- **Master the basic structure and use flow of single-machine multi-instance automated testing on automatic test platform.**
- **See file 2.AdvExps\e5_SingFrameMultiInsExp\readme.pdf for specific experiment operation**



Date	Frame	CaseID	S/HITL	TestInfo	DataPath	Normal?
2023-11-11 19:00:44	Quadcopter	2	SITL	0s: Armed! 5s: Send fixed-point flight command, fly target position [0,0,-15] 15s: Send fault injection command Fault injection type:123544 Fault injection parameters:[5.0] 25s: Exit test!	data\single\minstance\Quadcopter\TestCase_2_1\mav1	Yes
2023-11-11 19:00:39	Quadcopter	1	SITL	0s: Armed! 5s: Send fixed-point flight command, fly target position [0,0,-15] 15s: Send fault injection command Fault injection type:123450 Fault injection parameters:[1.0, 1.0, 1.0, 1.0] 25s: Exit test!	data\single\minstance\Quadcopter\TestCase_2_1\mav2	Yes

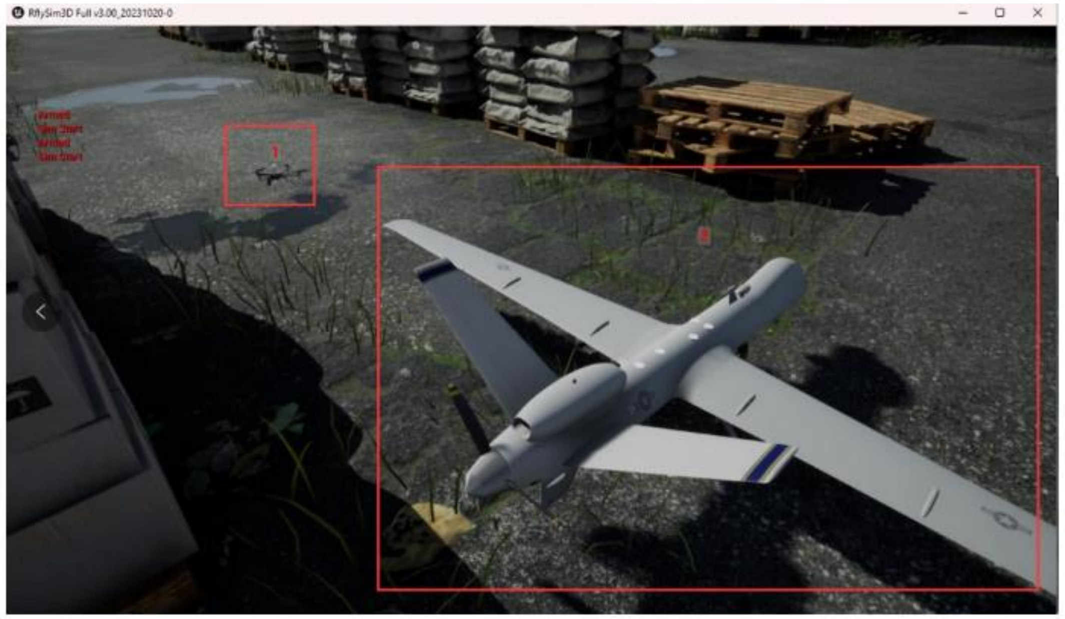


CaseID	FaultID	CaseDescription	FaultMode	ControlSequence	TestResult
2	123544	Frame: Quadcopter S/HITL: SITL Injection Magnetometer noise During Normal Flight	Magnetometer noise interference	0s: Armed! 5s: Send fixed-point flight command, fly target position [0,0,-15] 15s: Send fault injection command Fault injection type:123544 Fault injection parameters:[5.0] 25s: Exit test!	Is_Fall: No Fall Time: No Fall Vel: No Fall Energy: No Failsafe Trigger: No Flight Status After Fault Injection: Smooth flight Deviation From Expected Speed After Fault Injection: None Deviation From Expected Position After Fault Injection: [0. 5. 15.07] Failure Safety Score: 1.0 Failure Safety Level: Unaffected
1	123450	Frame: Quadcopter S/HITL: SITL Injection Motor Fault During Normal Flight	Decreased efficiency of motor execution	0s: Armed! 5s: Send fixed-point flight command, fly target position [0,0,-15] 15s: Send fault injection command Fault injection type:123450 Fault injection parameters:[1.0, 1.0, 1.0, 1.0] 25s: Exit test!	Is_Fall: No Fall Time: No Fall Vel: No Fall Energy: No Failsafe Trigger: No Flight Status After Fault Injection: Smooth flight Deviation From Expected Speed After Fault Injection: None Deviation From Expected Position After Fault Injection: [0. 0. 15.07] Failure Safety Score: 1.0 Failure Safety Level: Unaffected

飞思实验室

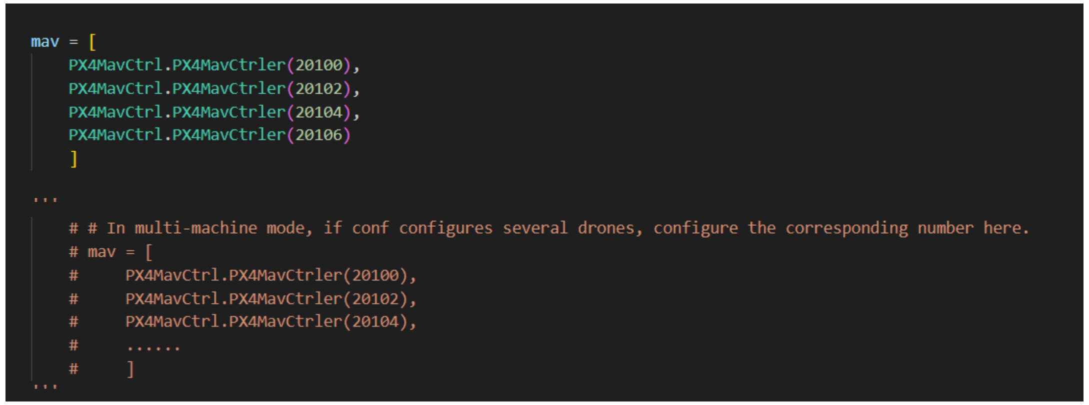
4. Advanced case experiment

- **4.6 Automatic test platform Use multi-model single-instance automated test**
- **Master the basic structure and application process of multi-model single-instance automated testing on automatic test platform.**
- **See file 2.AdvExps\e6_MultiFrameSingleInsExp\readme.pdf for specific experiment operation**



4. Advanced case experiment

- **4.7 Automatic test platform uses multi-model and multi-instance automated testing**
- **Master the basic structure and application process of multi-model and multi-instance automated testing of automatic test platform.**
- **See file 2.AdvExps\e7_MultiFrameMultiInsExp\readme.pdf for specific experiment operation**

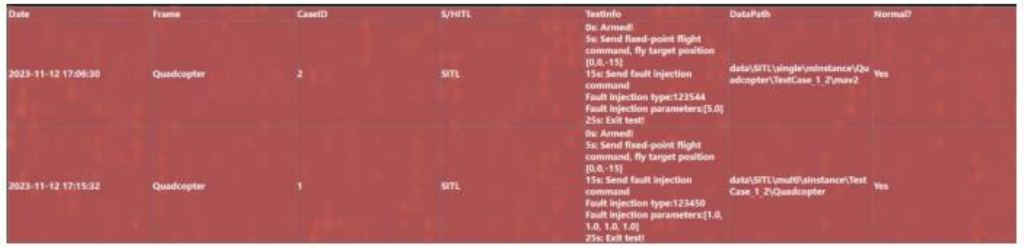


```
mav = [
    PX4MavCtrl.PX4MavCtrler(20100),
    PX4MavCtrl.PX4MavCtrler(20102),
    PX4MavCtrl.PX4MavCtrler(20104),
    PX4MavCtrl.PX4MavCtrler(20106)
    ]

'''
    # # In multi-machine mode, if conf configures several drones, configure the corresponding number here.
    # mav = [
    #     PX4MavCtrl.PX4MavCtrler(20100),
    #     PX4MavCtrl.PX4MavCtrler(20102),
    #     PX4MavCtrl.PX4MavCtrler(20104),
    #     ......
    #     ]
'''
```

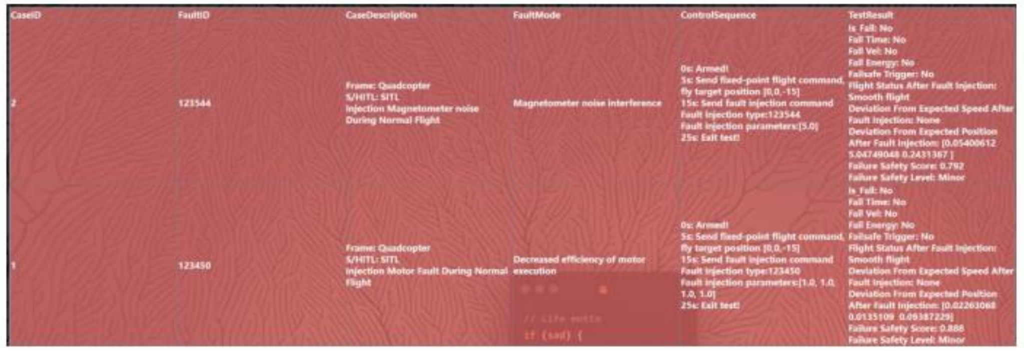
飞思实验室

4. Advanced case experiment

- **4.8 Security assessment**
- **Master the basic structure and application process of the security assessment algorithm.**
- **See file 2.AdvExps\e8_SafetyAssExp\readme.pdf for specific experimental operation**



Date	Frame	CaseID	S/HITL	TestInfo	DataPath	Normal?
2023-11-12 17:06:30	Quadcopter	2	SITL	0s: Armed! 5s: Send fixed-point flight command, fly target position [0,0,-15] 15s: Send fault injection command Fault injection type:123544 Fault injection parameters:[5.0] 25s: Exit test!	data\SITL\single\minstance\Quadcopter\TestCase_1_2\mav2	Yes
2023-11-12 17:15:32	Quadcopter	1	SITL	0s: Armed! 5s: Send fixed-point flight command, fly target position [0,0,-15] 15s: Send fault injection command Fault injection type:123450 Fault injection parameters:[1.0, 1.0, 1.0, 1.0] 25s: Exit test!	data\SITL\multi\sinstance\TestCase_1_2\Quadcopter	Yes

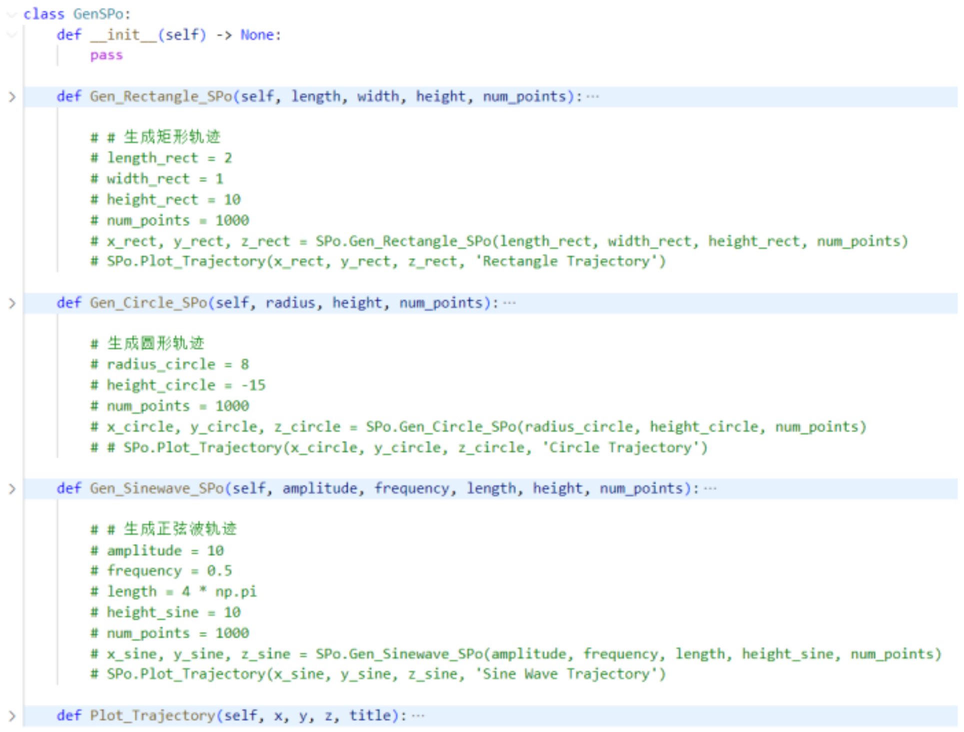


CaseID	FaultID	CaseDescription	FaultMode	ControlSequence	TestResult
2	123544	Frame: Quadcopter S/HITL: SITL Injection Magnetometer noise During Normal Flight	Magnetometer noise interference	0s: Armed! 5s: Send fixed-point flight command, fly target position [0,0,-15] 15s: Send fault injection command Fault injection type:123544 Fault injection parameters:[5.0] 25s: Exit test!	Is Fall: No Fall Time: No Fall Vel: No Fall Energy: No Failsafe Trigger: No Flight Status After Fault Injection: Smooth flight Deviation From Expected Speed After Fault Injection: None Deviation From Expected Position After Fault Injection: [0.05400612 5.04749048 0.2431367] Failure Safety Score: 0.792 Failure Safety Level: Minor
1	123450	Frame: Quadcopter S/HITL: SITL Injection Motor Fault During Normal Flight	Decreased efficiency of motor execution	0s: Armed! 5s: Send fixed-point flight command, fly target position [0,0,-15] 15s: Send fault injection command Fault injection type:123450 Fault injection parameters:[1.0, 1.0, 1.0, 1.0] 25s: Exit test!	Is_Fall: No Fall Time: No Fall Vel: No Fall Energy: No Failsafe Trigger: No Flight Status After Fault Injection: Smooth flight Deviation From Expected Speed After Fault Injection: None Deviation From Expected Position After Fault Injection: [0.02263068 0.0135109 0.09387229] Failure Safety Score: 0.888 Failure Safety Level: Minor

飞思实验室

4. Advanced case experiment

- **4.9 Automated test track tracking**
- **Master the basic structure and use process of track tracking.**
- **See file 2.AdvExps\e9_SetpointCtrlExp\readme.pdf for specific experiment operation**



```
class GenSPo:
    def __init__(self) -> None:
        pass

    def Gen_Rectangle_SPo(self, length, width, height, num_points):…

        # # 生成矩形轨迹
        # length_rect = 2
        # width_rect = 1
        # height_rect = 10
        # num_points = 1000
        # x_rect, y_rect, z_rect = SPo.Gen_Rectangle_SPo(length_rect, width_rect, height_rect, num_points)
        # SPo.Plot_Trajectory(x_rect, y_rect, z_rect, 'Rectangle Trajectory')

    def Gen_Circle_SPo(self, radius, height, num_points):…

        # 生成圆形轨迹
        # radius_circle = 8
        # height_circle = -15
        # num_points = 1000
        # x_circle, y_circle, z_circle = SPo.Gen_Circle_SPo(radius_circle, height_circle, num_points)
        # # SPo.Plot_Trajectory(x_circle, y_circle, z_circle, 'Circle Trajectory')

    def Gen_Sinewave_SPo(self, amplitude, frequency, length, height, num_points):…

        # # 生成正弦波轨迹
        # amplitude = 10
        # frequency = 0.5
        # length = 4 * np.pi
        # height_sine = 10
        # num_points = 1000
        # x_sine, y_sine, z_sine = SPo.Gen_Sinewave_SPo(amplitude, frequency, length, height_sine, num_points)
        # SPo.Plot_Trajectory(x_sine, y_sine, z_sine, 'Sine Wave Trajectory')

    def Plot_Trajectory(self, x, y, z, title):…
```

飞思实验室

Outline

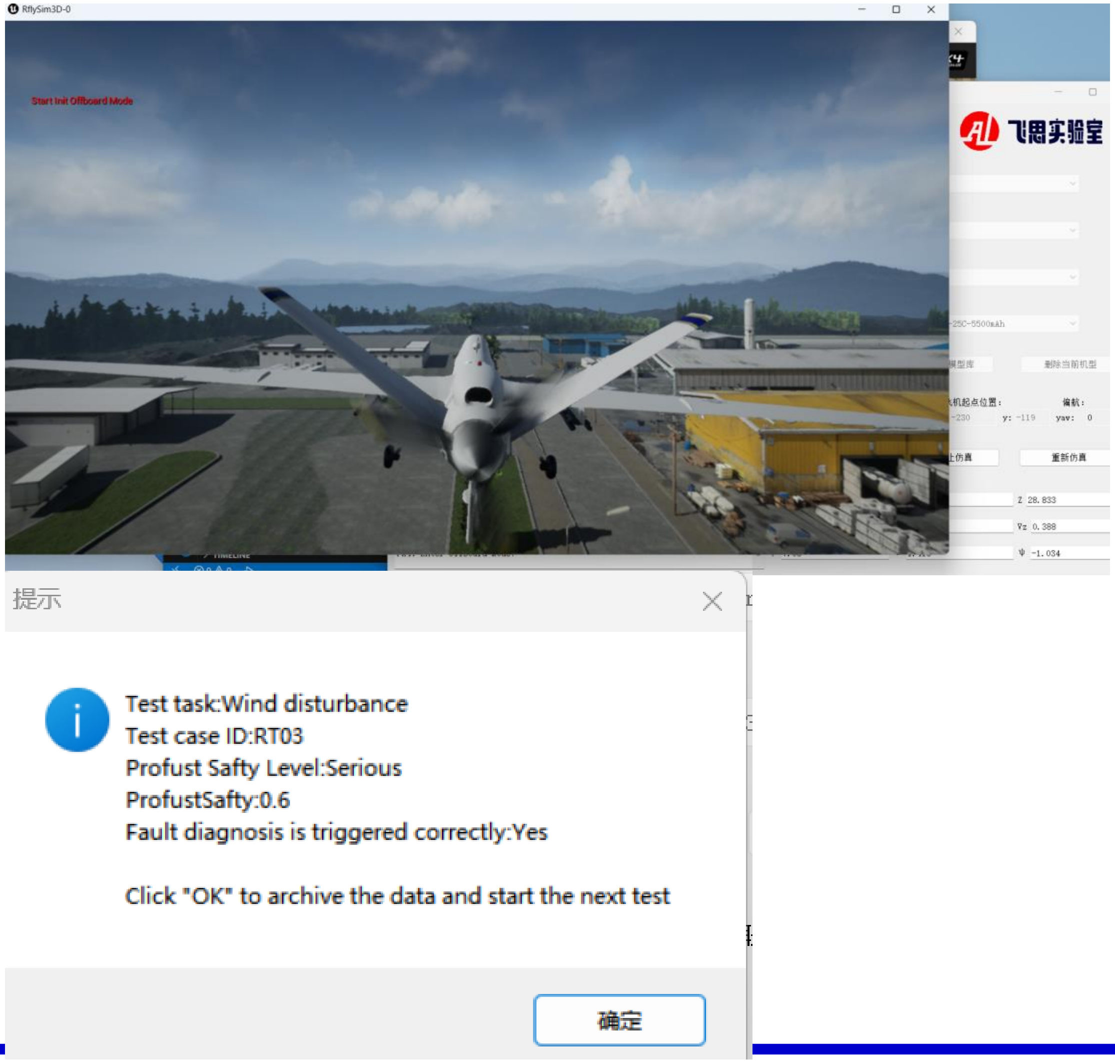
飞思实验室

5 Extended Case

5.1 Motor failure safety evaluation experiment

Carry out pwm output fault modeling for the motor, export the fault modeling model as DLL file, and then load the DLL file through CopterSim, and finally inject fault code through udp mode (python/ matlab form) for security testing, and record the test results.

See 3.CustExps\e1_HealthProjPlatform\readme.pdf for details of the experimental process



Test task:Wind disturbance
Test case ID:RT03
Profust Safty Level:Serious
ProfustSafty:0.6
Fault diagnosis is triggered correctly:Yes

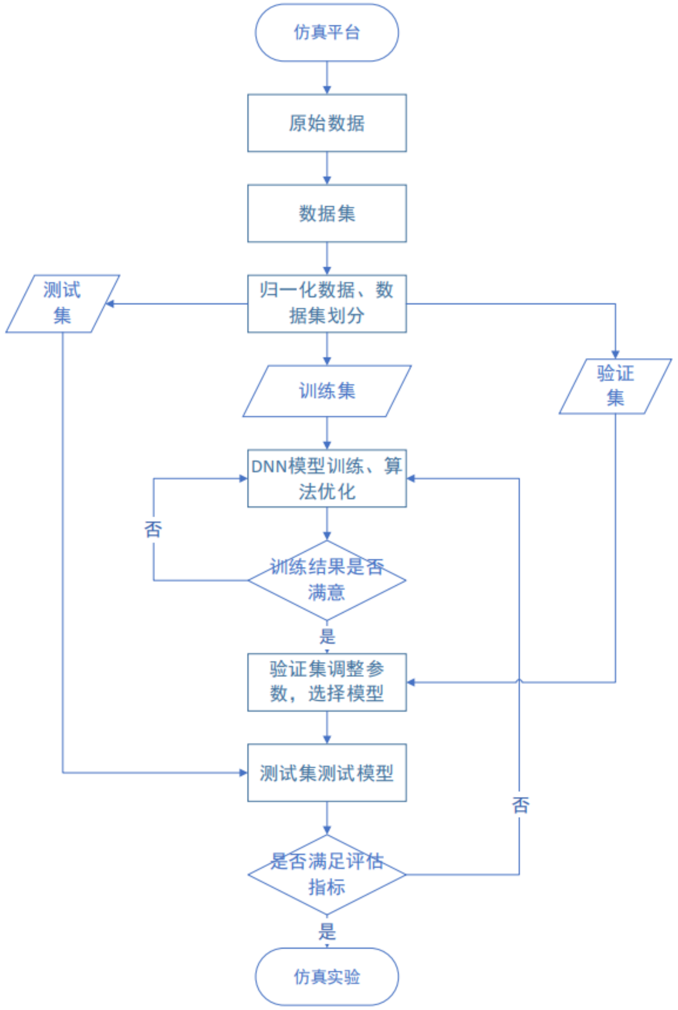
Click "OK" to archive the data and start the next test

5. Extended Case

- **5.2 Drone fault diagnosis based on digital twin and deep learning**
- **This experiment aims to explore the application of digital twin technology and deep learning method in UAV fault diagnosis**
- **See 3.CustExps\e2_DigitalTwinExp\readme.pdf for details of the experiment process**

5. Extension case

- **5.3 Health Platform**
- **Health assessment is conducted through data collection.**
- **See 3.CustExps\e3_health_ass_0\readme.pdf for details of the experimental procedure**



Outline

6 Summary

- This lecture mainly explains the fault injection model of UAV system, which is divided into three parts: basic experiment, advanced experiment and extended case, which can realize the model fault injection and flight control source code injection tutorial.

If in doubt, please go to https://doc.rflysim.com/ for more information.

RflySim for more tutorials

Scan code for consultation and communication

Feisi RflySim technical exchange group

飞思实验室

Thank you!