

1. 实验名称及目的

1.1 实验名称

电机故障注入测试仿真（仅限完整版及以上版本）

1.2 实验目的

本实验需要电脑中部署Visual Studio

2022环境，部署方式见：[\[安装目](#)

[录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e6.VisualStudioInstall](#)

该例程通过平台的故障注入接口，给飞行中的飞机注入电机故障，从而实现飞机的故障坠机。

注：这里的故障注入仅使用简化协议对通信接口进行验证，与第7讲所用的故障注入接口有所区别。

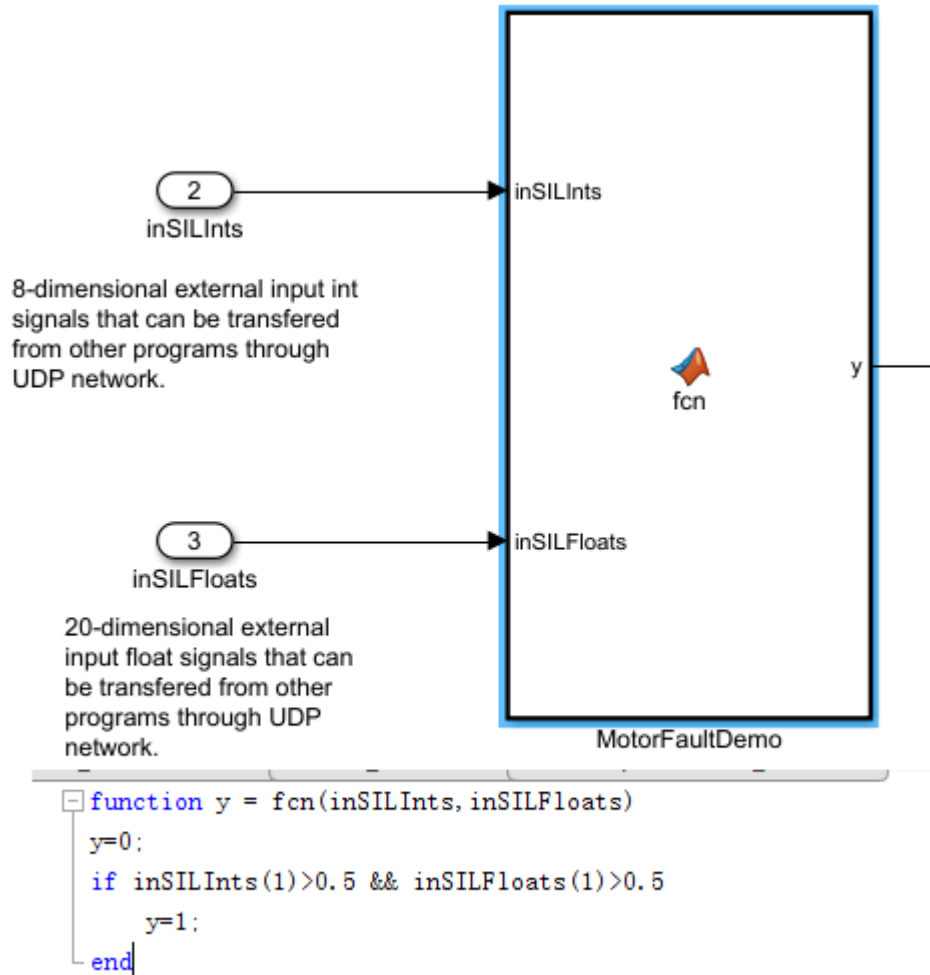
1.3 关键知识点

本实验需要电脑中部署Visual Studio

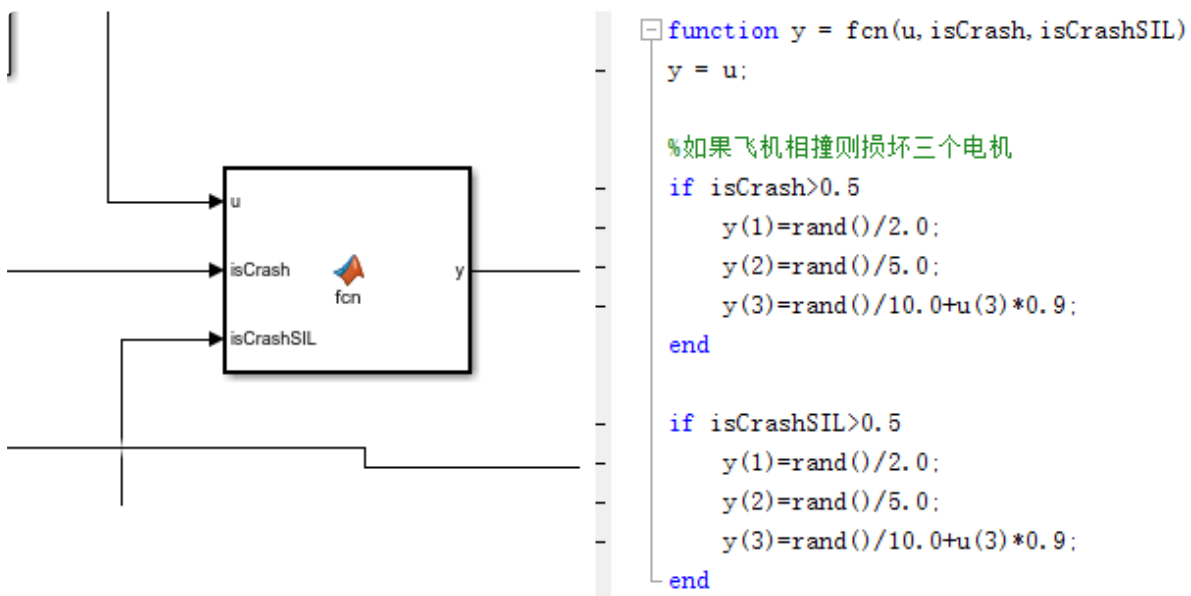
2022环境，部署方式见：[\[安装目](#)

[录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e6.VisualStudioInstall](#)

本例程在平台最大系统模版的基础上完成，Exp2_MaxModelTemp.slx中的MotorFaultDemo修改如下，输入均大于0.5时输出为1。



然后将上述模块输出作为isCrashSIL输入，模拟故障情况。



PX4MavCtrlV4.py库中的sendSILIntFloat()函数按对输入数据进行打包，以UDP的方式通过30100++2系列端口发送出去。

```
def sendSILIntFloat(self, inSILInts=[0]*8, inSILFloats=[0]*20, copterID=-1):
    checksum=1234567897
    ID=copterID
    if copterID<=0:
        ID=self.CopterID
    PortNum = 30100+(ID-1)*2
    buf = struct.pack("10i20f", checksum, ID, *inSILInts, *inSILFloats)
    self.SendBufTrue(buf, PortNum)
    #self.udp_socket.sendto(buf, (self.ip, PortNum))
```

用户在使用RflySim平台进行仿真时，CopterSim会始终监听该UDP端口，当checksum为1234567897时，将收到的数据发给inSILInts接口与inSILFloats接口。

```
#np.zeros()
silInt=np.zeros(8).astype(int).tolist()
silFloat=np.zeros(20).astype(float).tolist()
silInt[0]=1
silFloat[0]=1
mav1.sendSILIntFloat(silInt, silFloat)
print('Inject a fault, and start logging')
```

2. 实验效果

在测试仿真中，正常飞行中的飞机在注入电机故障后坠机。

3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\4.RflySimModel\3.CustExps\e0_AdvApiExps\5.ParamAPI\2.FaultInParams\1.InFaultAPITest

| 文件夹/文件名称 | 说明 |
|------------------------------------|------------------------------------|
| <./Readme.pdf> | inSILIntsFloats外部数据输入接口实验原理 |
| InFaultAPITest.py | 电机故障注入测试控制脚本。 |
| InFaultAPITest.bat | 电机故障注入测试仿真批处理文件。 |
| PX4MavCtrlV4.py | RflySim平台视觉/集群控制接口文件。 |
| Python38Run.bat | Python程序运行脚本 |

4. 运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；Python。

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；\ \台。




①：推荐配置请见：<https://rflysim.com/>

5. 实验步骤

5.1 必做实验：故障注入接口测试

Step 1：启动仿真

右键以管理员身份运行“[InFaultAPITest.bat](#)”批处理文件，在弹出的终端窗口中输入1，启动1架飞机的软件在环仿真。

| 名称 | 修改日期 | 类型 | 大小 |
|--|-----------------|----------------|--------|
|  InFaultAPITest.bat | 2023/7/27 15:02 | Windows 批处理... | 6 KB |
|  InFaultAPITest.py | 2021/12/9 19:14 | Python 源文件 | 3 KB |
|  PX4MavCtrlV4.py | 2022/12/5 22:15 | Python 源文件 | 135 KB |

```
C:\Windows\System32\cmd.exe
已复制 1 个文件。
-----
Please input UAV swarm number:1_
```

Step 2: 等待初始化完成

完成初始化。

```
PX4: Init MAVLink
PX4: Awaiting GPS/EKF fixed for Position control...
PX4: EKF2 Estimator start initializing...
PX4: Found firmware version: 1.12.3dev
PX4: Command ID: 512 ACCEPTED
PX4: Command ID: 512 ACCEPTED
PX4: Command ID: 512 ACCEPTED
PX4: GPS 3D fixed & EKF initialization finished.
PX4: Enter Auto Loiter Mode!
```



Step 3: 运行控制程序

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 InFaultAPITest.py 文件，输入 `python InFaultAPITest.py`，回车运行

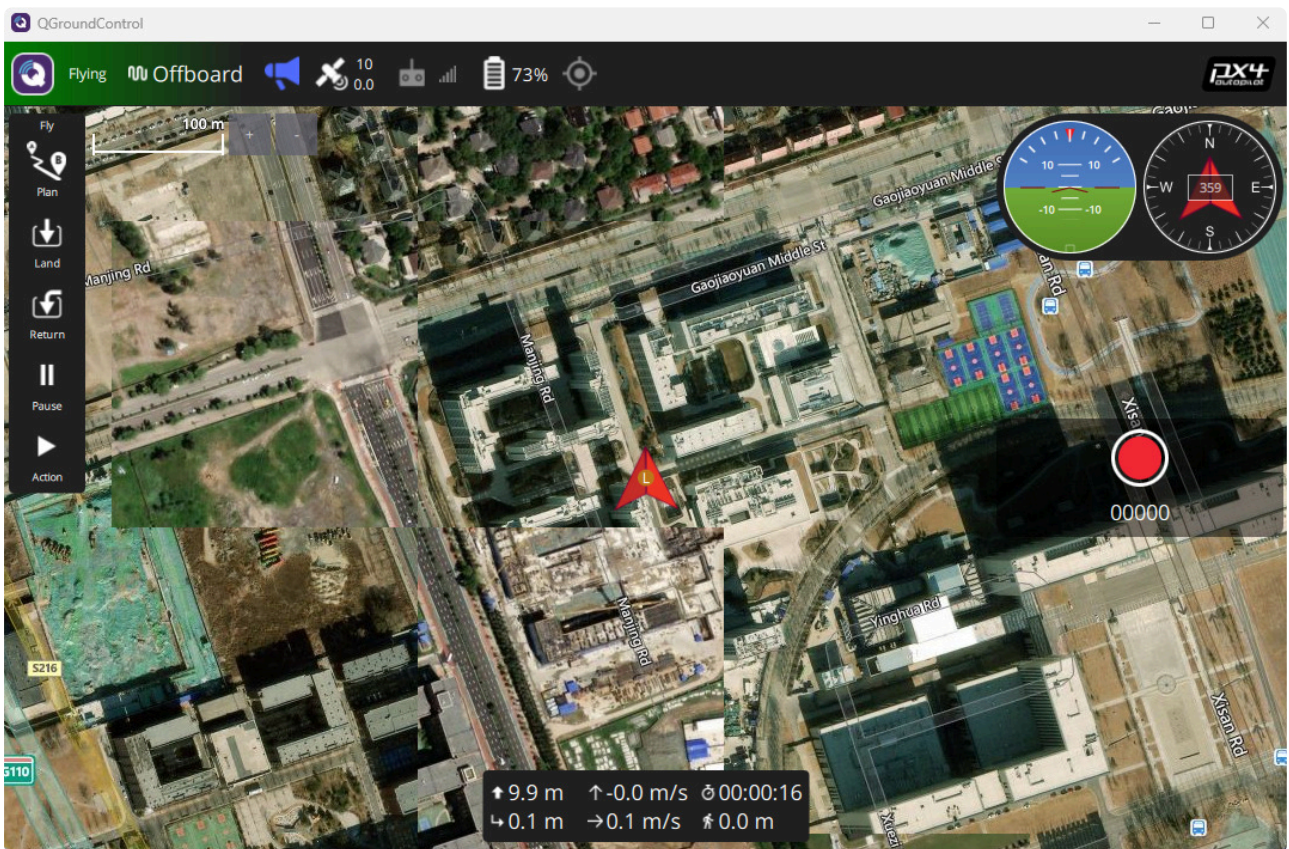
| | | | |
|--------------------|-----------------|----------------|--------|
| InFaultAPITest.bat | 2023/7/27 15:02 | Windows 批处理... | 6 KB |
| InFaultAPITest.py | 2021/12/9 19:14 | Python 源文件 | 3 KB |
| PX4MavCtrlV4.py | 2022/12/5 22:15 | Python 源文件 | 135 KB |

```
C:\Windows\system32\cmd.e. x + v
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:\RflySim\RflySimLearn\4.RflySimModel\3.CustExps\eo_AdvApiExps\5.ParamAPI\2.FaultInParams\1.InFaultAPITest>python InFaultAPITest.py
```

Step 4: 观察结果

观察飞机的飞行状态：

首先，飞机垂直起飞到10m高度后悬停



在悬停数秒后电机故障，飞机坠机，同时python运行界面显示故障注入，同时开始记录日志。

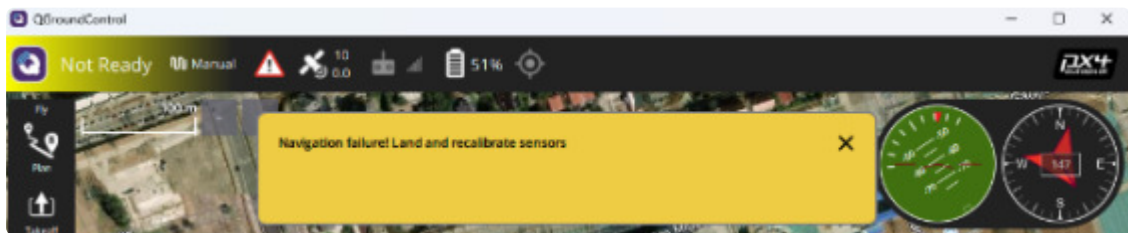


```

PX4 Armed!
5s, Arm the drone
Arm the drone!
开始起飞
Inject a fault, and start logging
[-0.0829692855477333, 0.014418662525713444, -10.009281158447266] [-0.18118765950202942, 0.041401177644729614, -11.015883445739746]
[-0.08096170425415039, 0.014190654270350933, -10.009678840637207] [-0.18168960511684418, 0.040296830236911774, -11.013781547546387]
[-0.07983513921499252, 0.013858065009117126, -10.0095796585083] [-0.18192142248153687, 0.03970284014940262, -11.010875701904297]
[-0.07576974481344223, 0.012992661446332932, -9.99972152709961] [-0.18233147263526917, 0.038606785237789154, -11.001635551452637]
[-0.07460302114486694, 0.012656991370022297, -9.994166374206543] [-0.1829315423965454, 0.03712688386440277, -10.98440933227539]
[-0.07262072712182999, 0.0119166923686862, -9.978239059448242] [-0.18390490114688873, 0.035140544176101685, -10.959709167480469]
[-0.07150185853242874, 0.011265977285802364, -9.965324401855469] [-0.1845797896385193, 0.0339394174516201, -10.945552825927734]
[-0.06963320076465607, 0.009165250696241856, -9.935005187988281] [-0.18648862838745117, 0.030902095139026642, -10.912124633789062]
[-0.06911630183458328, 0.008123639971017838, -9.921845436096191] [-0.18772496283054352, 0.029079807922244072, -10.892091751098633]
[-0.06862954795360565, 0.005182699766010046, -9.88189697265625] [-0.19044877588748932, 0.025263557210564613, -10.847625732421875]
[-0.06892061978578568, 0.0017229820368811488, -9.830265998840332] [-0.19409669935703278, 0.020415455102920532, -10.7859468460083]
[-0.07012301683425903, -0.0027727645356208086, -9.75865364074707] [-0.19849823415279388, 0.014871807768940926, -10.713186264038086]
[-0.07102957367897034, -0.005154466722160578, -9.721141815185547] [-0.20138584077358246, 0.011362898163497448, -10.667984008789062]

```

最后在飞机完全坠地停止运动后，QGC显示导航故障，python终端显示Sim End。



```
[35.9810676574707, 94.54837799072266, 2.75130558013916] [-1.6123989820480347, -0.6512558460235596, -1.0099999904632568]
[35.9810676574707, 94.54837799072266, 2.75130558013916] [-1.6123989820480347, -0.6512558460235596, -1.0099999904632568]
[35.9810676574707, 94.54837799072266, 2.75130558013916] [-1.6123989820480347, -0.6512558460235596, -1.0099999904632568]
[35.9810676574707, 94.54837799072266, 2.75130558013916] [-1.6123989820480347, -0.6512558460235596, -1.0099999904632568]
Sim End
```

5.2 选做实验（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，运行 `InFaultAPITest.py` 时，可使用VS Code（或Pycharm等工具）来打开 `InFaultAPITest.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 `InFaultAPITest.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。

```
if time.time() - startTime > 20 and flag==1:
    #np.zeros()
    silInt=np.zeros(8).astype(int).tolist()
    silFloat=np.zeros(20).astype(float).tolist()
    silInt[0]=1
    silFloat[0]=1
    mav1.sendSILIntFloat(silInt,silFloat)
    print('Inject a fault, and start logging')
    flag=2

if flag==2:
    print(mav1.uavPosNED,mav1.truePosNED)

    if time.time() - startTime > 30:
        break
```

6.参考资料

1. DLL/SO模型与通信接口..\..\PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf
2. 外部控制接口..\..\PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf
- 3.

7.常见问题

Q1:

A1:

Q2: 编译报错，无法加载库文件



A2: 这可能是由于安装平台时PX4PSP工具箱未更新到最新版，更新RflySim安装包后按照如下配置重新安装平台即可

Toolbox one-key installation script: RflySimA... — □ ×

(1) Software package installation directory
C:\PX4PSP

(2) PX4 firmware compiling command: firmware versions <= PX4-1.8 use format px4fmu-v3_default; >= PX4-1.9 use format px4_fmu-v3_default
px4_fmu-v6c_default

(3) PX4 firmware version (1: PX4-1.7.3, ... , 6: PX4-1.12.3, 7: PX4-1.13.2, 8: PX4-1.14.4, 9: PX4-1.15.0)
9

(4) PX4 firmware compiling toolchain (1: WinWSL[suitable for all versions], 2: Msys2[suitable for <= PX4-1.8], 3: Cygwin[for >=PX4-1.8])
1

(5) Whether to reinstall PSP toolbox (yes to reinstall and no to remain current installation)
yes

(6) Whether to reinstall the dependent software packages (CopterSim, QGroundControl, CopterSim, etc. About 5 minites)
no

(7) Whether to reinstall the selected compiling toolchain (yes to reinstall and no to remain unchanged, about 5 minites)
no

(8) Whether to reinstall the selected PX4 firmware source code (yes to reinstall and no to remain unchanged, about 5 minites)
no

(9) Whether to pre-compile the selected firmware with the selected command (yes to compile and no to remain unchanged, about 5 minites)
no

(10) Whether to block the actuator outputs in the PX4 firmware code ("yes" to use Simulink controller, "no" to use PX4 official controller)
no

OK Cancel