

1. 实验名称及目的

1.1. 实验名称

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

1.2. 实验目的

FaultParamAPI.DynModiParams 为 RflySim 平台 DLL 模型的动态参数输入接口，64 维 double 型。该例程介绍如何在仿真过程中通过调用 DllSimCtrlAPI.py 库中的 sendDynModiParams() 函数向 FaultParamAPI.DynModiParams 接口传入数据。

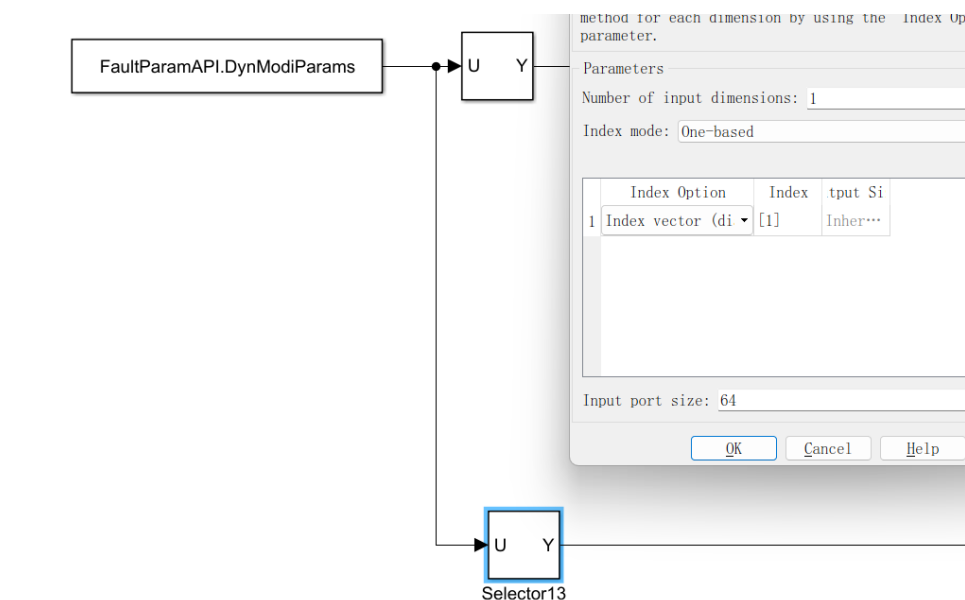
1.3. 关键知识点

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\c6_VisualStudioInstall](#)

使用该参数接口方法示例：在 Exp1_MinModelTemp_init.m 中对 FaultParamAPI.DynModiParams 进行初始化：

```
40 % 模型动态修改参数接口，64维
41 FaultParamAPI.DynModiParams = zeros(64, 1);
```

DLL 模型中将 FaultParamAPI.DynModiParams 的第 1 维作为 3DOutput 模块的 3DType 输入，3DType 对应了 MavVehileStateInfo 结构体中的 vehicleType，决定了 RflySim3D 中载具的显示模型。



CopterSim 中 FaultParamAPI.DynModiParams 结构体如下：

```
struct PX4DynModiParams{
    int checksum;//1234567893
    int CopterID;
```

```
uint64_t Bitmask;
double InParams[64];
};
```

其中 Bitmask 为位使能标志位，InParams[64]为传入模型的参数，使用实例如下：

```
dll.sendDynModiParams(0b1,[100])
```

其中 Bitmask 为 0b1，代表传入的 64 维参数中只有第 1 维具有修改权限，其他维即使传入参数也无法成功修改。

sendDynModiParams ()函数：

DllSimCtrlAPI.py 库中的 sendDynModiParams ()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 1234567893 时，将收到的数据发给 FaultParamAPI.DynModiParams。

```
def sendDynModiParams(self, Bitmask=0, InParams=[0]*64, copterID=-1):
    """
    checksum = 1234567893 # A fixed checksum value for the data packet
    ID = copterID
    if copterID <= 0:
        ID = self.CopterID
    PortNum = 30100 + (ID - 1) * 2 # Calculate the port number based on th
    InParams = self.fillList(InParams, 64) # Ensure that the InParams list
    buf = struct.pack("iiQ64d", checksum, copterID, Bitmask, *InParams) #
    self.udp_socket.sendto(buf, (self.ip, PortNum)) # Send the data packet
```

2. 实验效果

运行 Exp1_MinModelTemp.bat 启动软件在环仿真，运行 DynModiParamsTest.py 程序，可以在 RflySim3D 中依次看到四旋翼解锁起飞、悬停、三维显示模型切换为小型固定翼（vehicleType 为 100）

3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\4.RflySimModel\3.CustExps\e0_AdvApiExps\5.ParamAPI\3.DynModiParams](#)

文件夹/文件名称	说明
..\Intro.pdf	inDoubCtrls 接口实验原理
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
DynModiParamsTest.py	Python 测试例程
Python38Run.bat	Python 程序运行脚本

4. 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 工具链	\	\
3	MATLAB 2017B 及以上	\	\
4	Python		

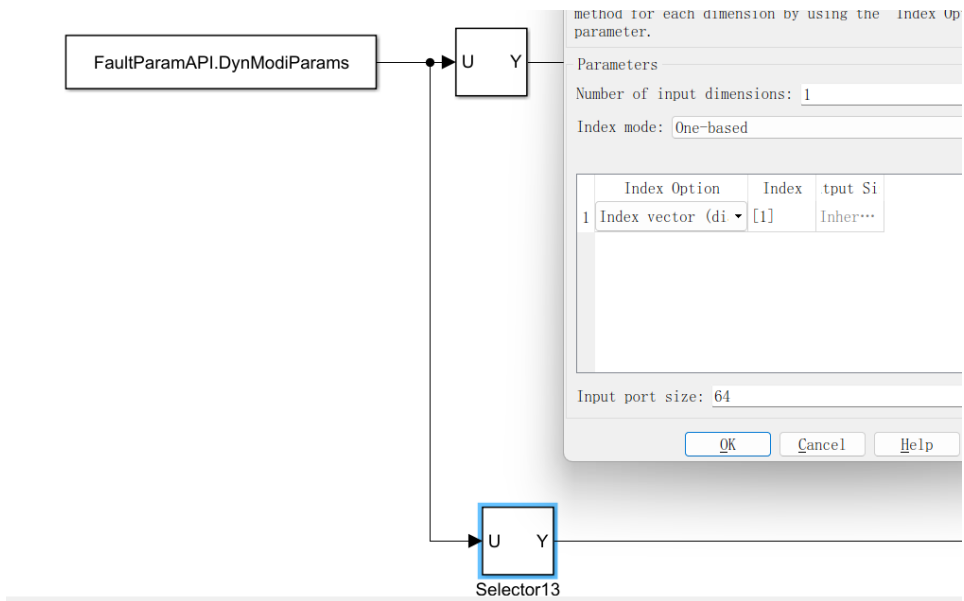
①：推荐配置请见：<https://rflysim.com/doc/en>

5. 实验步骤

5.1 必做实验：模型参数接口使用

Step 1: 修改模型并编译

将 3DOutput 模块的输入由 ModelParam_3DType 改为 FaultParamAPI.DynModiParams(1)。

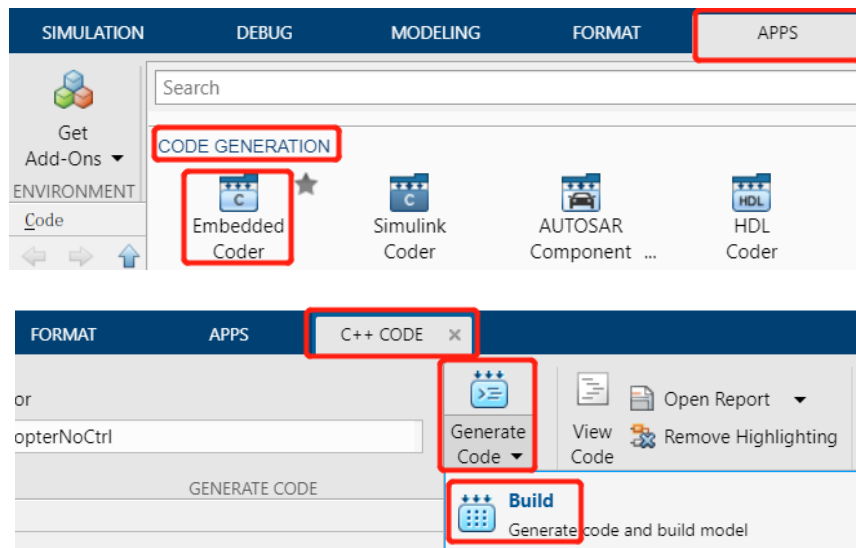


修改完成后，保存，并在 Simulink 中点击编译命令。

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。



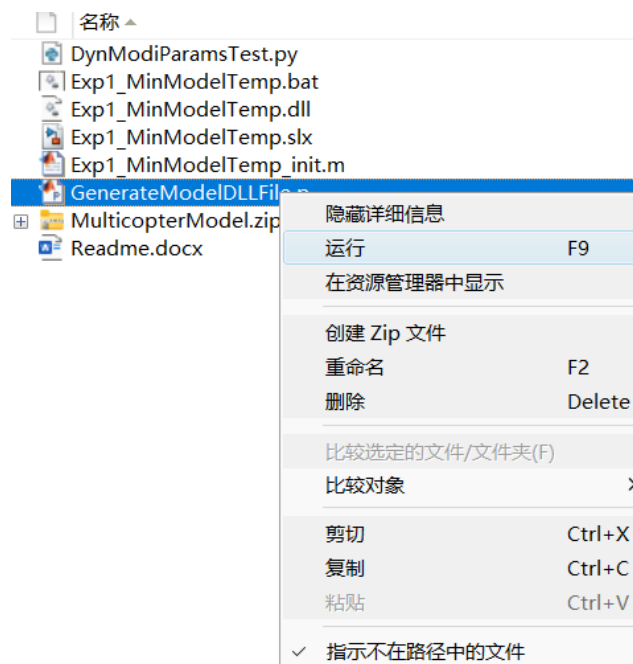
对于 2019b 及之后版本，点击 APPS - CODE GENERATION –Embedded Coder 才能弹出代码生成工具栏，在其中如下图所示点击“C++CODE”-“Generate Code”-“Build”按钮就能编译生成代码。



在 Simulink 的下方点击 View diagnostics 指令，即可弹出诊断对话框，可查看编译过程。
在诊断框中弹出 Build process completed successfully，即表示编译成功。

Step 2: 生成 DLL 文件

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: 启动仿真

运行 Exp1_MaxModelTempSITL.bat,

 Exp1_MinModelTemp.bat	2024/5/11 16:00	Windows 批处理...	6 KB
 Exp1_MinModelTemp.dll	2024/5/11 16:00	应用程序扩展	222 KB
 Exp1_MinModelTemp.slx	2024/5/11 15:59	Simulink Model	65 KB
 Exp1_MinModelTemp_init.m	2024/4/19 16:37	Objective C 源文件	3 KB
 GenerateModelDLLFile.p	2024/4/30 16:04	MATLAB.p.23.2.0	7 KB
 MulticopterModel.zip	2024/5/11 16:00	压缩(zipped)文件...	95 KB
 DynModiParamsTest.py	2024/5/11 16:07	Python 源文件	2 KB
 Readme.docx	2024/5/11 16:16	Microsoft Word ...	3,737 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 DynModiParamsTest.py 文件，输入 python DynModiParamsTest.py，回车运行。

```
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:\PX4PSP\RflySimAPIs\4.RflySimModel\3.CustExps\e0_AdvApiExps\5.ParamAPI\3.DynModiParams>python DynModiParamsTest.py
```

Step 5: 观察结果

可以依次看到：

- 1) 四旋翼初始化在地面。/



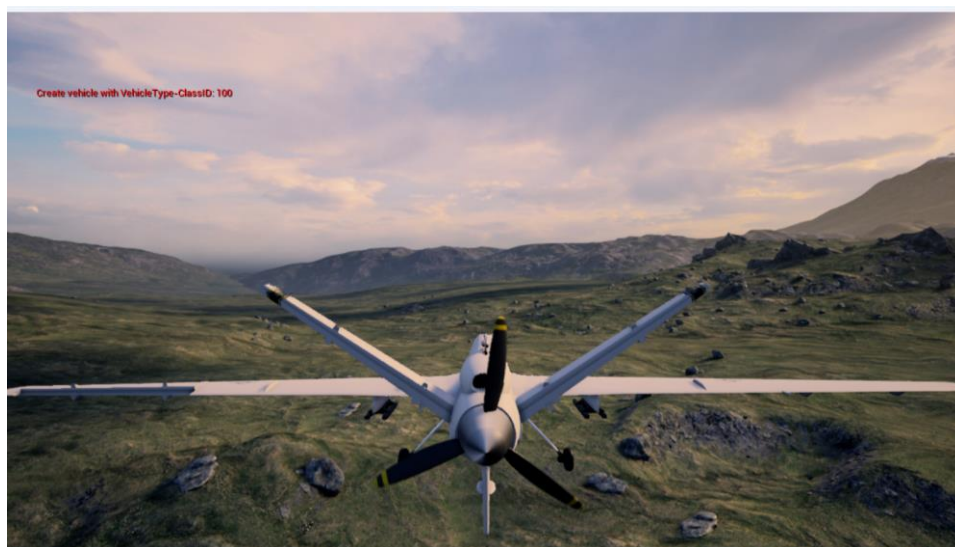
- 2) 四旋翼解锁起飞。



- 3) 四旋翼悬停在空中。



4) 三维显示模型切换为固定翼，同时能在 CopterSim 中看到打印消息。





机架类型
四旋翼

整机质量
1.5 kg

机架轴距
450 mm

飞行海拔
50 m

☐ 品牌型号
☒ 自定义设计



飞思实验室



电机品牌:
DJI (大疆)

螺旋桨品牌:
APC

电调品牌:
Hobbywing (好盈)

电池品牌:
ACE (格氏电池)

型号:
2312 KV960

型号:
10x4.5MR

型号:
XRotor 20A

型号:
LiPo 3S-11.1V-25C-5500mAh

机型数据库: [下拉菜单] [计算] [模型参数] [加入模型库] [删除当前机型]

CopterID: 三轴ClassID:	使用DLL模型文件:	仿真模式:	三维场景地图:	联机	起点位置:	☐ GPS坐标	航向:
1 -1	Exp1_MirModelTemp	PX4_SITL_RFLY	Grasslands	☐ x: 0	y: 0	☐	yaw: 0

飞行选择:	波特率:	通信格式:	[开始仿真]	[停止仿真]	[重新仿真]
[下拉菜单]	921600	Mavlink_Full			

```

1.7.0: RCLOS ALARM
PX4: Armed by external command
PX4: Command SET_MODE ACCEPTED
PX4: Command SET_MODE ACCEPTED
PX4: Enter Offboard Mode!
PX4: Command ARM/DISARM ACCEPTED
PX4: Command ARM/DISARM ACCEPTED
PX4: Takeoff detected
CopterSia: DynModlParans MSG Received!27.0.0.1
Bitmask:1, data[0~4]: 100.00 0.00 0.00 0.00 0.00
PX4: ROTORS DISARMED
PX4: Command ARM/DISARM ACCEPTED
        
```

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 DynModiParamsTest.py 时，可使用 VS Code（或 Pycharm 等工具）来打开 DynModiParamsTest.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 DynModiParamsTest.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。

```
mav1.SendPosNED(0,0,-10)
time.sleep(0.5)
print("开始起飞")
mav1.SendMavArm(True) # Arm the drone

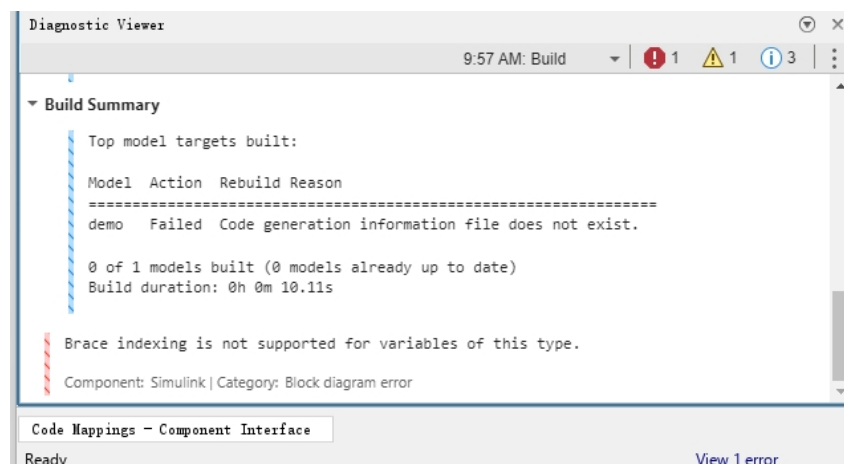
if time.time() - startTime > 20 and flag==1:
    #np.zeros()
    dll.sendDynModiParams(0b1,[100])
    print('修改三维显示模型为固定翼')
    flag=2
```

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