

1. 实验名称及目的

1.1 实验名称

Dll模型接口FaultParamAPI.FaultInParams Python接口实验（仅限完整版及以上版本）

1.2 实验目的

FaultParamAPI.FaultInParams为RflySim平台DLL模型的故障参数输入接口，32维double型。该例程介绍如何通过调用DllSimCtrlAPI.py库中的sendModelInParams()函数向FaultParamAPI.FaultInParams接口传入数据。

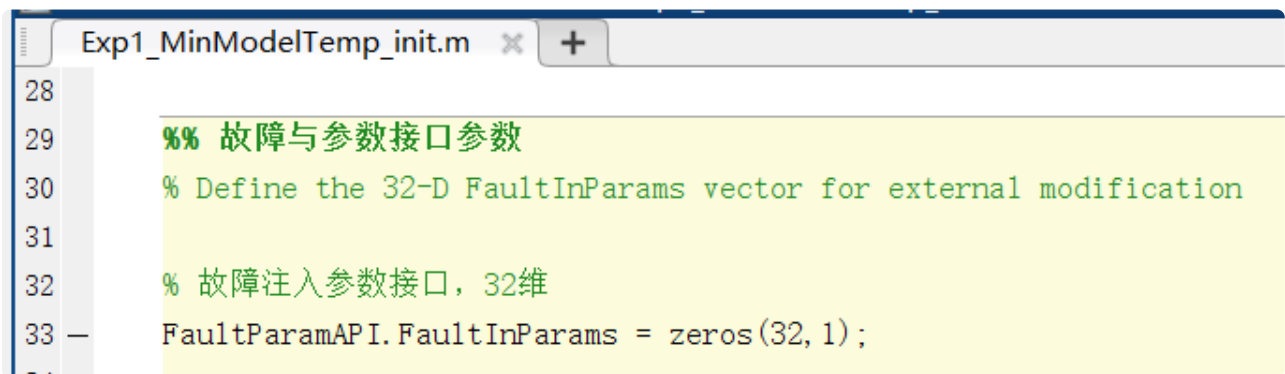
1.3 关键知识点

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\6.VisualStudioInstall](#)

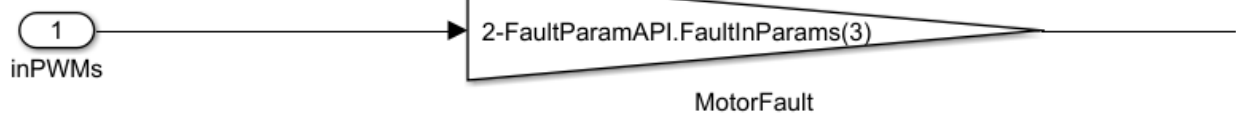
在 [Exp1_MinModelTemp_init.m](#) 中对FaultParamAPI.FaultInParams进行初始化：



```
Exp1_MinModelTemp_init.m x +
28
29 %% 故障与参数接口参数
30 % Define the 32-D FaultInParams vector for external modification
31
32 % 故障注入参数接口，32维
33 - FaultParamAPI.FaultInParams = zeros(32, 1);
```

DLL模型中将2-FaultParamAPI.FaultInParams(3)作为inPWMs输入的增益。

PWM inputs from autopilot (16-dimensional float vector, range from -1-1)



CopterSim中FaultParamAPI.FaultInParams结构体如下：

```
struct PX4ModelInParams{  
  
int checksum;//1234567891 for FaultInParam  
  
uint32_t Bitmask;  
  
double InParams[32];
```

DllSimCtrlAPI.py库中的sendModelInParams()函数按以上结构体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。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1234567891时，将收到的数据发给FaultParamAPI.FaultInParams。

而DLL模型中将2-FaultParamAPI.FaultInParams(3)作为了输入的增益，因此当FaultParamAPI.FaultInParams(3) \geq 2时，输入会为0，电机转速为0，四旋翼将会坠地。

2.实验效果

运行Exp1_MinModelTemp.bat启动软件在环仿真，运行InFaultAPITest.py程序，可以依次看到四旋翼从地面起飞、悬停在空中、坠地。

3.文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\4.RflySimModel\3.CustExps\3.AdvApiExps\5.ParamAPI\2.FaultInParams\3.Python](#)

文件夹/文件名称	说明
Exp1_MinModelTemp.dll	修改后的动态链接库
Exp1_MinModelTemp.slx	Simulink模型文件
Exp1_MinModelTemp_init.m	模型参数文件
Exp1_MinModelTemp.bat	软件在环仿真启动脚本
InFaultAPITest.py	Python测试例程
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；RflySi工具链；MATLAB 2017B及以上^③；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://rflsim.com/doc/zh/1/Hardware.html>

4.2 硬件要求

笔记本/台式电脑^① 1台；\\台；\\台；\\台。

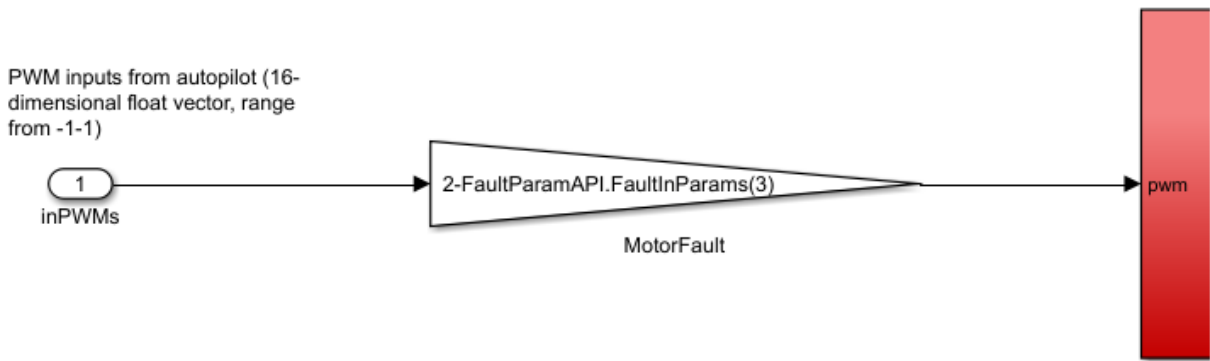
①：推荐配置请见：<https://rflsim.com/>

5. 实验步骤

5.1 必做实验：故障注入参数接口使用

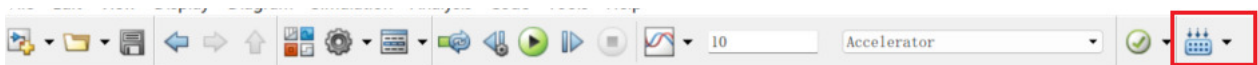
Step 1：修改模型并编译

在inPWMs输入链路上增加Gain，设置为2-FaultParamAPI.FaultInParams(3)。



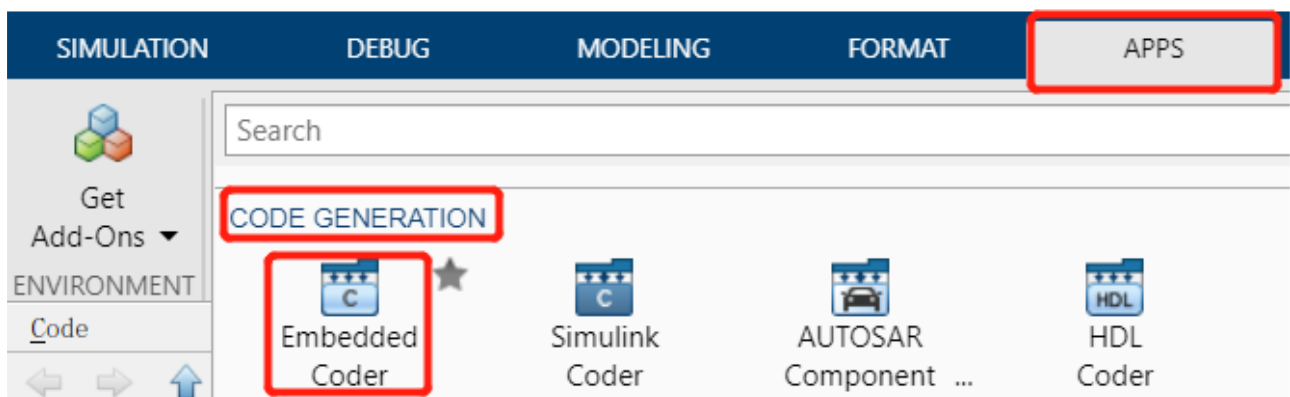
修改完成后，保存，并在Simulink中点击编译命令。

对于MATLAB 2019a及之前版本，工具栏样式见下图，直接点击它的编译按钮“Build”即可。

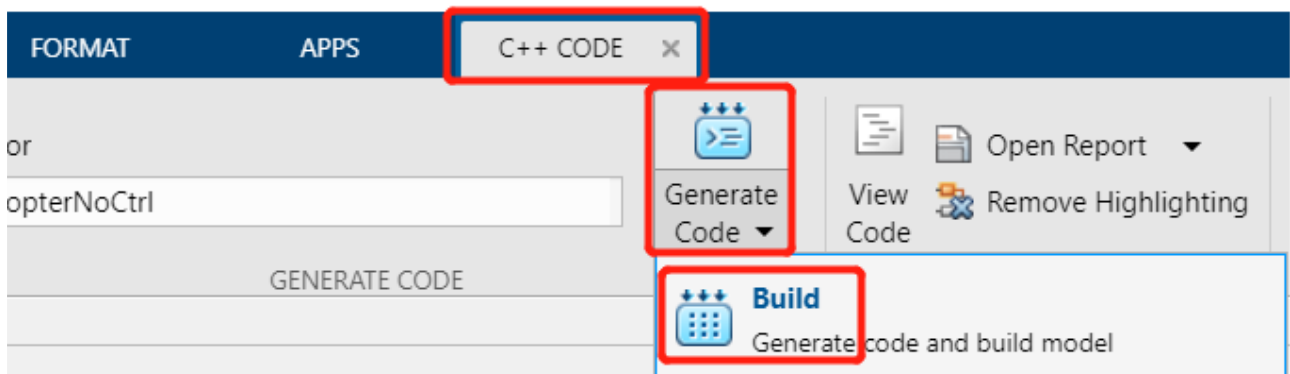


对于2019b及之后版本，点击APPS - CODE GENERATION - Embedded Coder才能弹出代码生成工具栏，在其中如下图所示点击“C++CODE” - “Generate Code” - “Build”按钮就能编译生成代码。

MulticopterCtrlVelocity/Force and Moment Model - Simulink

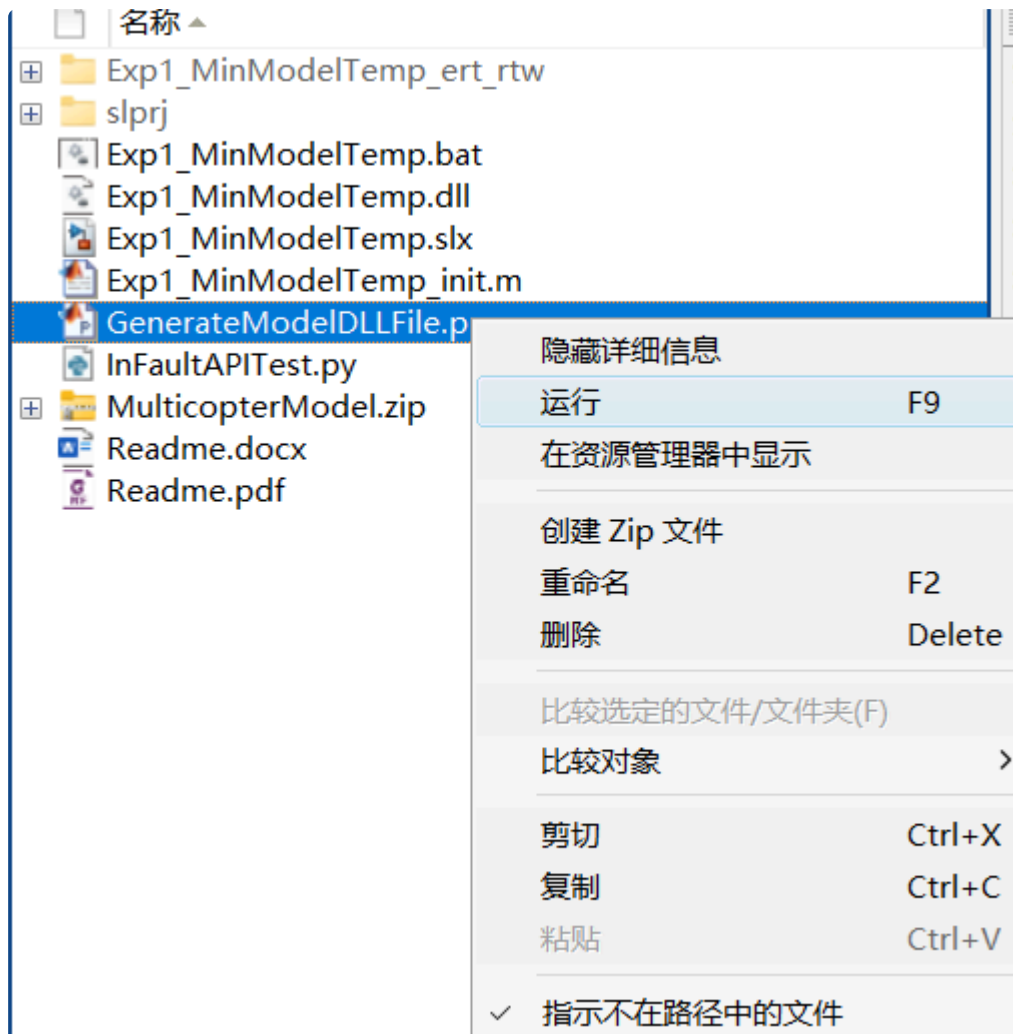


k



Step 2: 生成DLL模型

右键运行 GenerateModelDLLFile.p 文件或在命令行窗口中输入 GenerateModelDLLFile后回车，得到修改后的动态链接库Exp1_MaxModelTemp.dll。



Step 3: 开启仿真

运行Exp1_MaxModelTempSITL.bat,

Exp1_MinModelTemp_ert_rtw	2024/5/7 16:15	文件夹	
slprj	2024/5/7 16:15	文件夹	
Exp1_MinModelTemp.bat	2024/4/18 16:30	Windows 批处理...	6 KB
Exp1_MinModelTemp.dll	2024/5/7 16:15	应用程序扩展	221 KB
Exp1_MinModelTemp.slx	2024/5/7 17:00	Simulink Model	63 KB
Exp1_MinModelTemp_init.m	2024/5/7 14:49	Objective C 源文件	3 KB
GenerateModelDLLFile.p	2024/4/30 16:04	MATLAB.p.23.2.0	7 KB
InFaultAPITest.py	2024/5/7 16:38	Python 源文件	2 KB
MulticopterModel.zip	2024/5/7 16:15	压缩(zipped)文件...	95 KB
Readme.pdf	2024/4/23 0:49	Foxit PhantomP...	577 KB
Readme.docx	2024/5/7 16:57	Microsoft Word ...	413 KB

输入1，启动1架无人机的软件在环仿真。

```

C:\WINDOWS\System32\cmd.exe
1 file(s) copied.
-----
Please input UAV swarm number:1

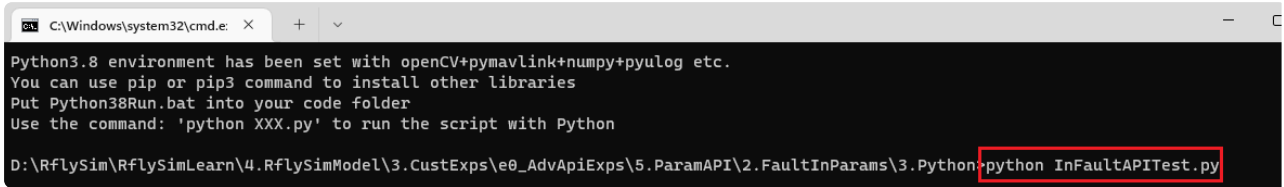
```

等待RflySim3D初始化完成。



Step 4: 运行Python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`



```
C:\Windows\system32\cmd.exe
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\e0_AdvApiExps\5.ParamAPI\2.FaultInParams\3.Python>python InFaultAPITest.py
```

Step 5: 观察结果

可以依次看到以下现象：

四旋翼解锁起飞。



四旋翼悬停在空中。

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

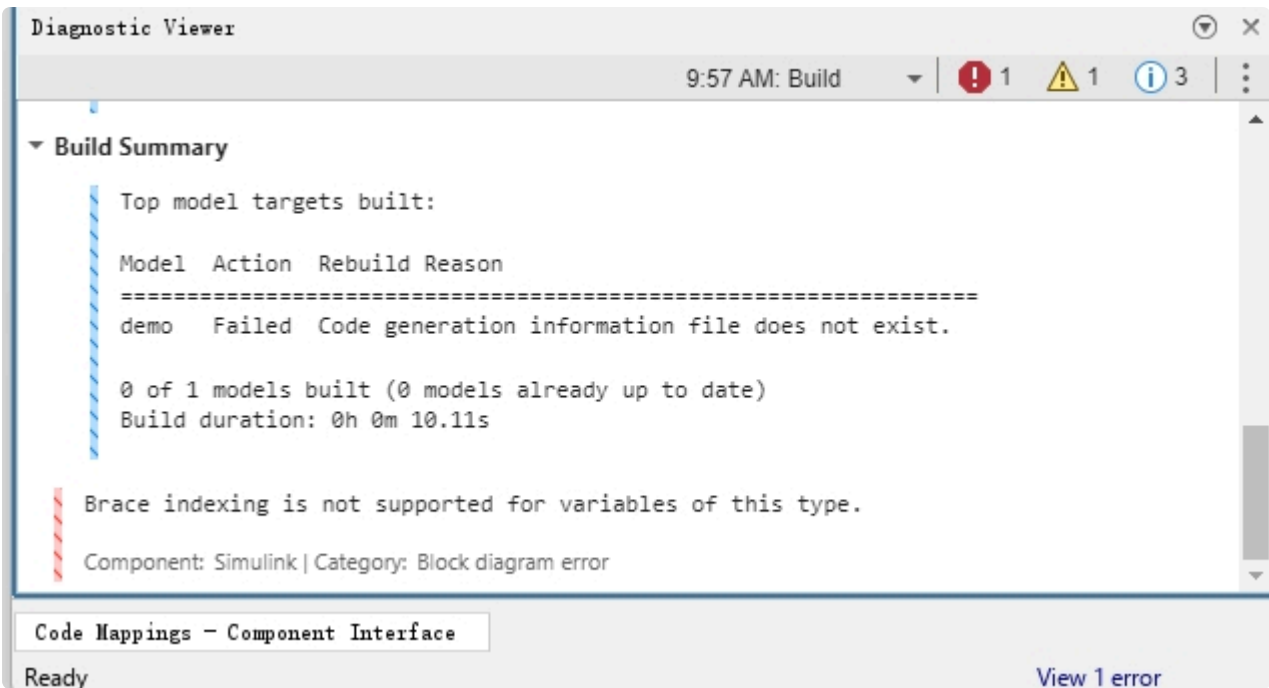
- 请自行使用VS Code阅读 [InFaultAPITest.py](#) 源码，通过程序跳转，了解每条代码的执行原理；再通过调试工具，验证每条指令的执行效果。

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: 未正确安装visual studio c++编译环境并配置mex，导致Simulink文件编译失败



A1: 首先将低于当前MATLAB版本的Visual Studio C++编译环境安装到VS默认安装目录，然后在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支持的编译器（例如Visual C++ 2017），命令行显示“MEX 配置使用 ‘Microsoft Visual C++ 2017’ 以进行编译”的字样说明安装正确。详细环境配置参考” [RflySim平台安装目录]\RflySimAPIs\4.RflySimModel\API.pdf “中的环境配置

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