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。

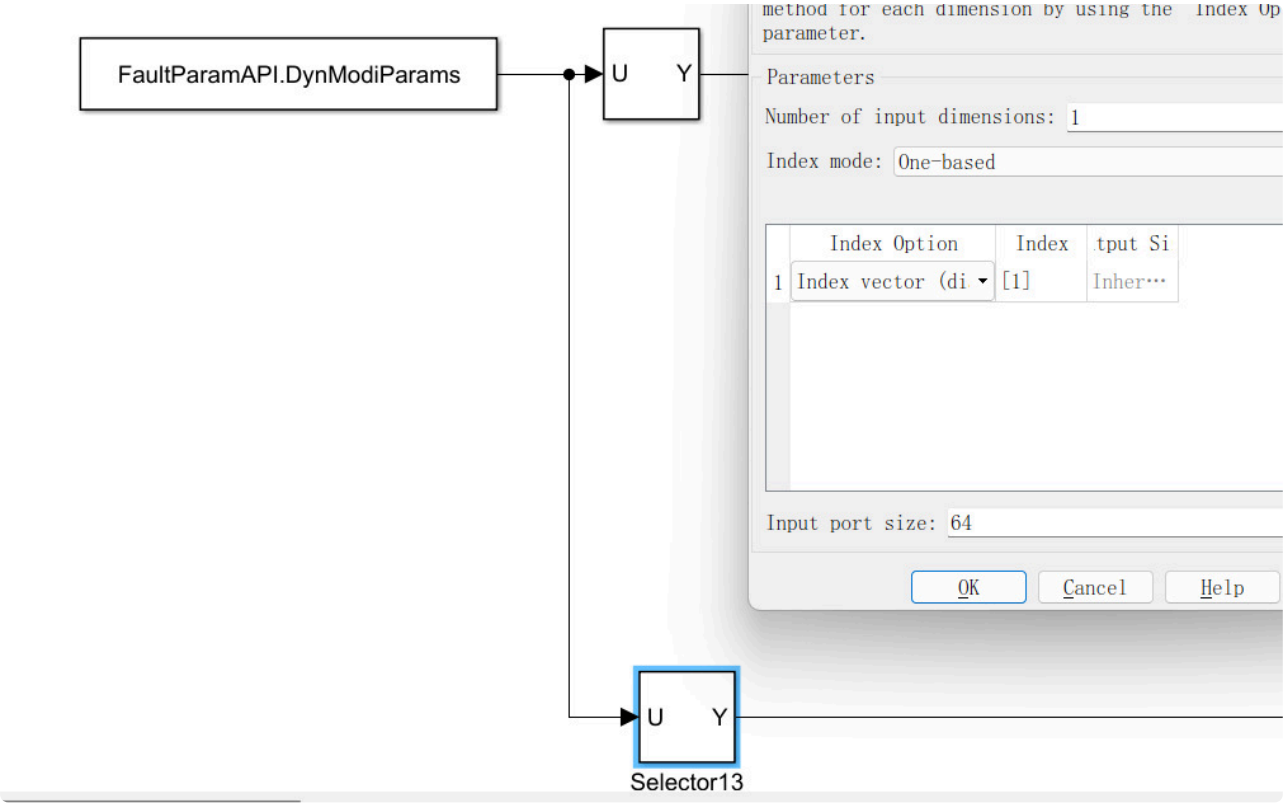
1.3关键知识点

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\e6.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。

```python
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
..\Readme.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

4.1 软件要求

Windows 10及以上版本；RflySim工具链；MATLAB 2017B及以上；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；\ \台。

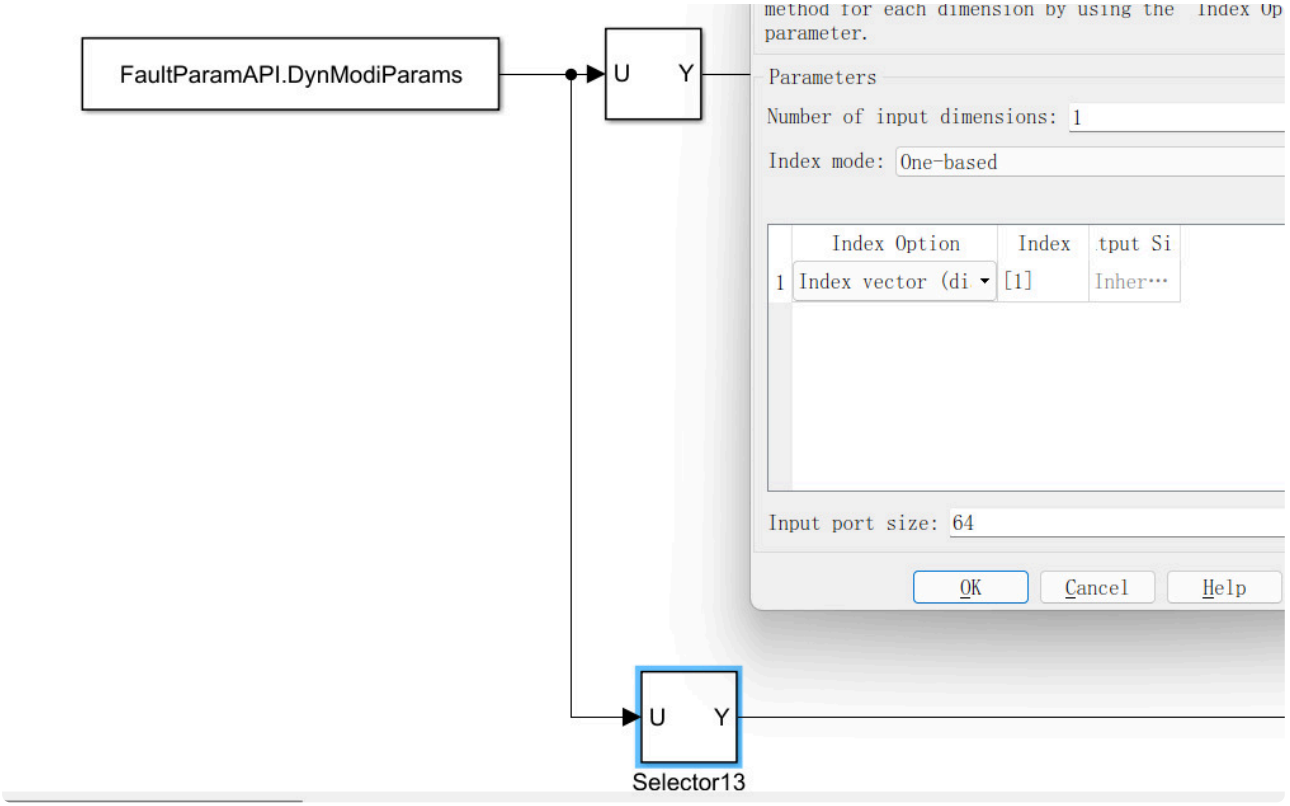
> ①：推荐配置请见：https://rflysim.com/

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

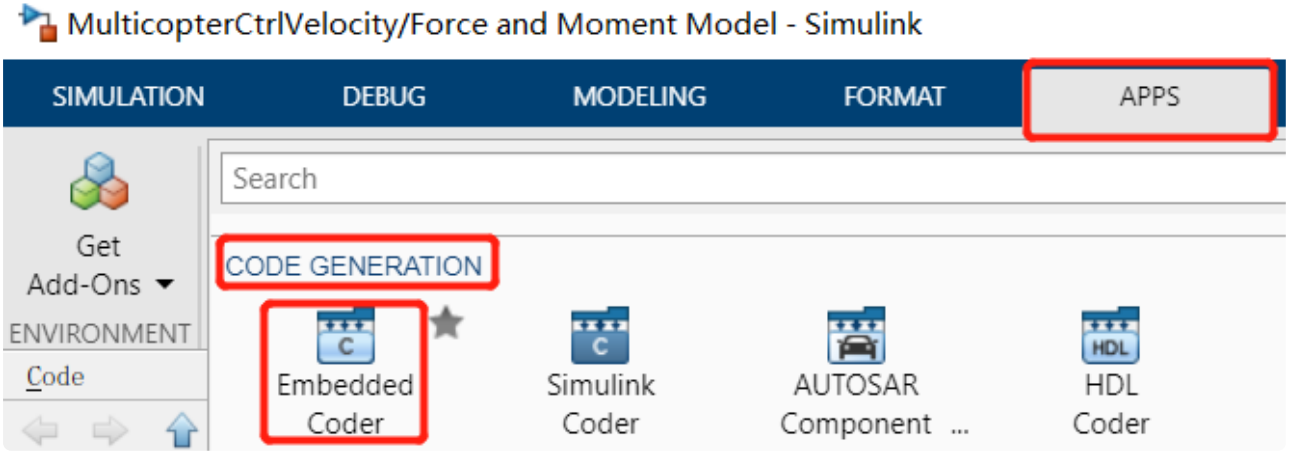
将3DOutput模块的输入由ModelParam_3DType改为FaultParamAPI.DynModiParams(1)。

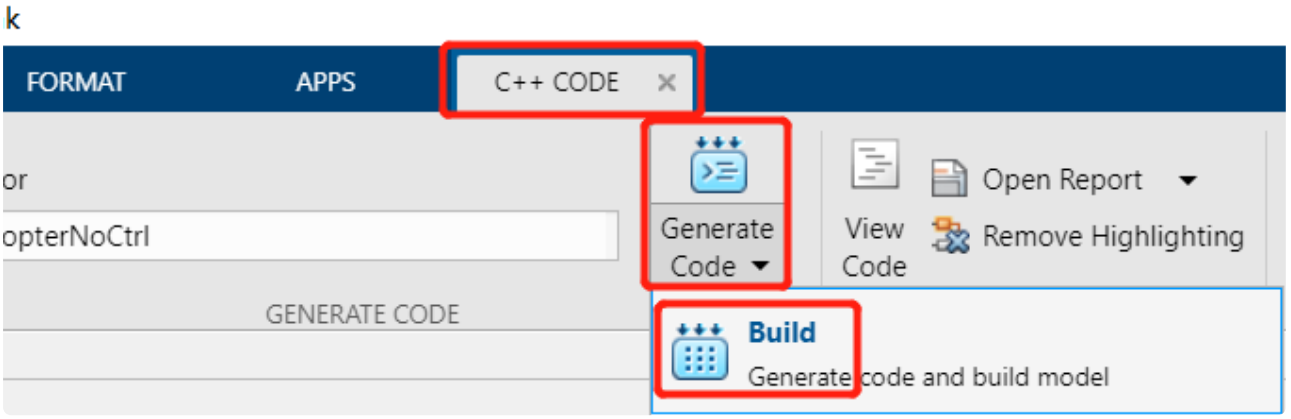


修改完成后，保存，并在Simulink中点击编译命令。

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。

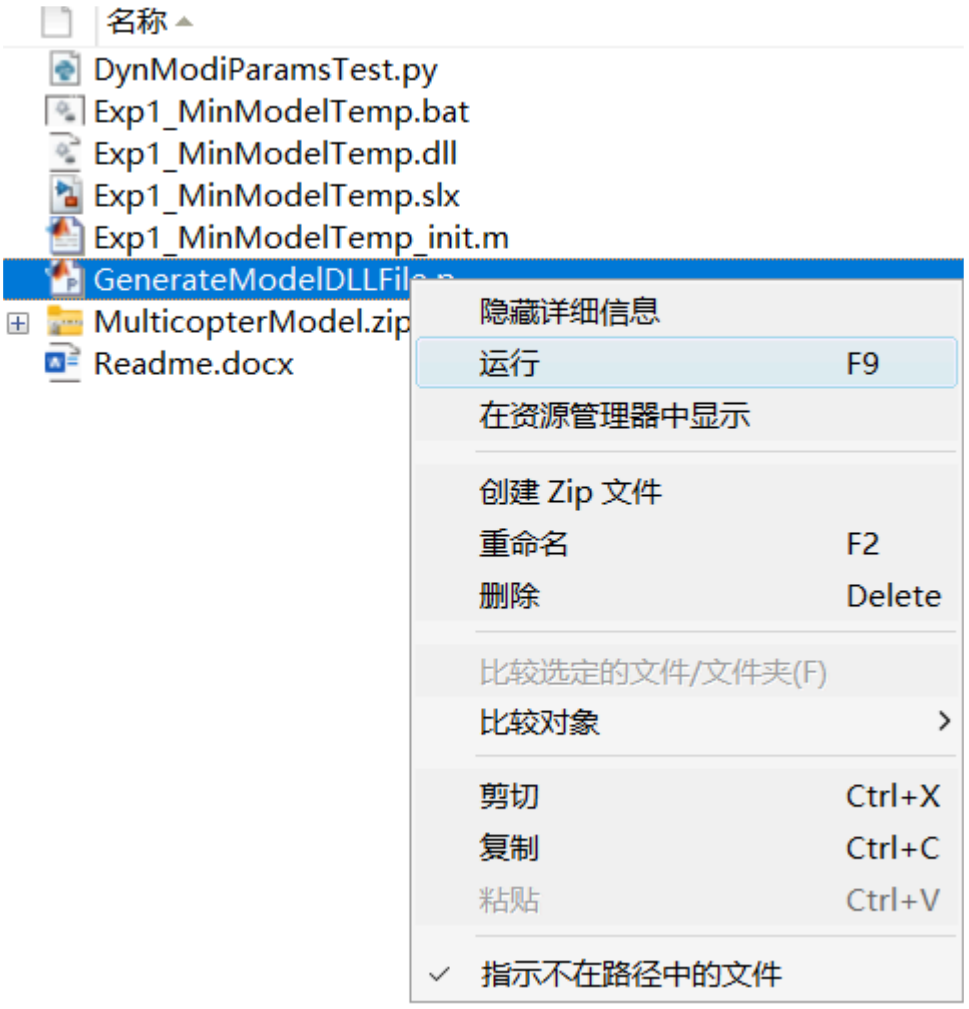




在 Simulink 的下方点击 View diagnostics
指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出 Build process completed successfully，即表示编译成功。

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。

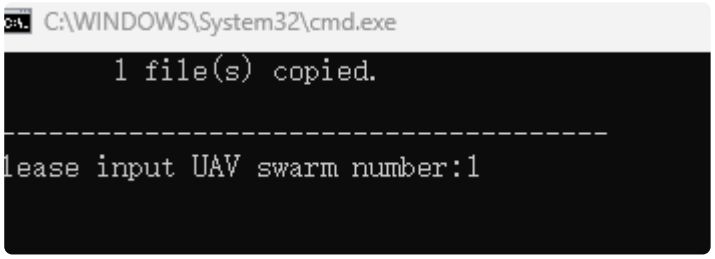


Step 3：启动仿真

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

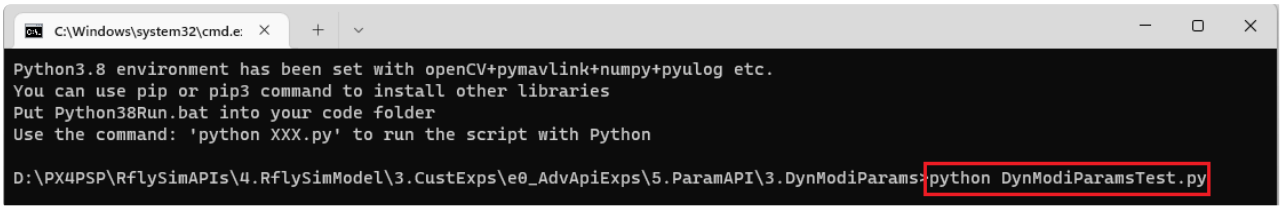
输入1，启动1架无人机的软件在环仿真。



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。



Step 5：观察结果

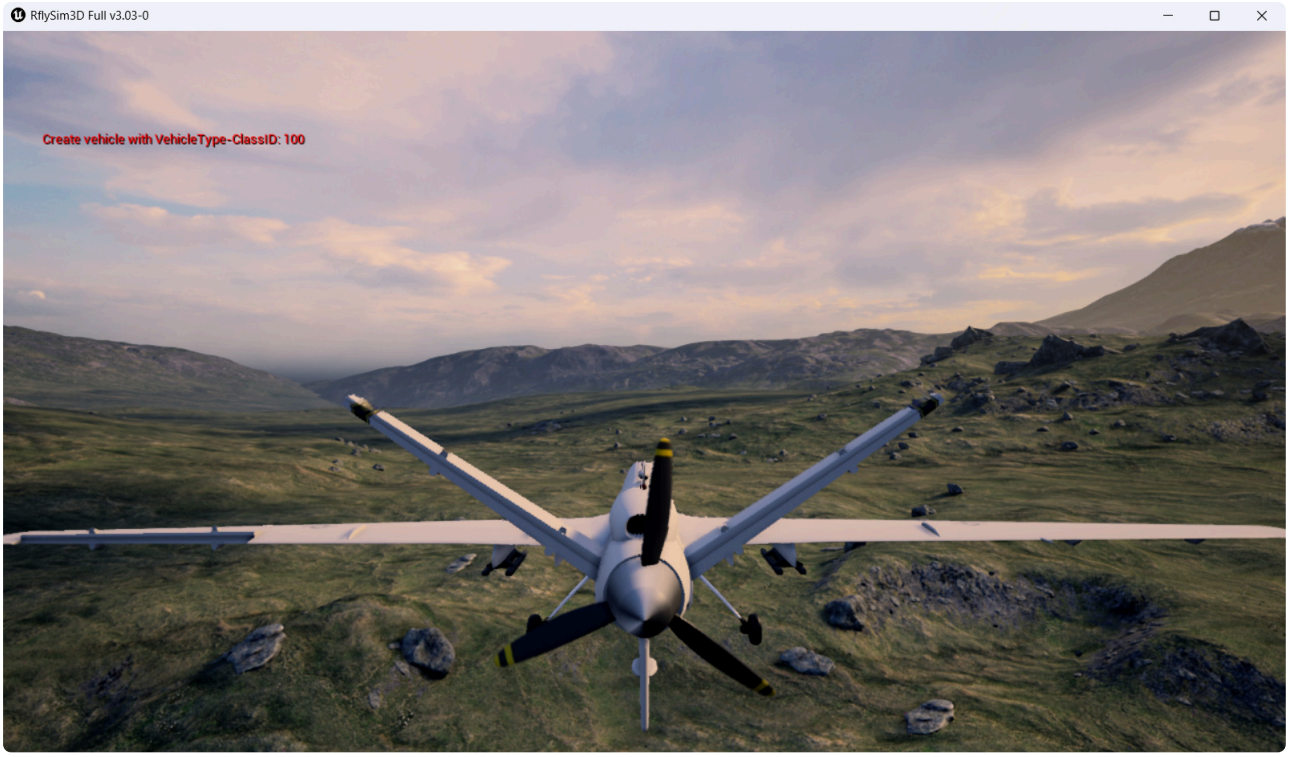
可以依次看到：

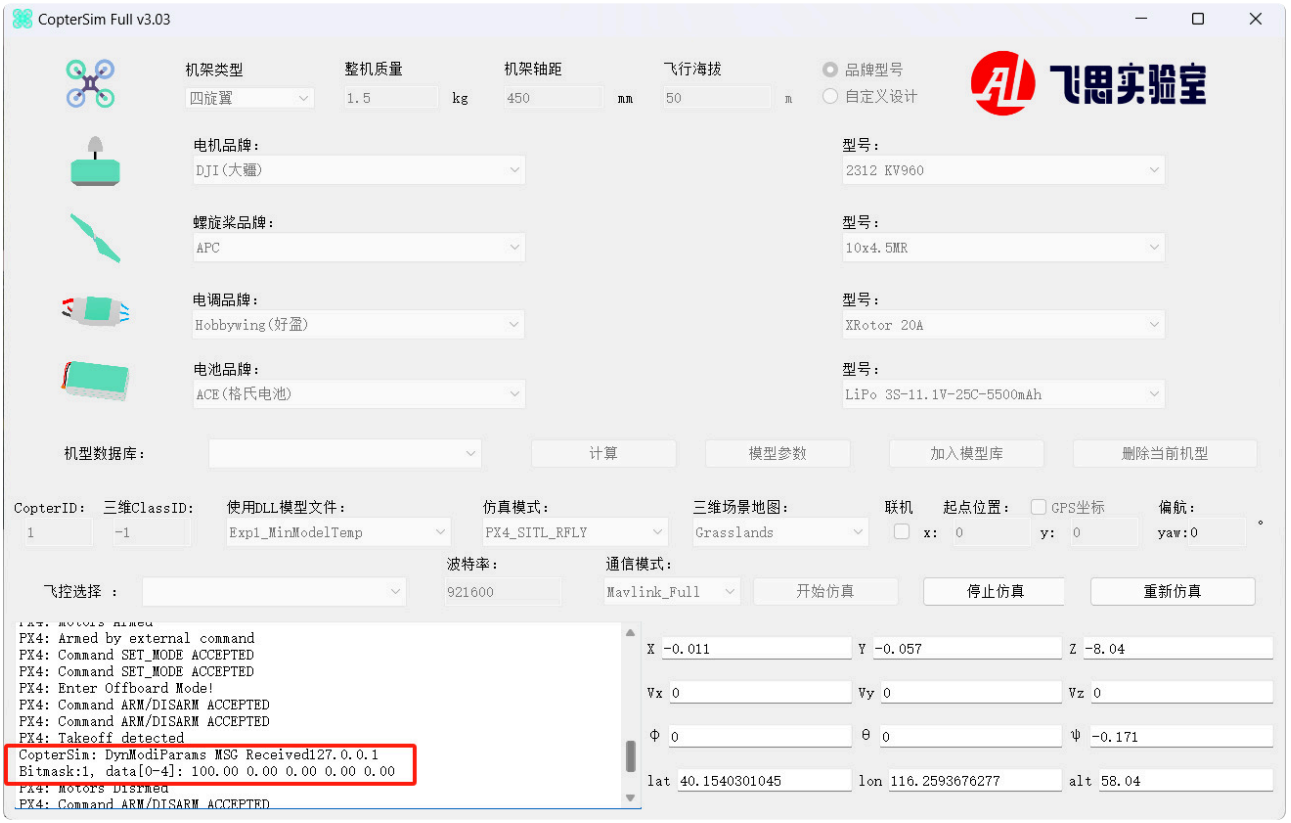
1）四旋翼初始化在地面。/

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。





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\e3.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。

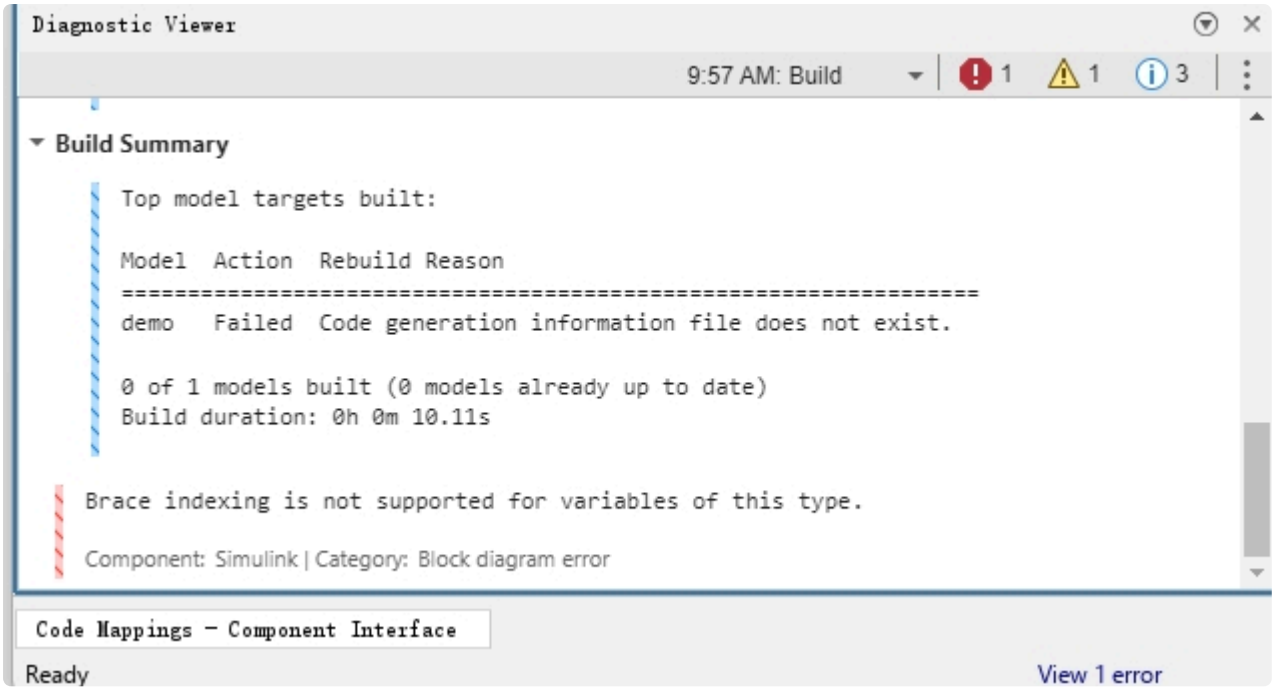
```
    mav1.SendPosNED(0,0,-10)
    time.sleep(0.5)
    print("开始起飞")
    mav1.SendMavArm(True) # Arm the drone


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

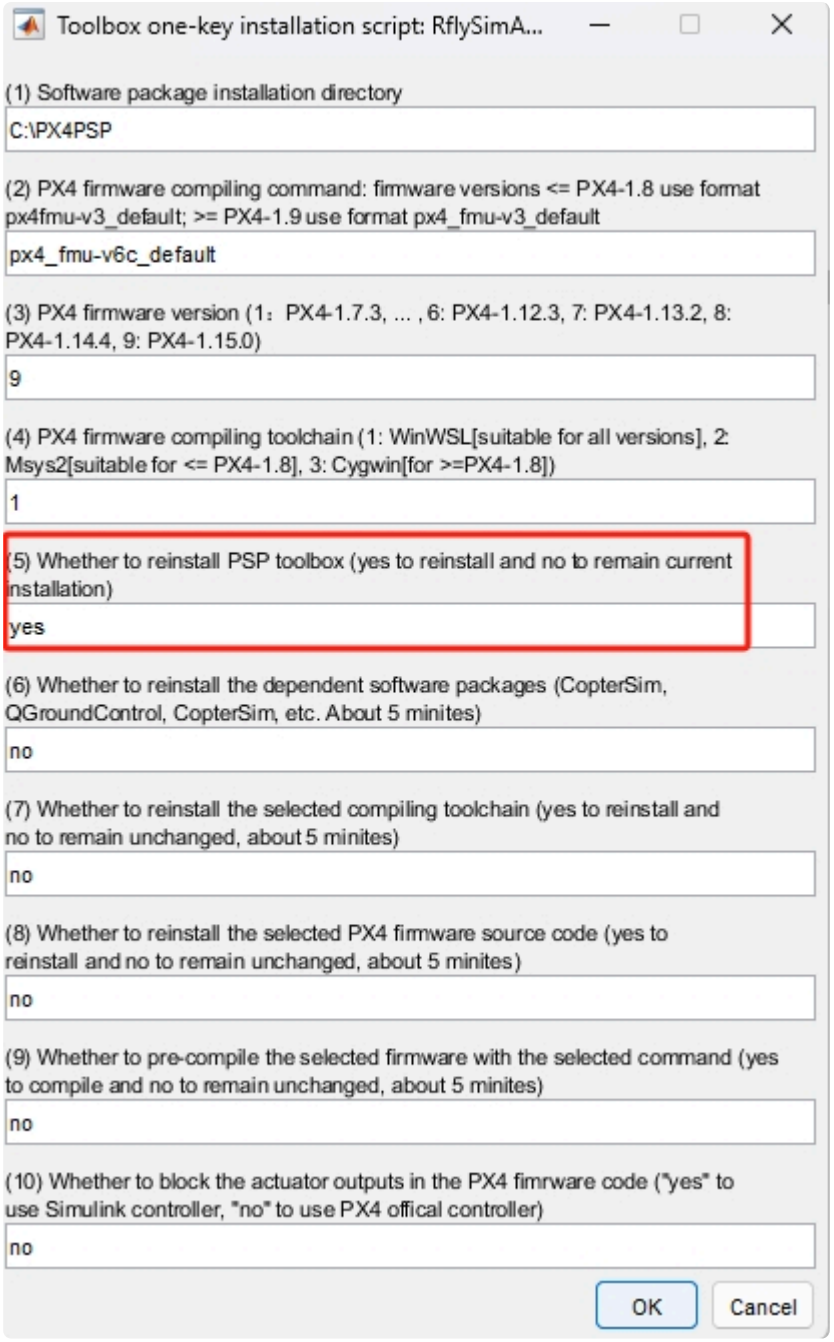
Q1：未正确安装visual studio c++编译环境并配置mex，导致Simulink文件编译失败



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 fimrware code ("yes" to use Simulink controller, "no" to use PX4 offical controller)
no
OK
Cancel