

1. 实验名称及目的

电机故障注入测试仿真：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。

2. 实验原理

2.1. 软/硬件在环仿真（SIL/HIL）的实现[1][2]

从实现机制的角度分析，可将 RflySim 平台分为运动仿真模型、底层控制器、三维引擎、外部控制和地面控制站五部分。

- **运动仿真模型：**这是模拟飞行器运动的核心部分。在 RflySim 平台中，运动仿真模型是通过 MATLAB/Simulink 开发的，然后通过自动生成的 C++代码转化成 DLL（动态链接库）文件。在使用 RflySim 平台进行软硬件在环仿真时，会将 DLL 模型导入到 CopterSim，形成运动仿真模型。这个模型在仿真中负责生成飞行器的运动响应，它拥有多个输入输出接口与底层控制器、三维引擎、地面控制站和外部控制进行数据交互，具体数据链路、通信协议及通信端口号见 [API.pdf 中的通信接口部分](#)。
- **底层控制器：**在软/硬件在环仿真（SIL/HIL）中，真实的飞行控制硬件（如 PX4 飞行控制器）被集成到一个虚拟的飞行环境中。在软件在环仿真（SIL）中，底层控制器（通过 wsl 上的 PX4 仿真环境运行）通过网络通信与运动仿真模型交互数据。在硬件在环仿真（HIL）中，它（将 PX4 固件在真实的飞行控制器（即飞控）硬件上运行）则通过串口通信与运动仿真模型进行数据交互。飞控与 CopterSim 通过串口（硬件在环 HITL）或网络 TCP/UDP（软件在环 SITL）进行连接，使用 MAVLink 进行数据传输，实现控制闭环。
- **三维引擎：**这部分负责生成和处理仿真的视觉效果，提供仿真环境和模型的三维视图，使用户能够视觉上跟踪和分析飞行器的运动。CopterSim 发送飞机位姿、电机数据到三维引擎，实现可视化展示。
- **外部控制（offboard）：**从仿真系统外部对飞行器进行的控制，包括自动飞行路径规划、远程控制指令等。在平台例程中主要通过地面控制站（QGC）、MATLAB 和 Python 调用对应接口实现。

2.2. 通过外部控制接口（python）注入故障

故障注入脚本 InFaultAPITest.py 脚本中依次调用了 RflySim 平台飞机控制接口协议文件 PX4MavCtrlV4.py 中定义的以下接口函数

启用 Mavlink 消息监听循环

```
mav1.InitMavLoop()
```

配置 CopterSim 通信模式，该函数的参数定义如下：

```
def InitMavLoop(self,UDPMode=2):
    """ Initialize MAVLink listen loop from CopterSim
        0 and 1 for UDP_Full and UDP_Simple Modes, 2 and 3 for MAVLink_Full and
        MAVLink_Simple modes, 4 for MAVLink_NoSend
        The default mode is MAVLink_Full
    """
```

默认通信模式为 **Mavlink_Full**: Python 直接发送 MAVLink 消息给 CopterSim, 再转发给 PX4, 数据量较大适合单机控制; 适合单机或少量飞机仿真, 无人机数量小于 4;

启用飞机姿态仿真数据（模型真值）监听循环

```
mav1.InitTrueDataLoop()
```

通过 30101 系列端口接收 CopterSim 转发的模型真值

启用外部控制（offboard）

```
mav1.initOffboard()
```

使 px4 控制器进入外部控制模式, 且以 30HZ 的频率发送 offboard 指令

解锁飞机

```
mav1.SendMavArm(True)
```

设定起飞高度

```
mav1.SendPosNED(0,0,-10)
```

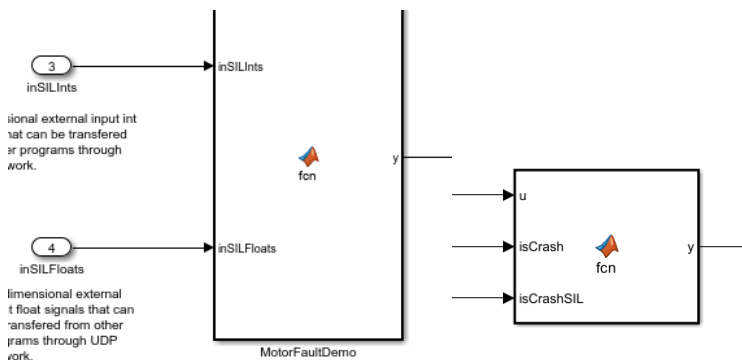
固连坐标系为北东地, 设置飞机起飞高度为 10m

电机故障注入

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。

```
silInt=np.zeros(8).astype(int).tolist()
silFloat=np.zeros(20).astype(float).tolist()
silInt[0]=1
silFloat[0]=1
mav1.sendSILIntFloat(silInt,silFloat)
```

这里的值会通过 UDP 30100 转发到 CopterSim 下的 DLL 模型的 inSILFloats 和 inSILInts 接口, 打开 Exp2_MaxModelTemp.slx 可见, 模型会通过以下两个函数作出判断: 当 InsilInt 的第一位或 InsilFloat 的第一位大于 0.5 时触发电机故障



3. 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。

4. 文件目录

文件夹/文件名称	说明
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。

5. 运行环境




序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台	\	\
3	Python3.8	\	\

①：推荐配置请见：<https://doc.rflysim.com>

6. 实验步骤

Step 1:

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:




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:

打开 InFaultAPITest.bat.py 文件并运行。

 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:\Users\admin> cd C:\PX4PSP\RflySimAPIs\OtherVehicleTypes\ExternalCtrlAPI\PythonAPI > SILIntFloatAPITest.py > ...
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(20100)
10 # mav2 = PX4MavCtrl.PX4MavCtrl(20102)
11 # mav2 = PX4MavCtrl.PX4MavCtrl(20104)
12 # mavN --> 20100 + (N-1)*2
13
14
15 # Init MAVLink data receiving loop
16 mav1.InitMavLoop()
17 #mav2.InitMavLoop(), ...
18
19 time.sleep(0.5)
20 mav1.InitTrueDataLoop()

```

PROBLEMS 32 OUTPUT TERMINAL DEBUG CONSOLE

```

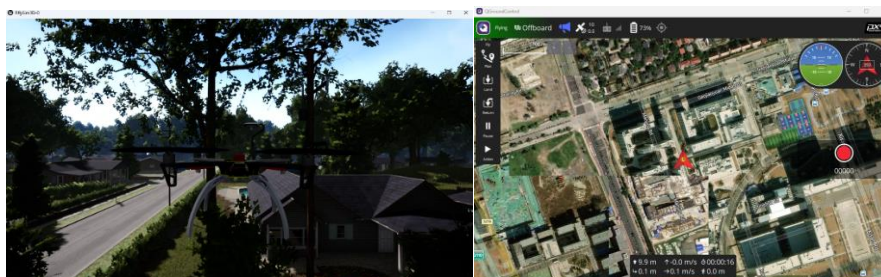
PS C:\Users\admin> & E:/python38/python.exe c:/PX4PSP/RflySimAPIs/OtherVehicleTypes/ExternalCtrlAPI/PythonAPI/SILIntFloatAPITest.py
PX4 Armed!
5s, Arm the drone
Arm the drone!
开始起飞

```

Step 4:

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 终端显示故障注入, 同时开始记录日志。



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。



7. 参考资料

[1]. DLL/SO 模型与通信接口 [..\..\API.pdf](#)

[2]. 外部控制接口 [..\..\API.pdf](#)

[3].

8. 常见问题

Q1.

A1.