



---

# 智能无人集群系统开发与实践

## 基于RflySim工具链的全栈开发案例

### 第7讲 安全测试与健康评估



# 大纲

---

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结



# 1. 安装方法

---

## 1.1 需要安装的组件

- Visual Studio 2017（体验版和完整版都需要安装）
- 为MATLAB配置C++ 编译器（体验版和完整版都需要安装）
- Matlab 2023a\*（高级完整版安装）

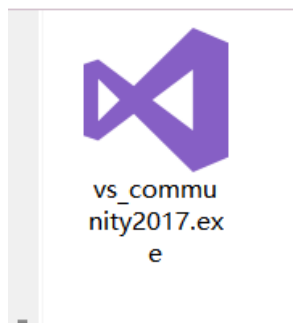
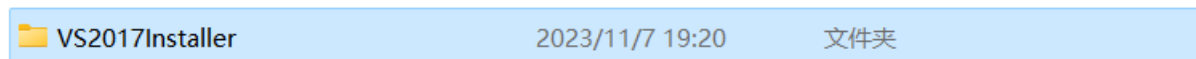
下面介绍Visual Studio 2017的安装方法（需要联网）：  
在本平台中，已经放置了Visual Studio 2017的安装包



# 1. 安装方法

## 1.2 Visual Studio 2017的安装方法

- 首先，我们可以打开平台安装的位置，找到\*:\PX4PSP\RflySimAPIs此处位置，此处放置的是平台中的一些例程以及软件的安装包
- 之后，我们可以打开第四章的内容，找到基础版的例程，4.RflySimModel\1.BasicExps，我们可以在其中找到名为VS2017Installer的文件夹，其中便是Visual Studio 2017的安装包。



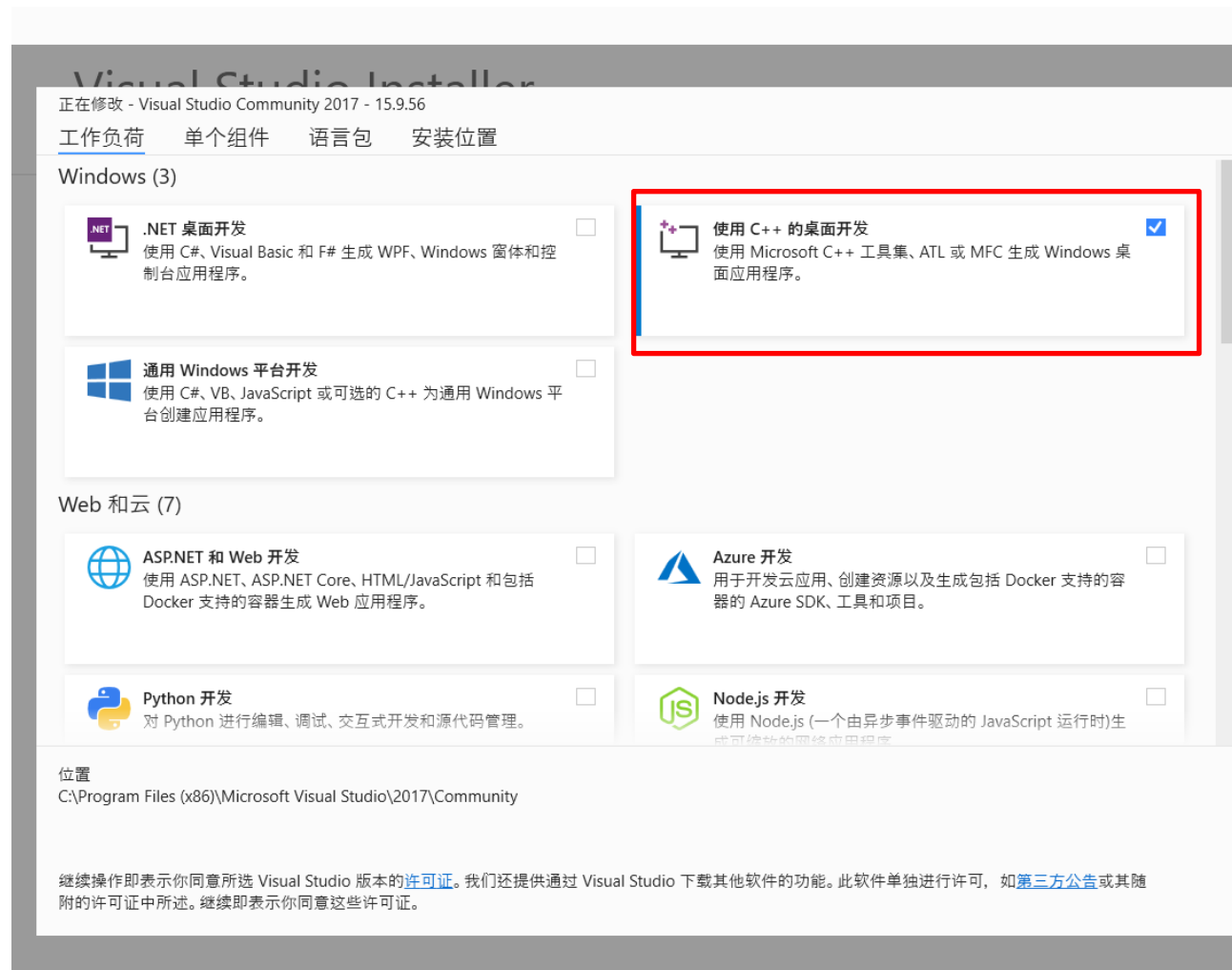
在线安装步骤（需联网）如下：双击  
[PX4PSP\RflySimAPIs\4.RflySimModel\1.BasicExps\VS2017Installer\vs\\_community2017.exe](#)



# 1. 安装方法

## 1.2 Visual Studio 2017的安装方法

- 安装 Visual Studio 2017 (也可以用其他版本, MATLAB能识别即可)。
- 后续课程很多地方都需要用到 Visual Studio 编译器, 例如 MATLAB S-Function Builder 模块的使用、Simulink 自动生成 C/C++ 模型代码等
- 本课程内容只需勾选右图的“C++的桌面开发”即可。





## 1. 安装方法

---

### 1.2 Visual Studio 2017的安装方法

- 注意：高版本MATLAB也可安装VS2019，但是MATLAB只能识别到低于自己版本的Visual Studio，因此MATLAB 2017b无法识别VS 2019。
- 注意：请不要更改VS默认安装目录（例如装到D盘），会导致MATLAB无法识别。
- 不能使用Mingw编译器，需VS



# 1. 安装方法

## • 1.3 为MATLAB 配置C++ 编译器

• 在MATLAB的命令行窗口中输入指令“mex -setup”

• 一般来说会自动识别并安装上VS 2017编译器，如右图所示显示“MEX 配置使用‘Microsoft Visual C++ 2017’以进行编译”说明安装正确

• 若有其他编译器，本页面还可以切换选择 VS 2013/2015等其他编译器

```
命令窗口
>> 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\R2017b\bin\win64\mexopts\msvc2017.xml C

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 >>
```



## 1. 安装方法

- 1.4 Matlab 2023a的安装方法
- MATLAB 安装包下载路径:
- <https://ww2.mathworks.cn/products/matlab.html>







# 大纲

---

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结



## 2.关键接口介绍

### 2.0 基础实验总览

包括基础功能接口“RflySimAPIs/7.RflySimPHM/0.ApiExps”以及基础例程“RflySimAPIs\7.RflySimPHM\1.BasicExps”

详细参见[API.pdf](#)以及[Readme.pdf](#)

名称	修改日期	类型
文件夹 e1_SignTAG	2023/11/7 19:20	文件夹
文件夹 e2_FaultParamStruct	2023/11/7 19:20	文件夹
文件夹 e3_FaultInjectAPITest_mat	2023/11/7 19:20	文件夹
文件夹 e4_FaultInjectAPITest_py	2023/11/7 19:20	文件夹
文件夹 e5_ExtMsgSender	2023/11/7 19:20	文件夹
文件夹 e6_UseFaultLib	2023/11/7 19:20	文件夹
文件夹 e7_NoFaultModelMinTemplate	2023/11/7 19:20	文件夹
文件夹 e8_BaseMotorFault	2023/11/7 19:20	文件夹
文件夹 e1_NoFaultModelMaxTemplate	2023/11/7 19:20	文件夹
文件夹 e2_GPSFault	2023/11/7 19:20	文件夹
文件夹 e3_MotorFault	2023/11/7 19:20	文件夹
文件夹 e4_SensorFault	2023/11/7 19:20	文件夹
文件夹 e5_WindFault	2023/11/7 19:20	文件夹
文件夹 e6_LoadFault	2023/11/7 19:20	文件夹
文件夹 e7_PropFault	2023/11/7 19:20	文件夹
文件夹 e8_BatteryFault	2023/11/7 19:20	文件夹

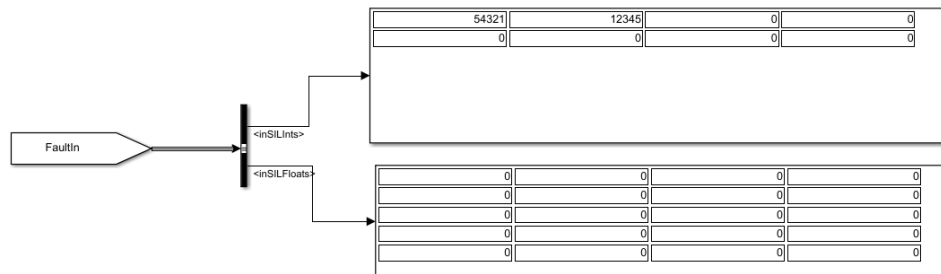
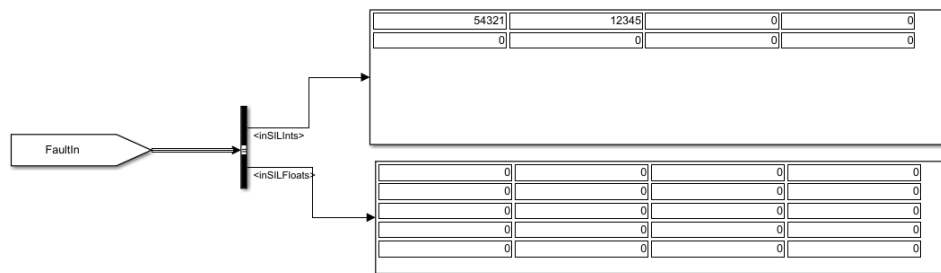
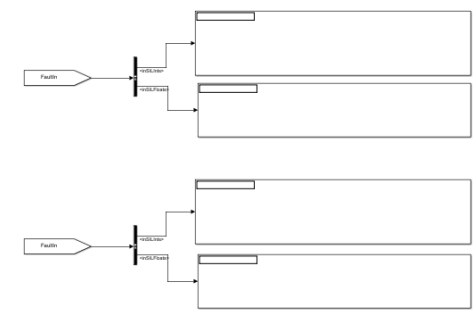
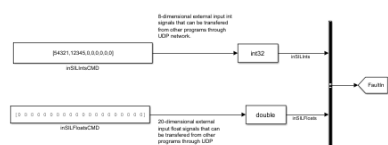
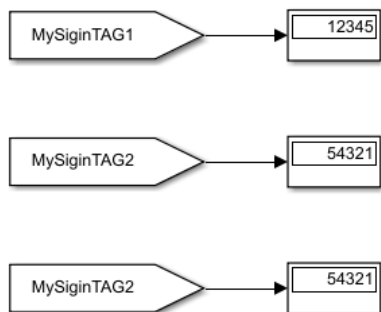


## 2.关键接口介绍

### 2.1信号标签模块的学习与使用

通过本次例程学习对 Goto 以及 From 模块的使用。详细操作及实验效果见

[0.ApiExps/e1\\_SignTAG/readme.pdf](0.ApiExps/e1_SignTAG/readme.pdf)





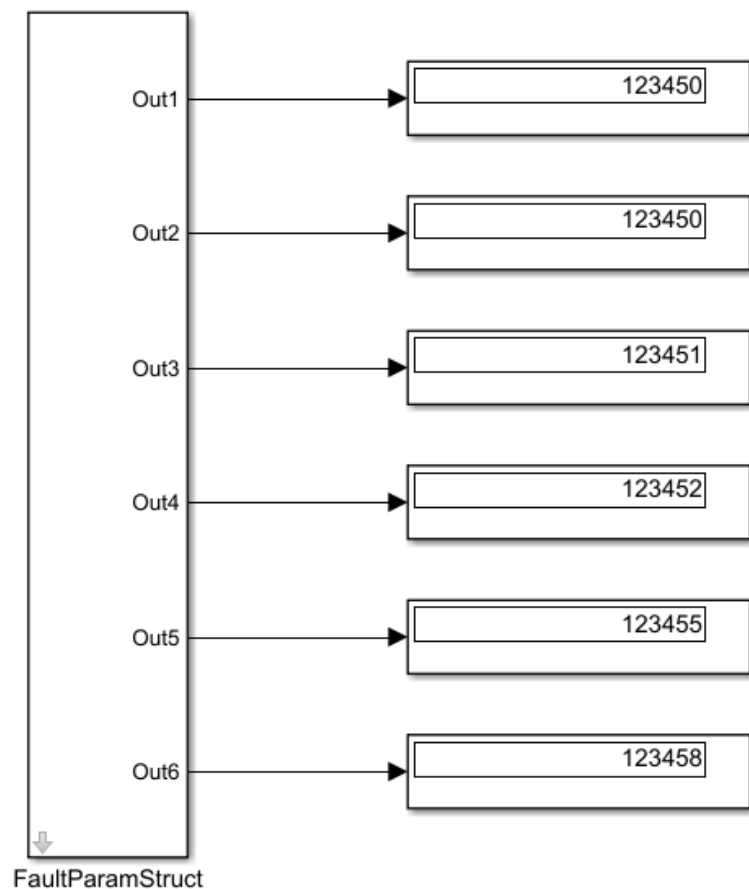
## 2.关键接口介绍

### 2.2故障参数与模块封装参数引用的学习与使用

学习通过创建封装参数从工作区读取所需故障参数。

详细操作及实验效果见

[0.ApiExps\e2\\_FaultParamStruct\readme.pdf](#)





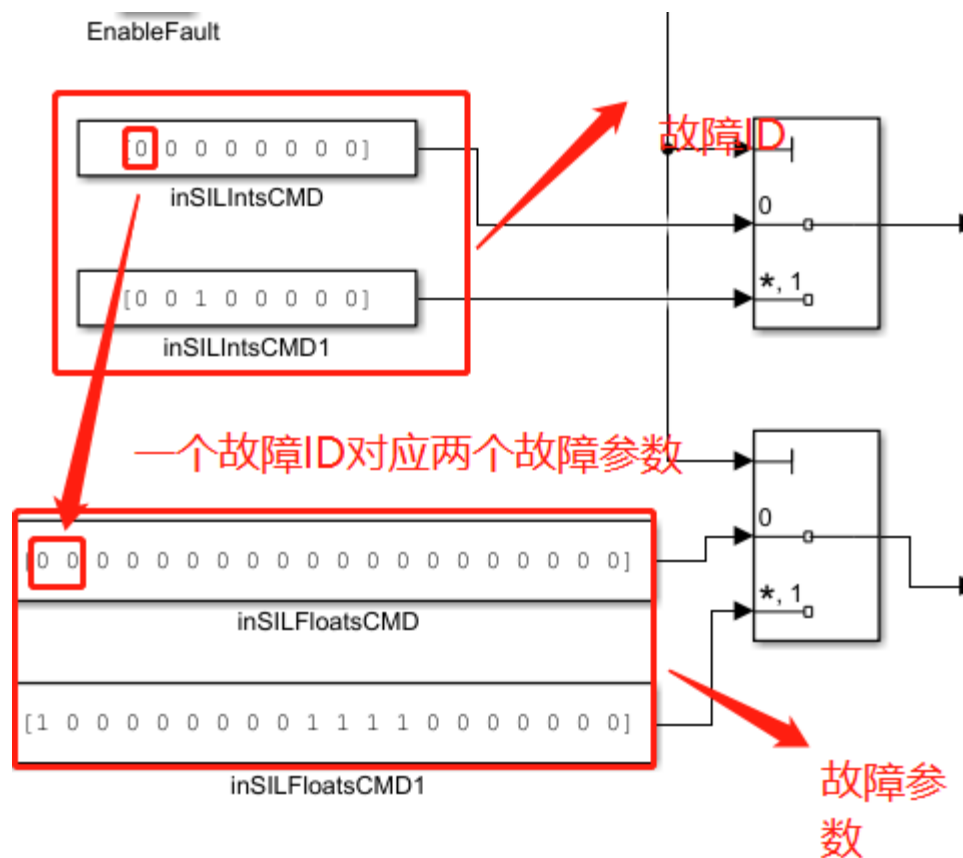
## 2.关键接口介绍

### 2.3 UDP模式发送故障注入参数模块的学习与使用

通过本次例程学习使用UDP模式发送故障注入参数代码。

具体实验操作及效果见

[0.ApiExps/e3\\_FaultInjectAPITest\\_mat/readme.pdf](0.ApiExps/e3_FaultInjectAPITest_mat/readme.pdf)



Send inSILs and inFloats signals to CooterSim DLL model through



## 2.关键接口介绍

### 2.4 UDP模式发送故障注入参数模块的学习与使用

通过本次例程学习使用UDP模式发送故障注入参数代码。部分实验效果如右图，详细操作及实验效果见[0.ApiExps\e4\\_FaultInjectAPITest\\_py\readme.pdf](#)

```
silFloat=np.zeros(20).astype(float).tolist()
输入两个故障
silInt[0:2]=[123450,123450]
silFloat[0:4]=[0,0,0,0]
输入单个故障
# silInt[0:1]=[123540]
# silFloat[0:2]=[15,20]
```



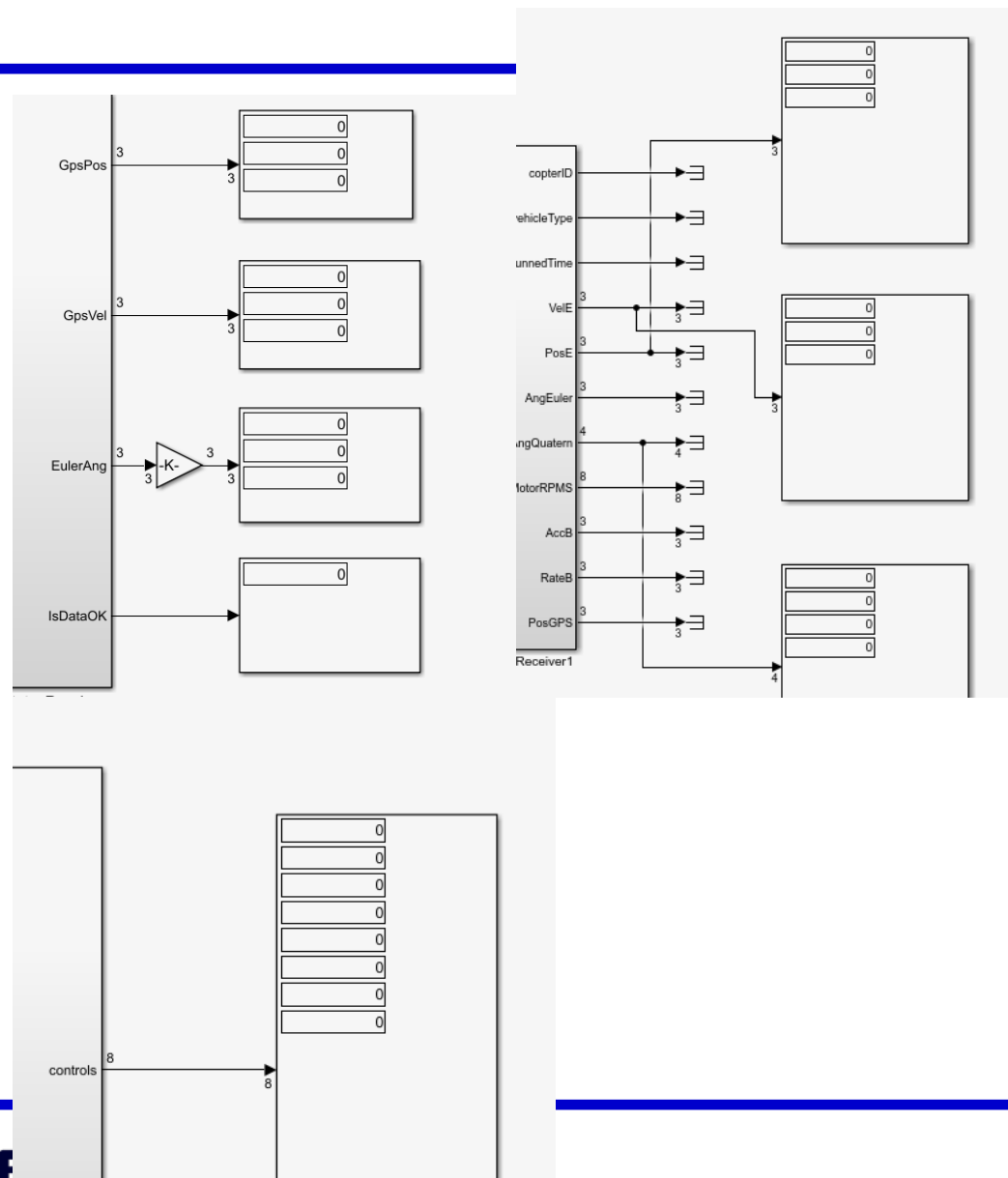
## 2.关键接口介绍

### 2.5 PX4外部消息发送与接收模块的学习与使用

通过本次例程学习如何向PX4的接口发送外部消息和接受PX4的状态信息。

具体实验操作及效果见

[0.ApiExps\e5\\_ExtMsgSender\readme.pdf](#)





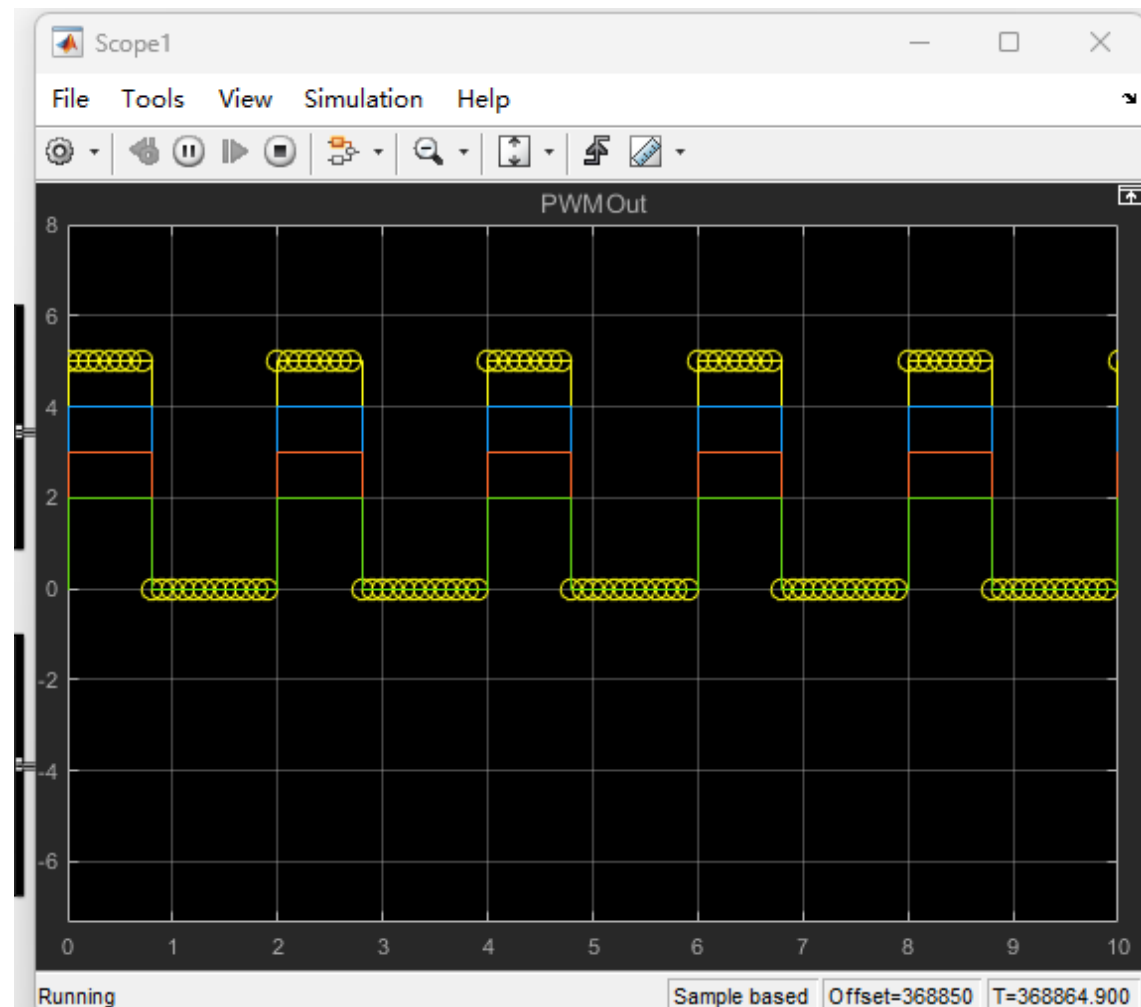
## 2.关键接口介绍

### 2.6电机故障建模原理的学习与使用

通过本次例程学习从0到1使用一个带有电机故障注入模块的学习与使用。

具体实验操作见文件

[0.ApiExps\e6 UseFaultLib\readme.pdf](#)







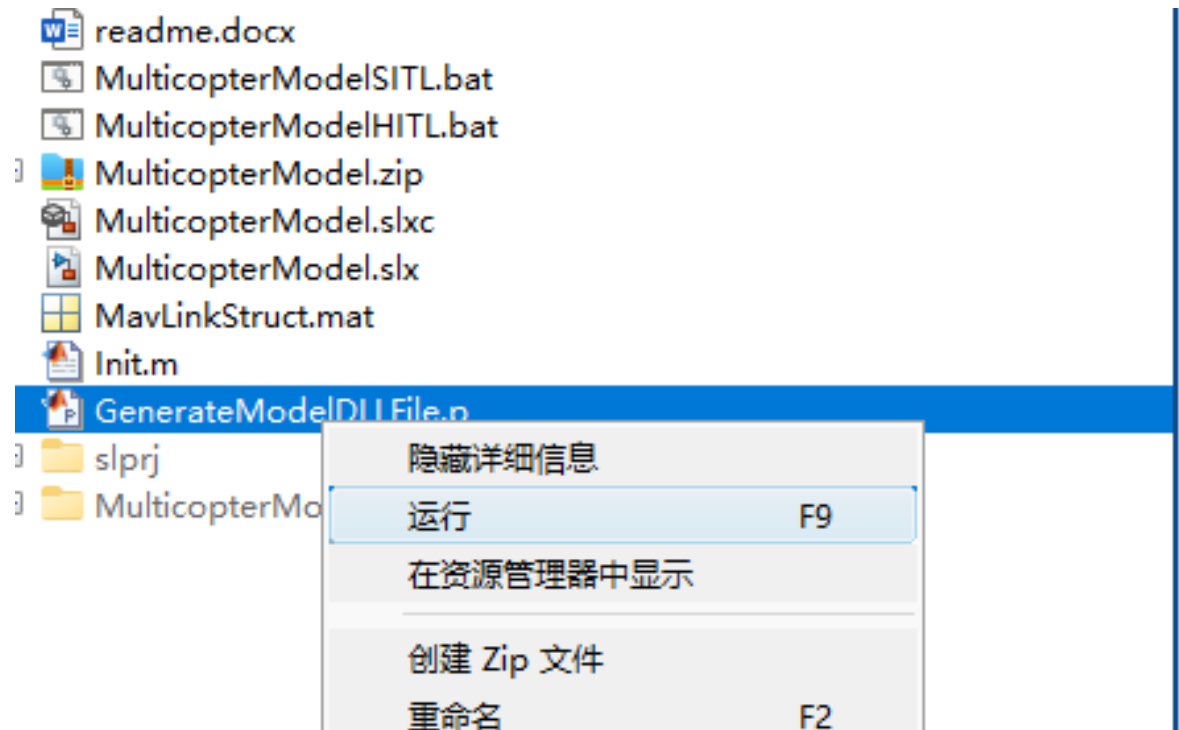
## 2.关键接口介绍

### 2.7故障注入模块的最小模板

通过本例程学习故障注入最小模板的使用，学习如何生成DLL文件。

具体实验操作见文件

[0.ApiExps\e7\\_NoFaultModelMinTemplate\readme.pdf](#)





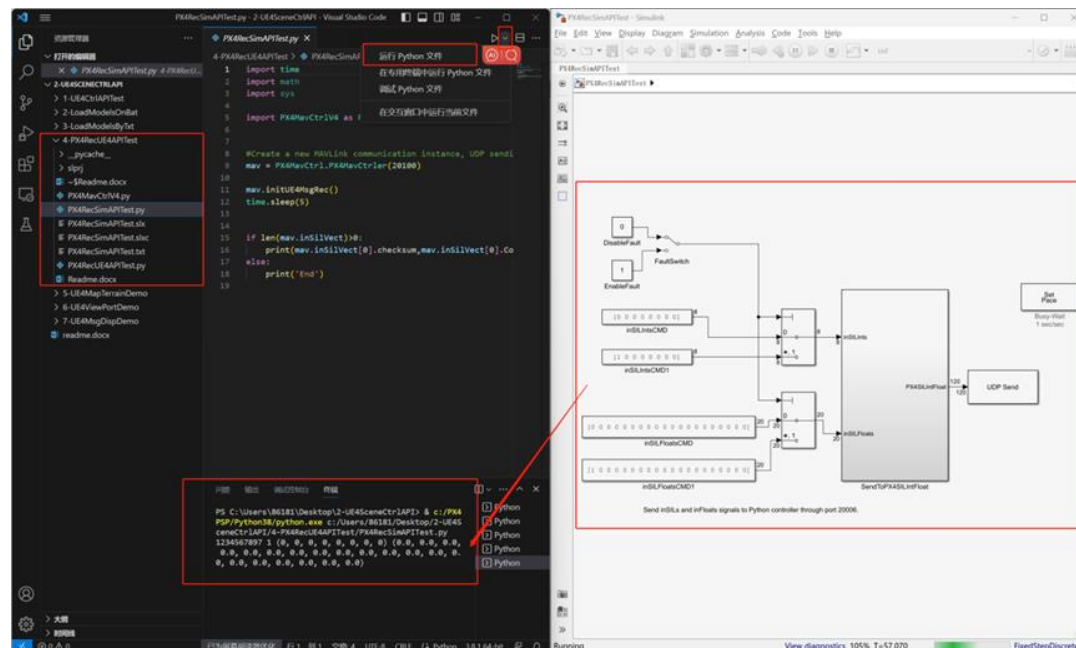
## 2.关键接口介绍

### 2.8基于最小模板的电机故障注入的学习与使用

通过本例程学习使用基于最小模板的电机故障注入。

具体实验操作见文件

[0.ApiExps/e8\\_BaseMotorFault/readme.pdf](0.ApiExps/e8_BaseMotorFault/readme.pdf)





## 2.关键接口介绍

- 2.9数据收集
- 收集数据方便对训练模型进行训练及验证。
- 具体实验操作见文件

[0.ApiExps\e9\\_data\\_collect/readme.pdf](#)

```
1 {
2   "faultcase": [
3     {
4       "CaseID": 1,
5       "Subsystem": "Sensor subsystem",
6       "FaultType": "123545",
7       "ControlSequence": "2,1;1,1,5;2,3,0,0,-10;1,1,10;2,6,123544,5,0;1,1,10",
8       "TestStatus": "UnFinished"
9     },
10    {
11      "CaseID": 2,
12      "Subsystem": "Power subsystem",
13      "FaultType": "123450",
14      "ControlSequence": "2,1;1,1,5;2,3,0,0,-10;1,1,10;2,6,123450,0,1;1,1,10",
15      "TestStatus": "UnFinished"
16    },
17    {
18      "CaseID": 3,
19      "Subsystem": "Environment subsystem",
20      "FaultType": "123540",
21      "ControlSequence": "2,1;1,1,5;2,3,0,0,-10;1,1,10;2,6,123540,15,1;1,1,10",
22      "TestStatus": "UnFinished"
23    }
24  ],
25  "testcase": "1,3"
26 }
```



## 2.关键接口介绍

- 2.10数据处理
- 对收集到的数据进行处理，生成 Excel 表格。
- 具体实验操作见文件

[0.ApiExps\e10\\_data\\_process/readme.pdf](#)

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



## 2.关键接口介绍

---

- 2.11模型验证
- 对训练模型进行验证。
- 具体实验操作见文件

[0.ApiExps\e11\\_moder\\_ver\readme.pdf](#)

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.关键接口介绍

- 2.12故障判断
- 通过软硬件在环，根据飞机位置的变化进行故障判断。
- 具体实验操作见文件 [0.ApiExps/e12\\_health\\_basic/readme.pdf](0.ApiExps/e12_health_basic/readme.pdf)

```
## 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.关键接口介绍

- 2.13神经网络训练模型
- 为了验证模型的健康与否
- 具体实验操作见文件

[C:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e13\\_model\\_train\readme.pdf](C:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e13_model_train\readme.pdf)

```
▶ e 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\lunus\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}")
```



## 2.关键接口介绍

- 2.14测试用例自动化生成与导入
- 掌握测试用例的基本使用与配置。
- 具体实验操作见文件

[0.ApiExps\e14 DBExp\readme.pdf](#)

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

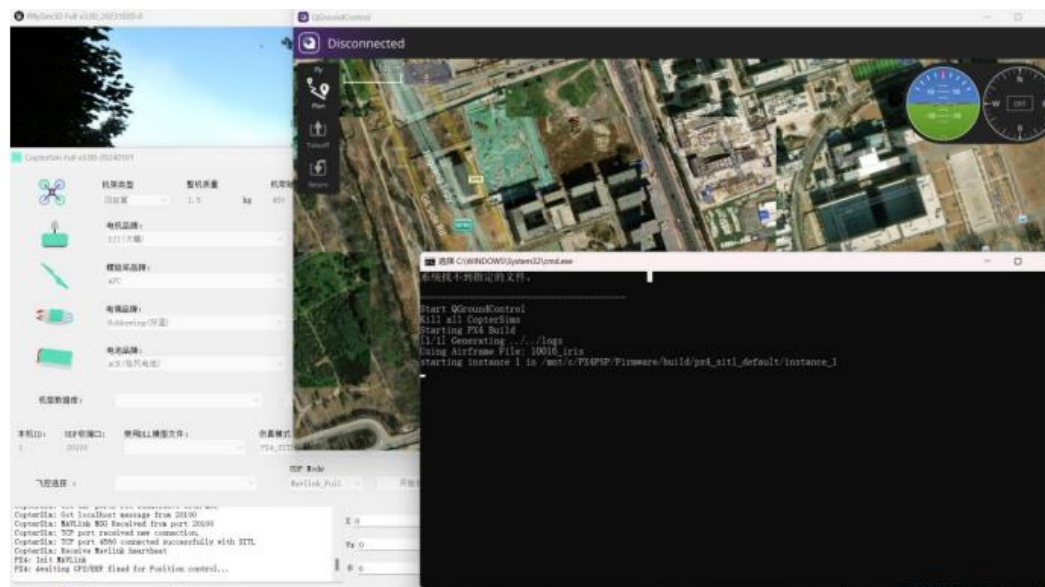




## 2.关键接口介绍

- 2.15自定义控制序列配制与使用
- 掌握自定义控制序列的基本使用与配置。
- 具体实验操作见文件

[0.ApiExps\e15\\_CmdExp\readme.pdf](#)





## 2.关键接口介绍

- 2.16自动化日志下载的使用
- 掌握自动化日志下载与配置。
- 具体实验操作见文件

[0.ApiExps\e16\\_QGCLoadExp  
\readme.pdf](#)

```
# 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.关键接口介绍

- 2.17带视觉的自定义控制序列配制与使用
- 掌握视觉的自定义控制序列配制与使用。
- 具体实验操作见文件

[0.ApiExps\e17\\_VisCmdExp\r  
eadme.pdf](#)

```
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])
```



# 大纲

---

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结

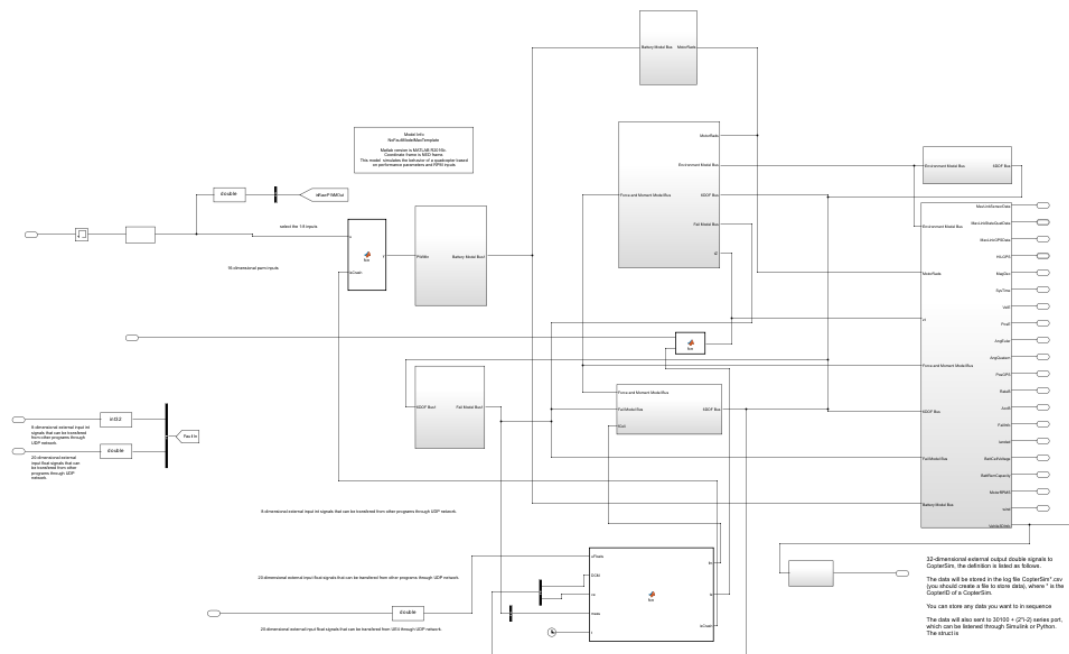


## 3.基础实验案例

### 3.1故障注入模块的最大模板

本实验没有产生任何故障效果，只是一个没有任何故障的最大模板，可以将任何故障注入模块来替换。最小模板跟最大模板的区别在于最小模板没有外部故障注入接口，最大模板有外部故障注入接口。

具体实验操作见文件[1.Basic Exps\e1\\_NoFaultModelMaxTemplate\readme.pdf](#)



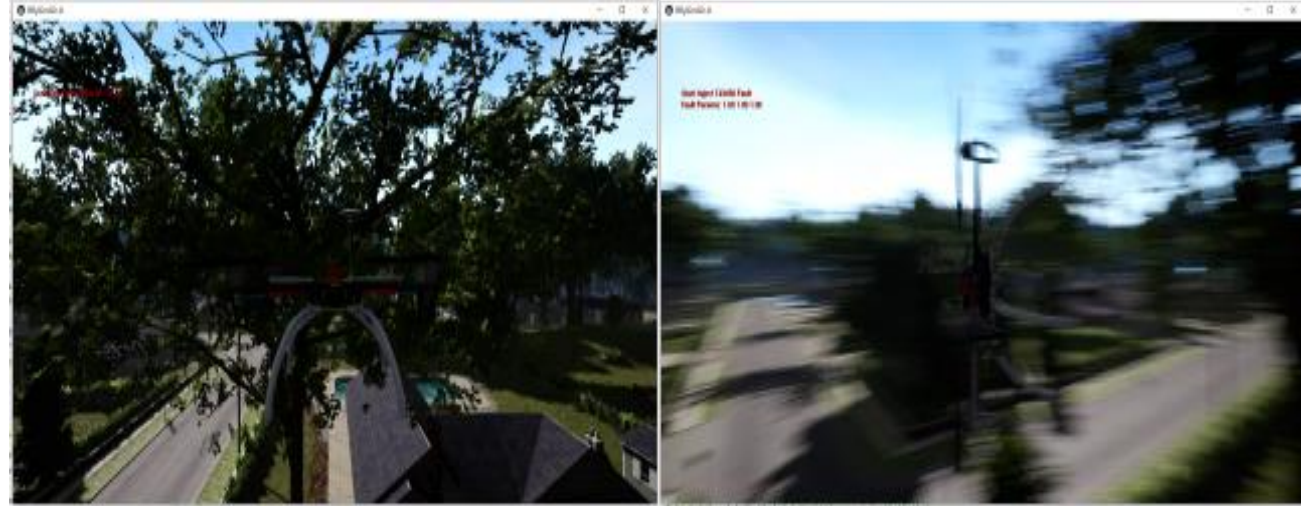


## 3.基础实验案例

### 3.2基于最大模板的GPS模块故障注入的原理

基于最大模板进行GPS模块故障建模，将故障建模的模型导出为DLL文件，再通过CopterSim加载DLL文件，最后通过udp模式注入故障码进行故障注入仿真。

具体实验操作见文件[1.Basic Exps/e2\\_GPSFault/Readme.pdf](#)



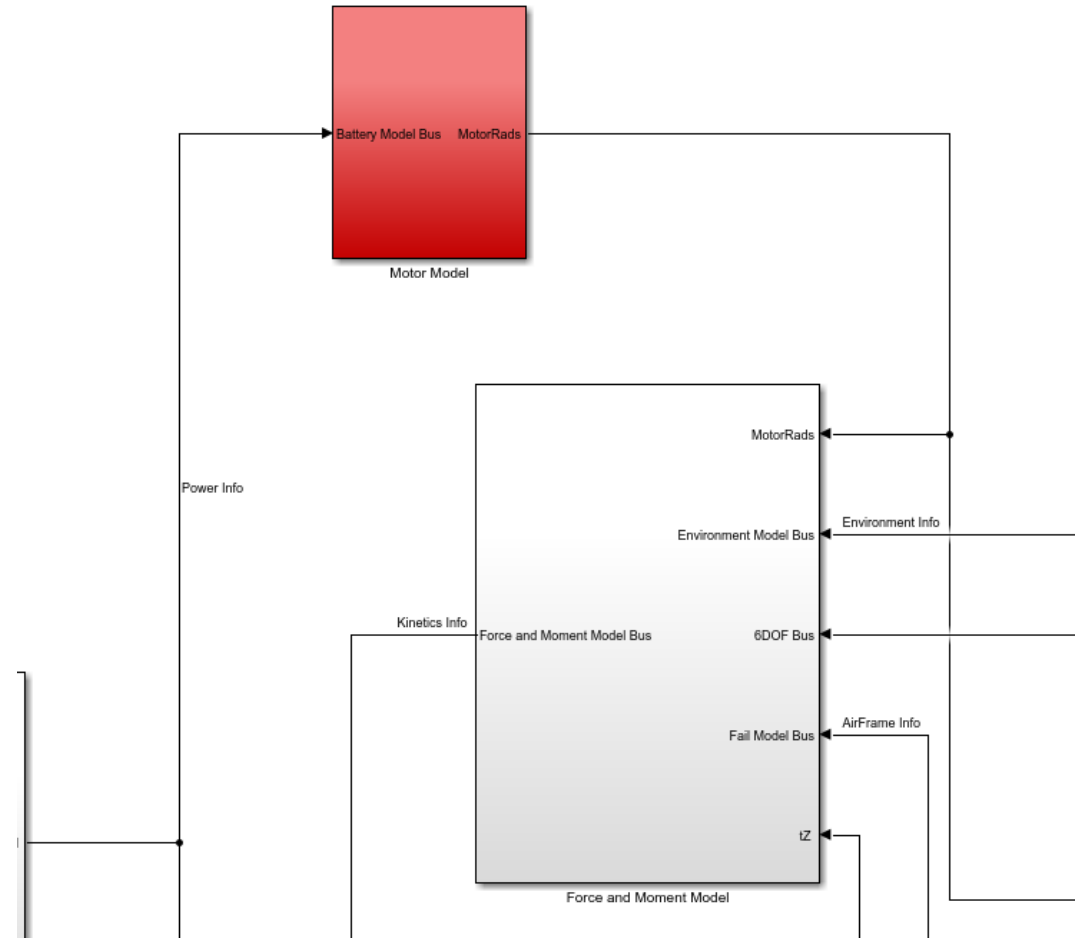


### 3.基础实验案例

#### 3.3基于最大模板的电机模块故障注入的原理

基于最大模板进行电机模块故障建模，将故障建模的模型导出为DLL文件，再通过CopterSim加载DLL文件，最后通过udp模式注入故障码进行故障注入仿真。

具体实验操作见文件[1.Basic Exps/e3\\_MotorFault/Readme.pdf](#)



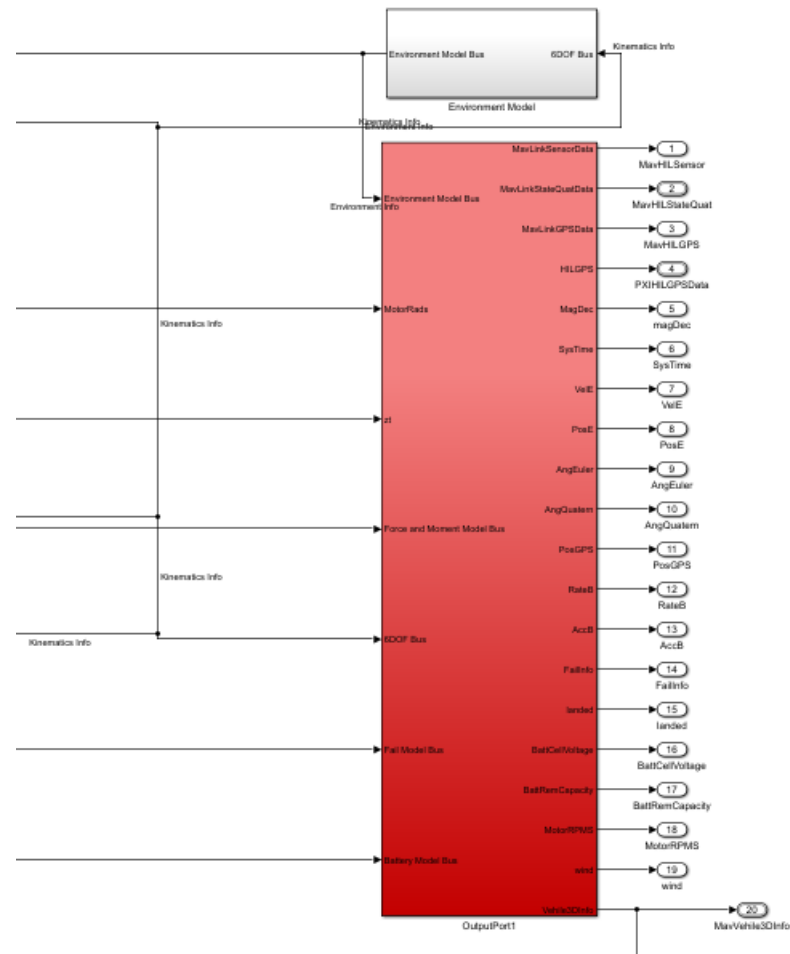


### 3.基础实验案例

#### 3.4基于最大模板的传感器模块故障注入的原理

对最大模板的传感器模块故障建模，将故障建模的模型导出为 DLL 文件，再通过 Copter Sim 加载 DLL 文件，最后通过 udp 模式（python/matlab 形式）注入故障码进行故障注入仿真。

具体实验操作见文件 [1.Basic Exps\e4\\_SensorFault\Readme.pdf](#)



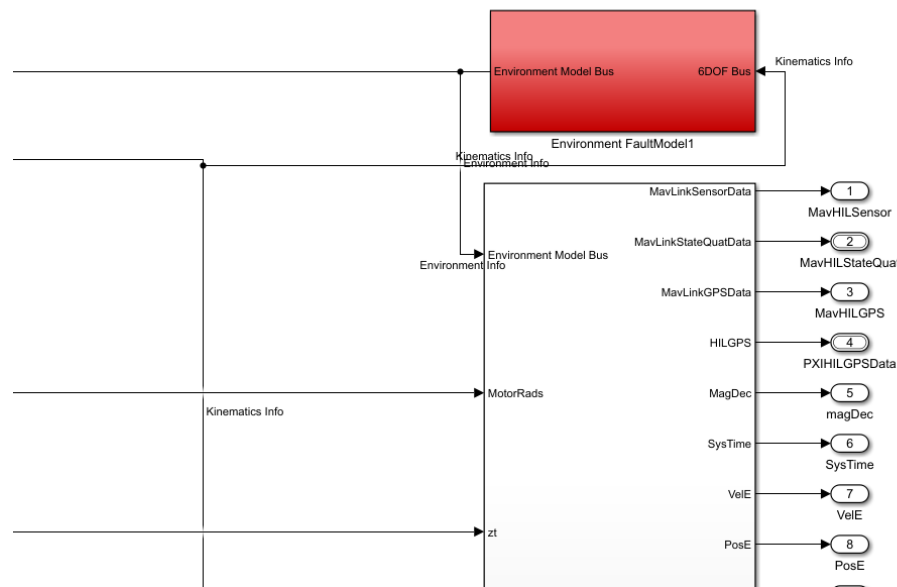




### 3.基础实验案例

#### 3.5基于最大模板的环境风模块故障注入的原理

对最大模板的环境风模块故障建模，将故障建模的模型导出为DLL文件，再通过CopterSim加载DLL文件，最后通过udp模式（python/matlab形式）注入故障码进行故障注入仿真。具体实验操作见文件[1.BasicExps/e5\\_WindFault/Readme.pdf](#)



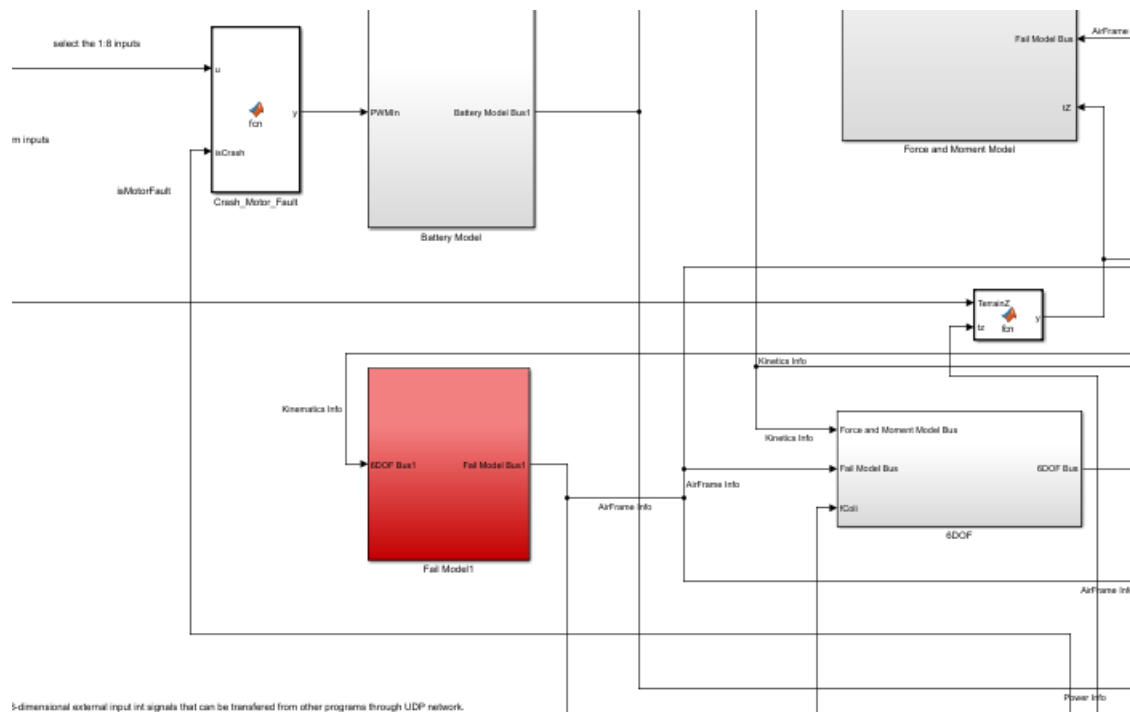


### 3.基础实验案例

#### 3.6基于最大模板的负载模块故障注入的原理

对最大模板的负载模块故障建模，将故障建模的模型导出为DLL文件，再通过CopterSim加载DLL文件，最后通过udp模式（python/matlab形式）注入故障码进行故障注入仿真。

具体实验操作见文件[1.Basic Exps\e6\\_LoadFault\Readme.pdf](#)

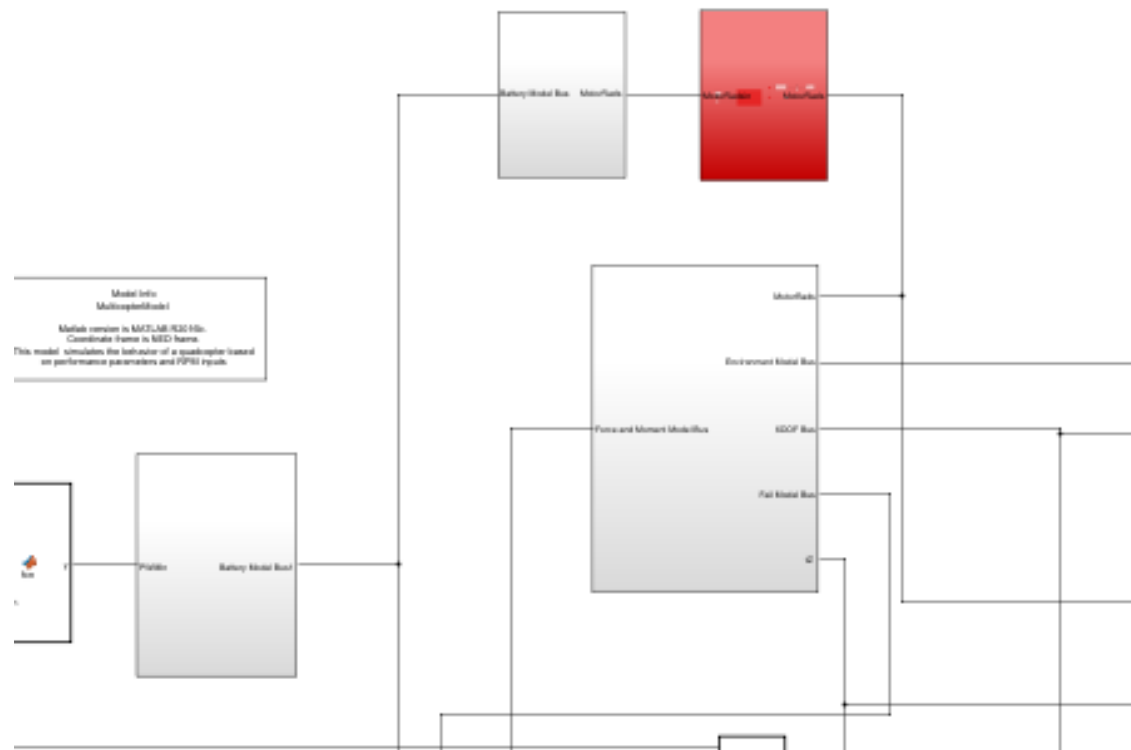




### 3.基础实验案例

#### 3.7基于最大模板的螺旋桨模块故障注入的原理

对最大模板的螺旋桨模块故障建模，将故障建模的模型导出为DLL文件，再通过CopterSim加载DLL文件，最后通过udp模式（python/matlab形式）注入故障码进行故障注入仿真。具体实验操作见文件[1.BasicExps/e7\\_PropFault/Readme.pdf](#)



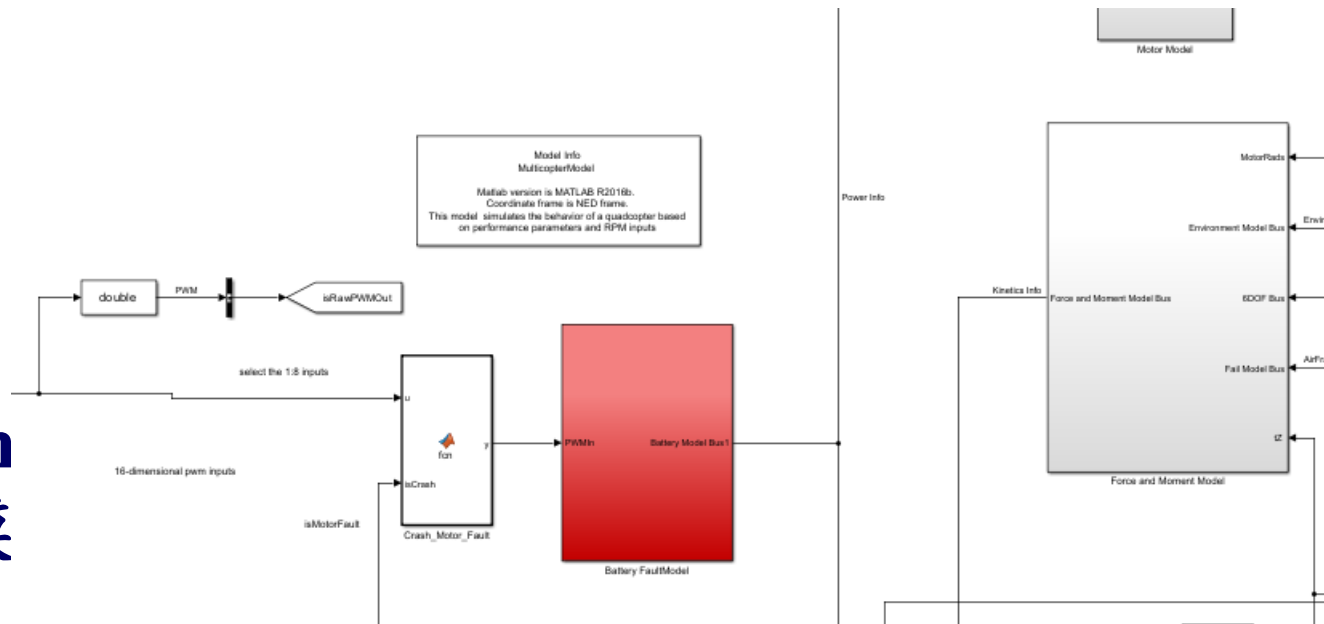


### 3.基础实验案例

#### 3.8基于最大模板的电池模块故障注入的原理

对最大模板的电池模块故障建模，将故障建模的模型导出为DLL文件，再通过CopterSim加载DLL文件，最后通过udp模式（python/matlab形式）注入故障码进行故障注入仿真。

具体实验操作见文件[1.Basic Exps\e8\\_BatteryFault\Readme.pdf](#)





# 大纲

---

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结

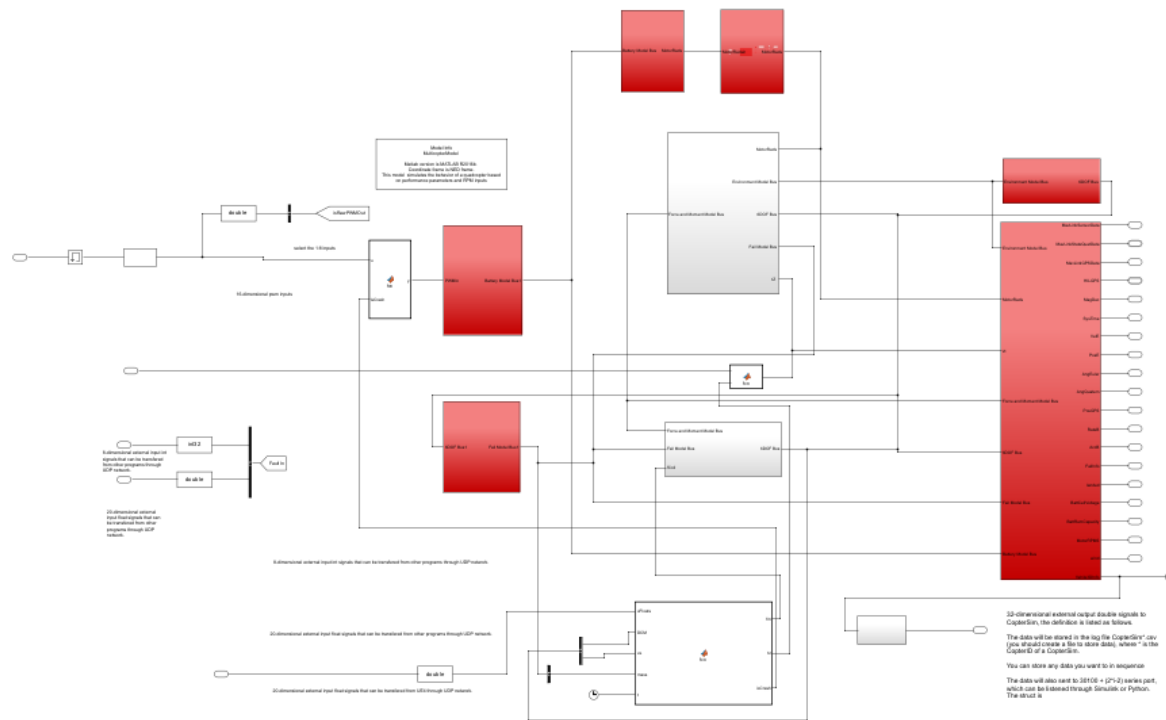


## 4.进阶案例实验

### 4.1基于最大模板的全故障模块注入的原理

对最大模板的全故障模块故障建模，将故障建模的模型导出为DLL文件，再通过CopterSim加载DLL文件，最后通过udp模式（python/matlab形式）注入故障码进行故障注入仿真。

具体实验操作见文件[2.AdvE xps/e1\\_FullFaultModelTemplate \readme.pdf](#)



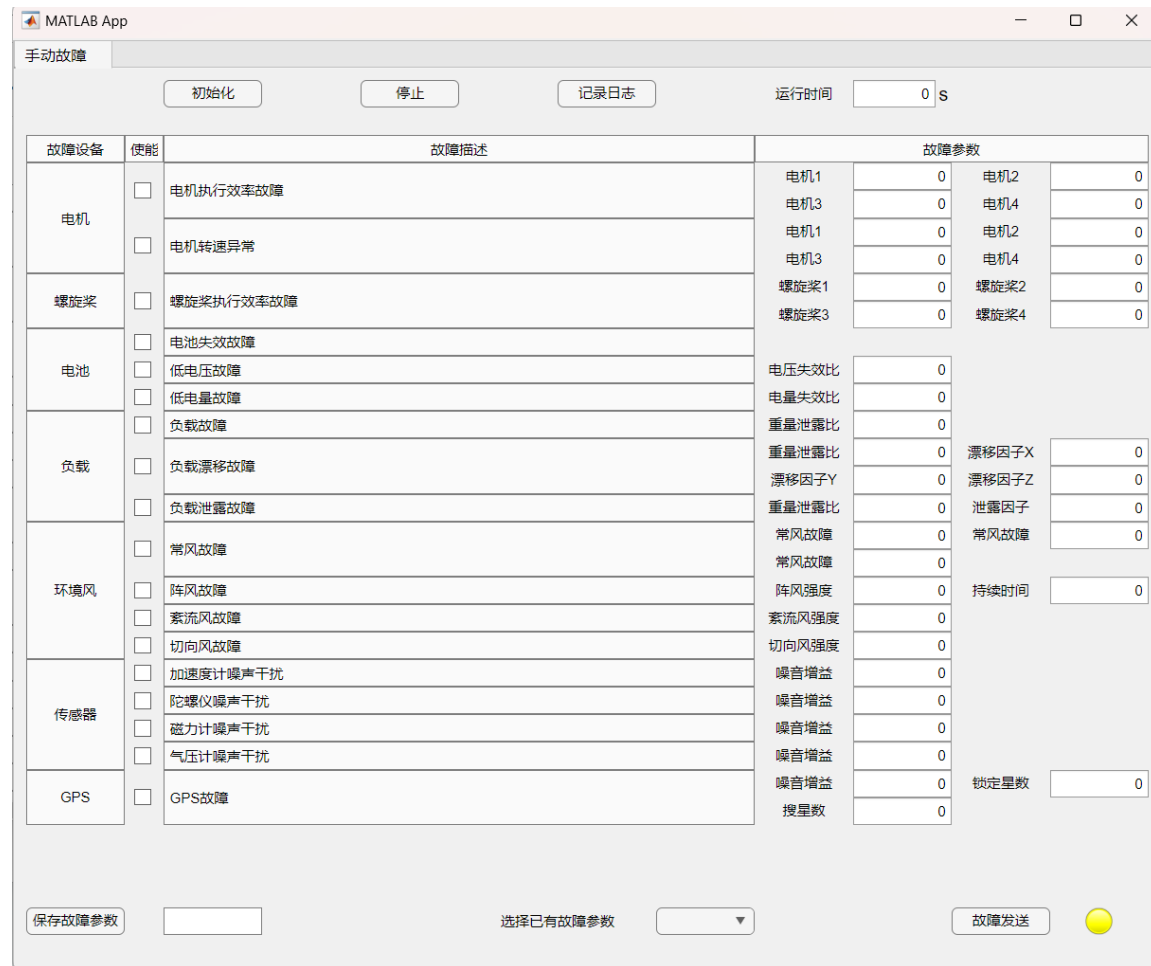


## 4.进阶案例实验

### 4.2故障生成注入界面应用程序

基于最大模板进行各种故障注入，通过MATLAB APP设计出一个可以对模型注入各种故障参数的APP，通过此方法可以明显的看出注入的故障，并能够将生成的故障注入到软件在环中，达到故障注入的效果。

具体实验操作见文件[2.AdvE xps/e2 FailureGenerator GUI APP\readme.pdf](#)





## 4.进阶案例实验

### 4.3飞控源码故障注入实验

本实验提供了一套将场景通过本实验不在依靠自动生成代码进行故障注入实验，而是直接对源码进行修改，从而实现故障注入的效果。

具体实验操作见文件

[2.AdvExps/e3 PX4](#)

[FailureGenerator/readme.pdf](#)

```
(G) 运行(R) 终端(T) 帮助(H) PythonSender.py - PythonSender - Visual Studio Code
PythonSender.py x PX4MavCtrlV4.py
PythonSender > PythonSender.py > ...
9 mav = PX4MavCtrl.PX4MavCtrl(20100)
10
11
12 #Turn on MAVLink to monitor CopterSim data and update it in real time.
13 # 可以监听数据来自20100端口, 主要是outHILStateData结构体
14 mav.InitMavLoop()
15 time.sleep(0.5)
16
17 #Turn on MAVLink to monitor CopterSim data and update it in real time.
18 # 开始监听数据来自30100端口, 主要是SOut2Simulator, 表示飞机真值数据
19 # 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
20 mav.InitTrueDataLoop()
21 time.sleep(5)
22
23 ctrls=[123450,2,120,120,120,0,120,120,0,0,0,0,0,0]
24
25 # 发送SendHILCtrlMsg数据, 在PX4内部产生rfly_ctrl的uORB消息
26 mav.SendHILCtrlMsg(ctrls)
27
28
29 time.sleep(5)
30 #Display Position information received from CopterSim
31 print(mav.uavPosNED) # 飞控数据来自20100端口
32
33 time.sleep(3)
34 print(mav.truePosNED)# 真值数据来自30100端口
35
```





## 4.进阶案例实验

- 4.4自动测试平台使用单机单实例自动化测试
- 掌握自动测试平台的基本结构和使用流程。
- 具体实验操作见文件

[2.AdvExps\e4\\_SingFramSingleInsExp\readme.pdf](#)

```
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()
```



## 4.进阶案例实验

- 4.5自动测试平台使用单机多实例自动化测试
- 掌握自动测试平台单机多实例自动化测试的基本结构和使用流程。
- 具体实验操作见文件

[2.AdvExps\e5\\_SingFrameMultiInsExp\readme.pdf](#)

Date	Frame	CaseID	S/HTL	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! 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] 25s: Exit test!	data\single\instance\Quadcopter\testCase_2_1\test1	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] 25s: Exit test!	data\single\instance\Quadcopter\testCase_2_1\test2	Yes

CaseID	FaultID	CaseDescription	FaultMode	ControlSequence	TestResult
2	123544	Frame: Quadcopter S/HTL: 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!	h: Fall: No Fall Time: No Fall Vel: No Fall Energy: No Failure 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/HTL: SITL Injection Motor Fault During Normal Flight	Decreased efficiency of motor	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!	h: Fall: No Fall Time: No Fall Vel: No Fall Energy: No Failure 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



## 4.进阶案例实验

- 4.6自动测试平台使用多机型单实例自动化测试
- 掌握自动测试平台多机型单实例自动化测试的基本结构和使用流程。
- 具体实验操作见文件  
[2.AdvExps\e6\\_MultiFrameSingleInsExp\readme.pdf](#)





## 4.进阶案例实验

- 4.7自动测试平台使用多机型多实例自动化测试
- 掌握自动测试平台多机型多实例自动化测试的基本结构和使用流程。
- 具体实验操作见文件

[2.AdvExps\e7\\_MultiFrameMultiInsExp\readme.pdf](#)

```
mav = [  
  PX4MavCtrl.PX4MavCtrl(20100),  
  PX4MavCtrl.PX4MavCtrl(20102),  
  PX4MavCtrl.PX4MavCtrl(20104),  
  PX4MavCtrl.PX4MavCtrl(20106)  
]  
  
...  
  
# # In multi-machine mode, if conf configures several drones, configure the corresponding number here.  
# mav = [  
#   PX4MavCtrl.PX4MavCtrl(20100),  
#   PX4MavCtrl.PX4MavCtrl(20102),  
#   PX4MavCtrl.PX4MavCtrl(20104),  
#   .....  
# ]  
...  

```



## 4.进阶案例实验

- 4.8安全评估
- 掌握安全评估算法的基本结构和使用流程。
- 具体实验操作见文件

[2.AdvExps\e8\\_SafetyAssExp\readme.pdf](#)

Date	Frame	CaseID	S/HTML	TestInfo	DataPath	Normal?
2023-11-12 17:06:30	Quadcopter	2	SITL	0c: Armed! 3c: Send fixed-point flight command, fly target position [0.0, -15] 15c: Send fault injection command Fault injection type:123544 Fault injection parameters:[3.0] 25c: Exit test! 0c: Armed! 3c: Send fixed-point flight command, fly target position [0.0, -15] 15c: Send fault injection command	data/SITL/single/instance/Quadcopter/TestCase_1_2Ymuv2	Yes
2023-11-12 17:15:32	Quadcopter	1	SITL	0c: Armed! 3c: Send fixed-point flight command, fly target position [0.0, -15] 15c: Send fault injection command Fault injection type:123450 Fault injection parameters:[1.0, 1.0, 1.0] 25c: Exit test!	data/SITL/mul0/instance/Test Case_1_2Quadcopter	Yes

CaseID	FaultID	CaseDescription	FaultMode	ControlSequence	TestResult
2	123544	Frame: Quadcopter S/HTML: SITL Injection Magnetometer noise During Normal Flight	Magnetometer noise interference	0c: Armed! 3c: Send fixed-point flight command, fly target position [0.0, -15] 15c: Send fault injection command Fault injection type:123544 Fault injection parameters:[3.0] 25c: Exit test!	Is Fall: No Fall Time: No Fall Vel: No Fall Energy: No Failure 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 Is Fall: No Fall Time: No Fall Vel: No Fall Energy: No Failure Trigger: No Flight Status After Fault Injection: Smooth Flight Deviation From Expected Speed After Fault Injection: None Deviation From Expected Position After Fault Injection: [0.02203068, 0.0135109, 0.09187229] Failure Safety Score: 0.888 Failure Safety Level: Minor
1	123450	Frame: Quadcopter S/HTML: SITL Injection Motor Fault During Normal Flight	Decreased efficiency of motor execution	0c: Armed! 3c: Send fixed-point flight command, fly target position [0.0, -15] 15c: Send fault injection command Fault injection type:123450 Fault injection parameters:[1.0, 1.0, 1.0] 25c: Exit test!	Is Fall: No Fall Time: No Fall Vel: No Fall Energy: No Failure Trigger: No Flight Status After Fault Injection: Smooth Flight Deviation From Expected Speed After Fault Injection: None Deviation From Expected Position After Fault Injection: [0.02203068, 0.0135109, 0.09187229] Failure Safety Score: 0.888 Failure Safety Level: Minor



## 4.进阶案例实验

- 4.9 自动化测试航迹跟踪
- 掌握航迹跟踪的基本结构和  
使用流程。
- 具体实验操作见文件

[2.AdvExps\e9 SetpointCtrlExp\readme.pdf](#)

```
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):...
```



# 大纲

---

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结

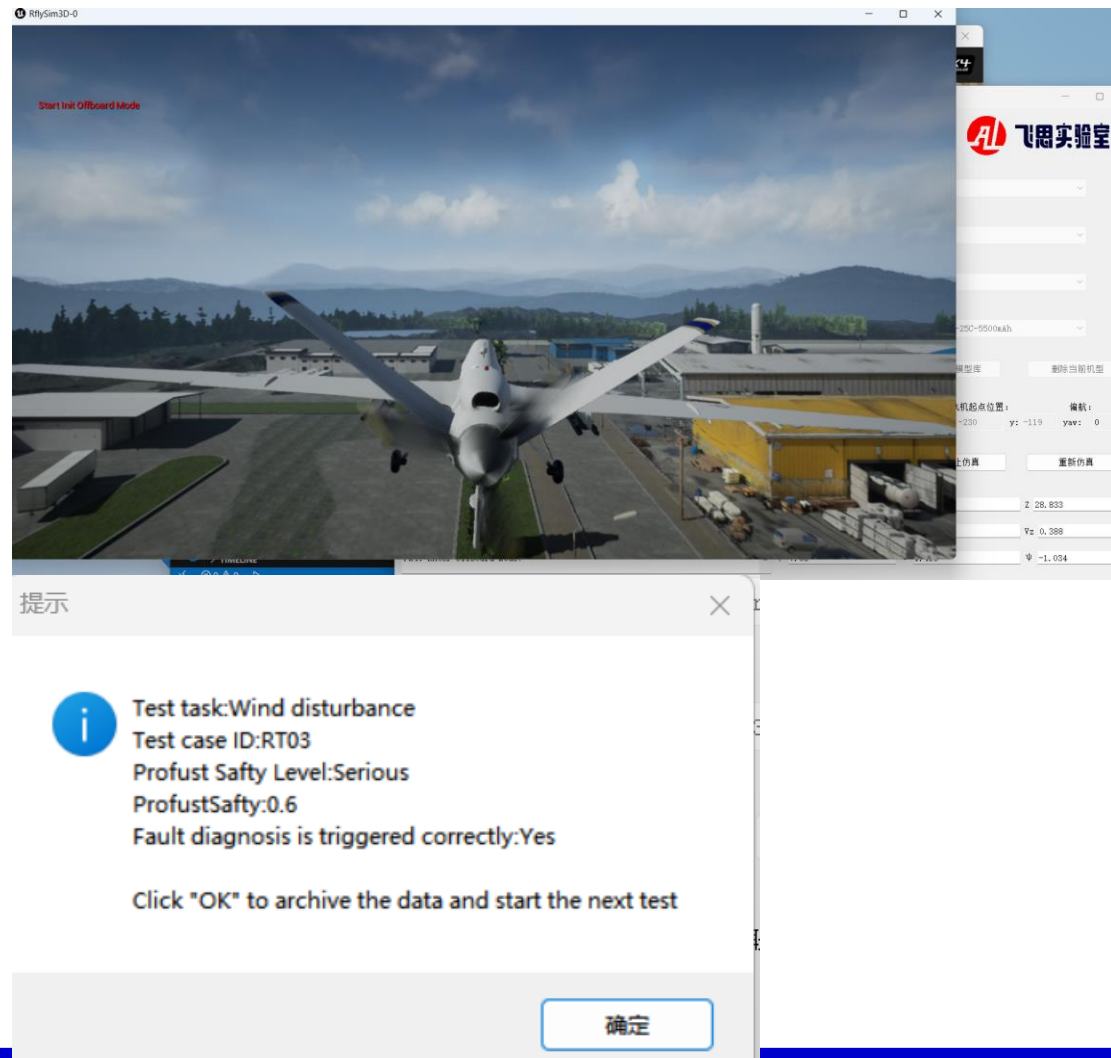


## 5.扩展案例

### 5.1电机故障安全评估实验

对电机进行pwm输出故障建模，将故障建模的模型导出为DLL文件，再通过CopterSim加载DLL文件，最后通过udp模式（python/matlab形式）注入故障码进行安全测试，并记录测试结果。

详细实验流程参见[3.CustExp s/e1\\_HealthProjPlatform/readme.pdf](#)







## 5.扩展案例

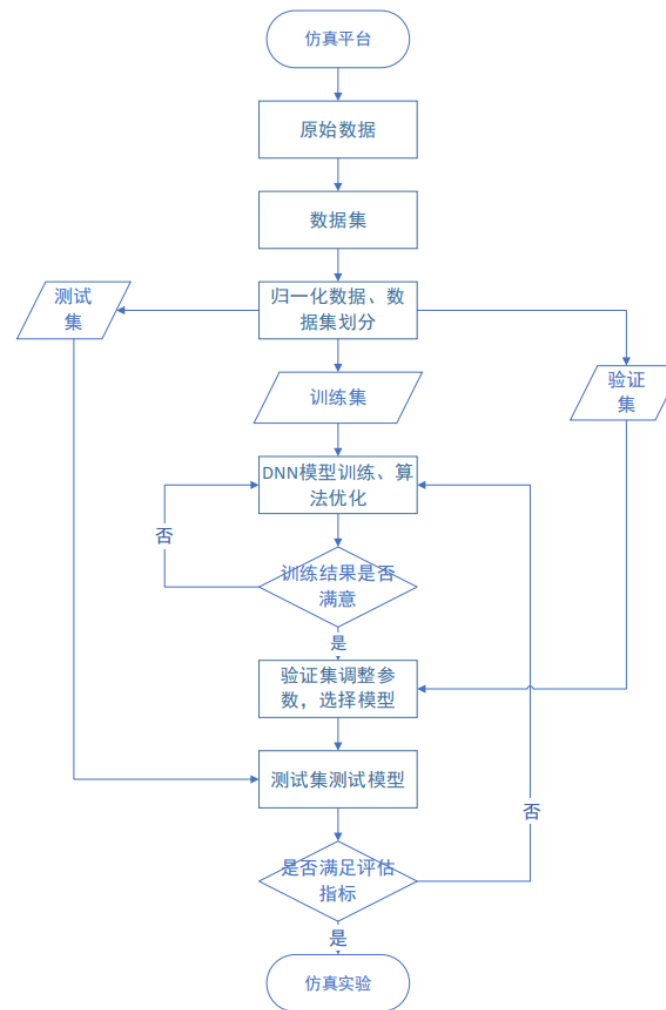
- 5.2基于数字孪生与深度学习的无人机故障诊断
- 本实验旨在探讨数字孪生技术与深度学习方法在无人机故障诊断中的应用
- 详细实验流程参见 [3.CustExps/e2 DigitalTwinExp/readme.pdf](https://github.com/3.CustExps/e2-DigitalTwinExp/readme.pdf)





## 5.扩展案例

- 5.3健康平台
- 通过对数据进行收集，进行健康评估。
- 详细实验流程参见 [3.CustExps/e3 health ass 0/readme.pdf](#)





# 大纲

---

1. 实验平台配置
2. 关键接口介绍（免费版）
3. 基础实验案例（免费版）
4. 进阶案例实验（集合版）
5. 扩展案例（完整版）
6. 小结



## 6. 小结

- 本讲主要对无人机系统的故障注入模型进行讲解，分为基础实验、进阶实验和扩展案例三部分，可以实现模型故障注入以及飞控源码注入教程。

如有疑问，请到<https://doc.rflysim.com/>查询更多信息。



RflySim更多教程



扫码咨询与交流



飞思RflySim技术交流群



谢谢！