

1. 实验名称及目的

1.1 实验名称

dll模型模型inSILInts和inSILFloats接口调用实验 (python)

1.2 实验目的

RflySim平台提供了丰富的模型输入输出接口，以实现复杂功能。通过该例程介绍模型inSILInts与inSILFloats输入接口Python的使用方法。

1.3 关键知识点

1.3.1 接口数据解析

结构体定义如下：

```
struct PX4SILIntFloat{  
  
int checksum;//1234567897  
  
int CopterID;//飞机的ID  
  
int inSILInts[8];//int标志位  
  
float inSILFloats[20];//float参数位  
  
};
```

1.3.2 输入接口的Python使用方法

关键接口函数sendSILIntFloat

接口使用实例解析：

本例程中的Python文件中的关键代码如下——

导入必需的库

```
import numpy as np
```

```
from DllSimCtrlAPI import DllSimCtrlAPI
```

numpy：用于创建和操作数值数组，常用于科学计算。

DllSimCtrlAPI：从DllSimCtrlAPI模块导入的DllSimCtrlAPI类，这个类包含用于传输外部数据给dll模型的多个方法。

初始化DLL控制接口

```
dll1 = DllSimCtrlAPI()
```

创建DllSimCtrlAPI类的一个实例，命名为dll1。

发送inSILInts整数数据

```
silInt = np.zeros(8).astype(int).tolist()
```

```
silInt[0] = 5
```

```
dll1.sendSILIntFloat(silInt)
```

使用numpy创建一个长度为8的整数数组 silInt，所有元素初始值为0。

将silInt数组的第一个元素设置为5

调用dll1.sendSILIntFloat(silInt)方法发送这个整数数组。由于未指定浮点数组，不起实际作用。

发送浮点数据

```
silFloat = np.zeros(20).astype(float).tolist()
```

```
silFloat[0] = 5
```

```
dll1.sendSILIntFloat(silInt,silFloat)
```

使用numpy创建一个长度为20的浮点数组silFloat，所有元素初始值为0。

将silFloat数组的第一个元素设置为5。

调用dll1.sendSILIntFloat(silInt, silFloat)方法发送整数数组和浮点数组。这次调用与前一次的区别在于同时指定了浮点数据。

2. 实验效果

四旋翼模型基础上通过inSILInts和inSILFloats输入接口连接载具显示输出接口，通过python外部控制的方式动态修改3D显示参数“3DType”，实现软件在环仿真过程中显示机型的实时变换。

3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\4.RflySimModel\0.ApiExps\11.inSILAPI\1.InSILIntsFloats\3.inSILIntsFloats_py

文件夹/文件名称	说明
..\Readme.pdf	inSILIntsFloats外部数据输入接口实验原理
MulticopterNoCtrl.slx	四旋翼飞机模型文件。
MulticopterNoCtrl_SITLRun.bat	软件在环仿真批处理文件。
GenerateModelDLLFile.p	DLL格式转化文件。
MulticopterNoCtrl_init.m	动力学模型相关参数。
MavLinkStruct.mat	MavLink数据结构体mat文件
inSIL.py	inSILInts和inSILFloats接口发送文件
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；MATLAB 2017b及以上^③。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4_fmuv6x_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。

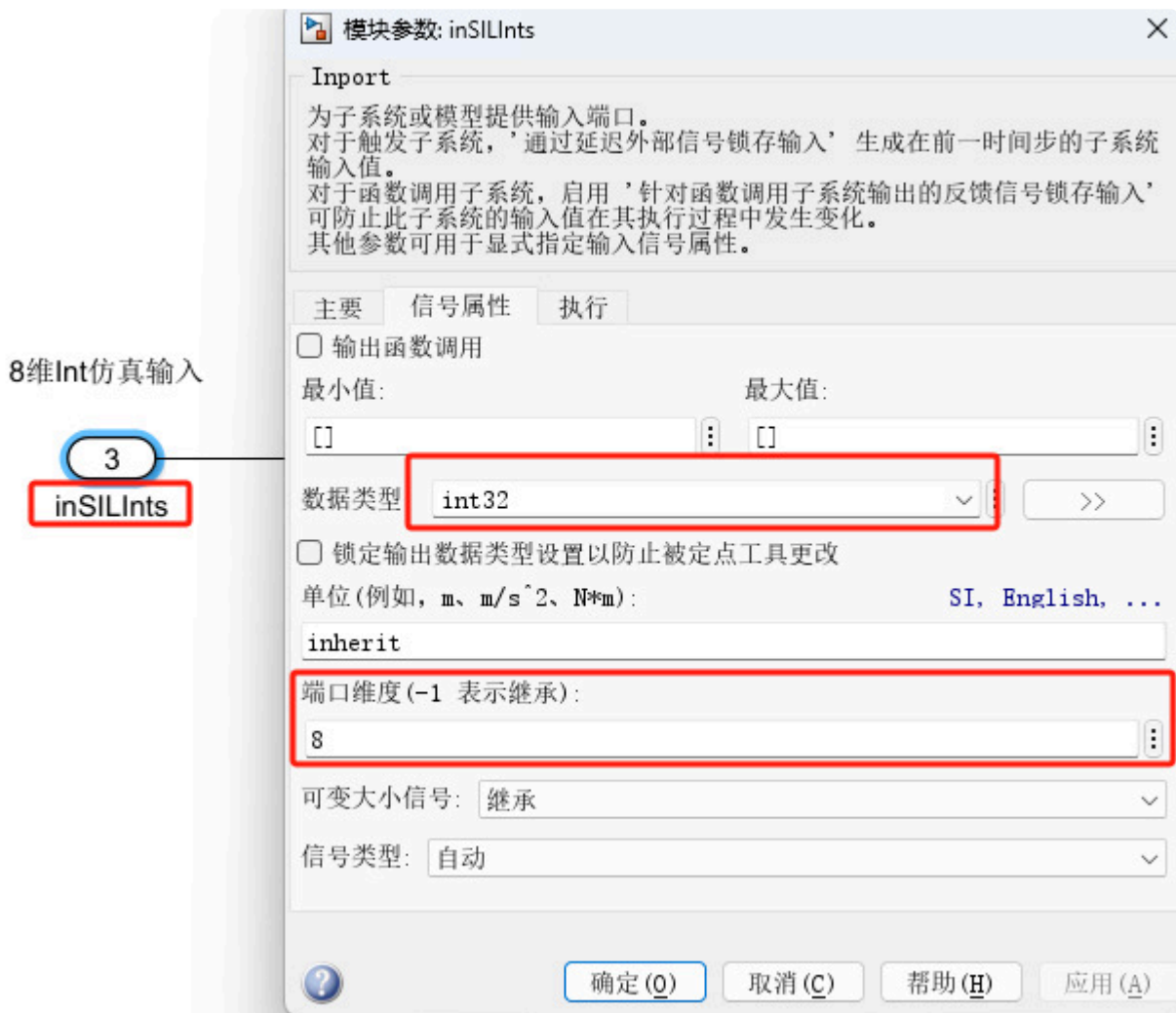
①：推荐配置请见：<https://rflysim.com/>

5. 实验步骤

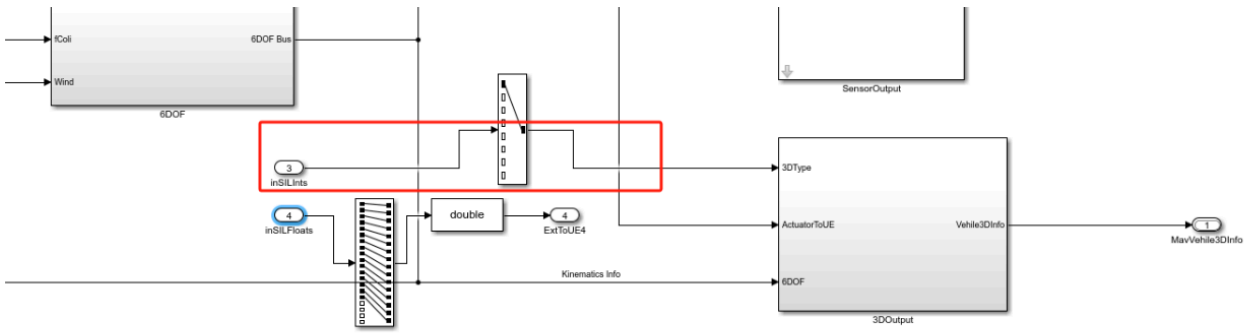
5.1 必做实验：inSILInts接口使用

Step 1：增加inSILInts输入接口

打开“MulticopterNoCtrl.slx”文件，然后新增inSILInts输入接口，在模型中增加该接口后需要严格按照既定协议命名，并且需要对数据类型与接口维度进行对应的修改。



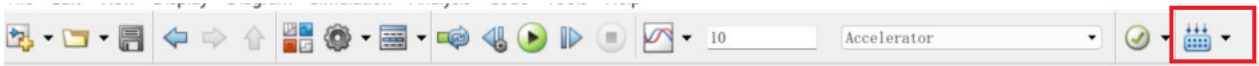
该接口为8维的int型数据接口，选取8维输入中的第一维作为UE的显示ID与3DType连接。



注意，inSILInts与inSILFloats作为simulink的输入接口要一并接通，生成的DLL模型才能收到coptersim转发的数据。

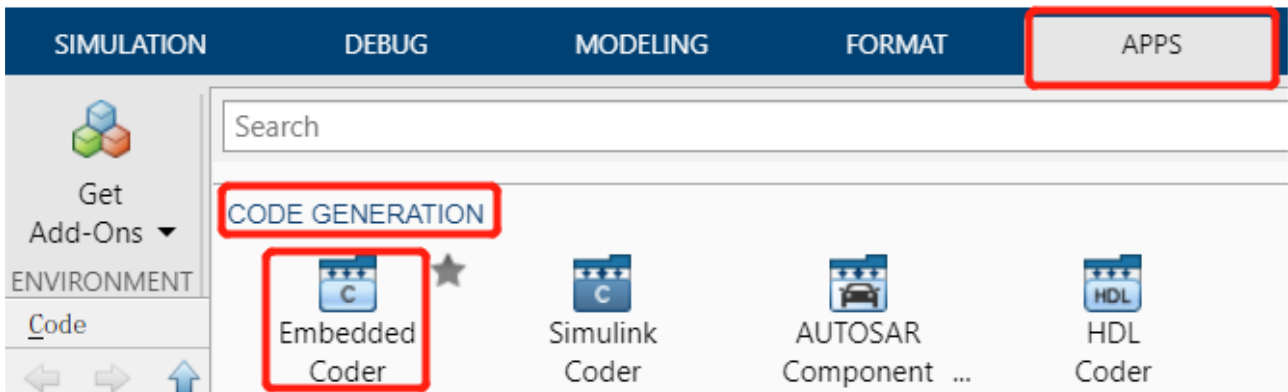
Step 2: 编译模型

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。

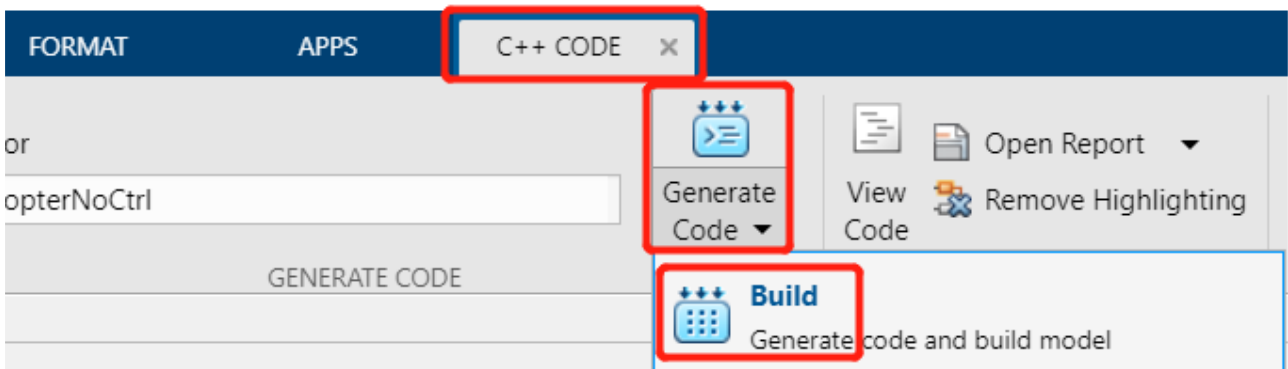


对于2019b及之后版本，点击APPS - CODE GENERATION - Embedded Coder才能弹出代码生成工具栏，在其中如下图所示点击“C++CODE” - “Generate Code” - “Build”按钮就能编译生成代码。

MulticopterCtrlVelocity/Force and Moment Model - Simulink

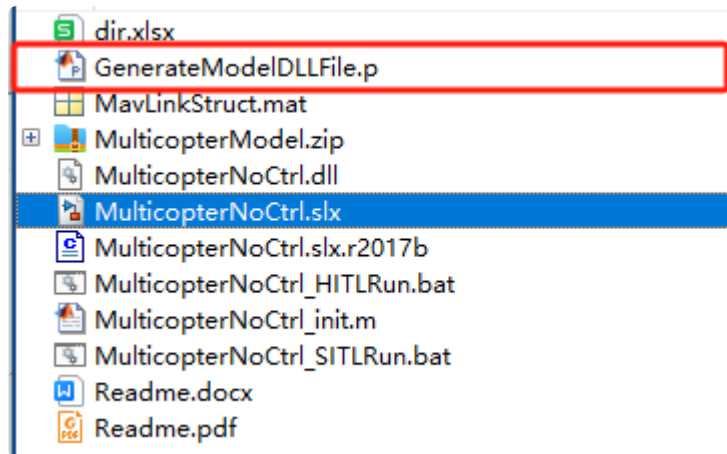


k



Step 3: 生成DLL文件

右键运行“GenerateModelDLLFile.p”文件，将上一步的代码生成动态链接库dll文件。



Step 4: 启动仿真

右键以管理员身份运行“MulticopterNoCtrl_SITLRun.bat”批处理文件，开启软件在环仿真。

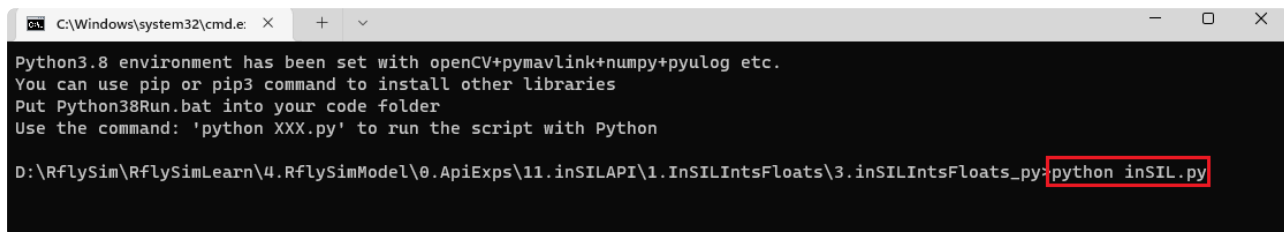
名称	修改日期	类型	大小
dir.xlsx	2024/1/17 10:04	XLSX 工作表	6 KB
GenerateModelDLLFile.p	2024/1/17 10:04	MATLAB.p.9.14.0	6 KB
MavLinkStruct.mat	2024/1/17 10:04	MATLAB.mat.9.1...	5 KB
MulticopterModel.zip	2024/1/17 10:04	360压缩 ZIP 文件	100 KB
MulticopterNoCtrl.dll	2024/1/17 10:04	应用程序扩展	226 KB
MulticopterNoCtrl.slx	2024/1/19 15:48	Simulink Model	135 KB
MulticopterNoCtrl.slx.r2017b	2024/1/17 15:01	R2017B 文件	75 KB
MulticopterNoCtrl_HITLRun.bat	2024/1/17 10:04	Windows 批处理...	6 KB
MulticopterNoCtrl_init.m	2024/1/17 15:00	MATLAB Code	3 KB
MulticopterNoCtrl_SITLRun.bat	2024/1/17 10:04	Windows 批处理...	6 KB
Readme.docx	2024/1/17 16:48	DOCX 文档	7,242 KB
Readme.pdf	2024/1/17 16:53	Foxit PDF Reade...	3,174 KB

Step 5: 运行Python程序

使用记事本或者vscode打开inSIL.py文件将sillnt[0]分别修改为5、8、10、50。修改完成后，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

inSIL.py文件，输入
python inSIL.py

```
# inSILInts接口发送
silInt=np.zeros(8).astype(int).tolist()
silInt[0]=5
dll1.sendSILIntFloat(silInt)
print('发送完成')
```



Step 6: 观察结果

根据机型定义文档可知，在UE中可以看到，在修改过后三维显示模型分别转变为六旋翼、八旋翼、四轴八旋翼和小车，由此也证明了inSILInts输入接口通过动态的修改3D显示参数实现了仿真中的显示模型变换。



Create vehicle with VehicleType-ClassID: 8



Create vehicle with VehicleType-ClassID: 10





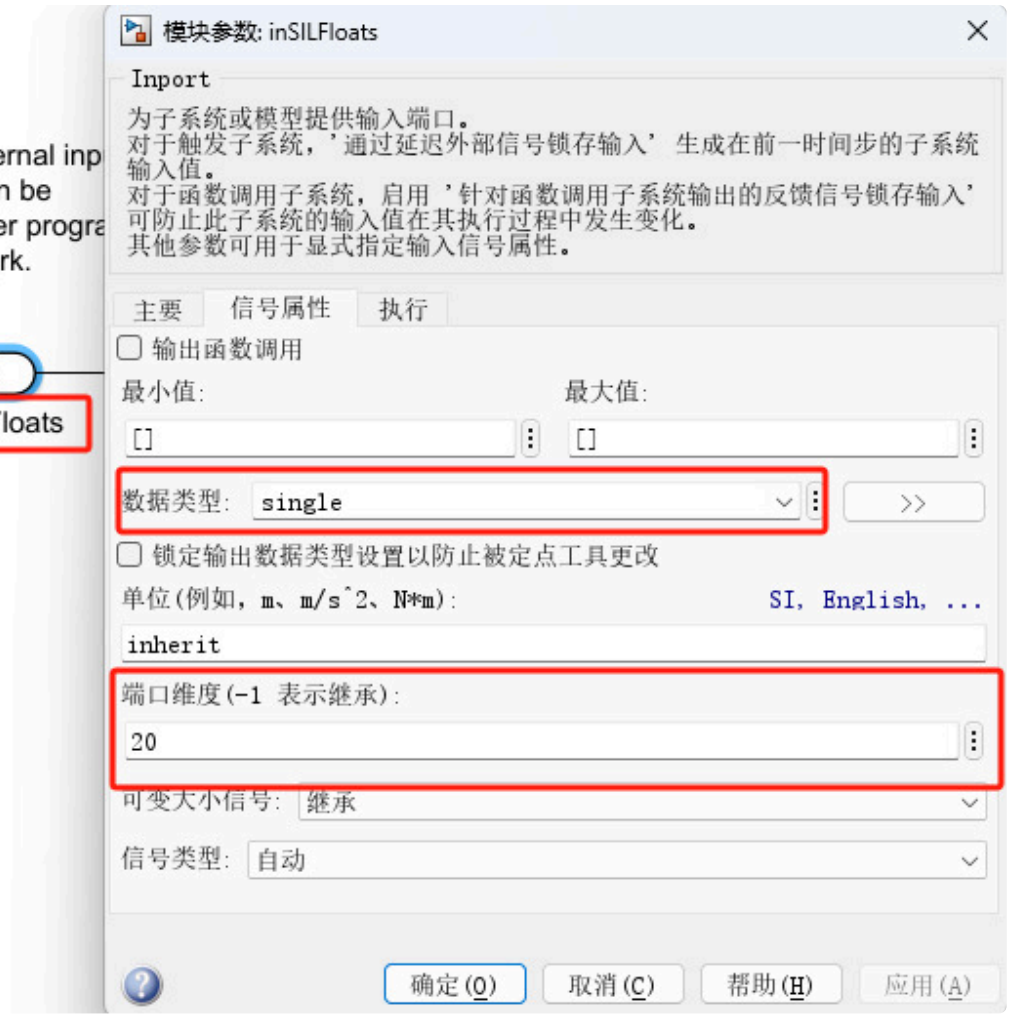
5.2 必做实验：inSILFloats接口使用

Step 1: 增加inSILFloats输入接口

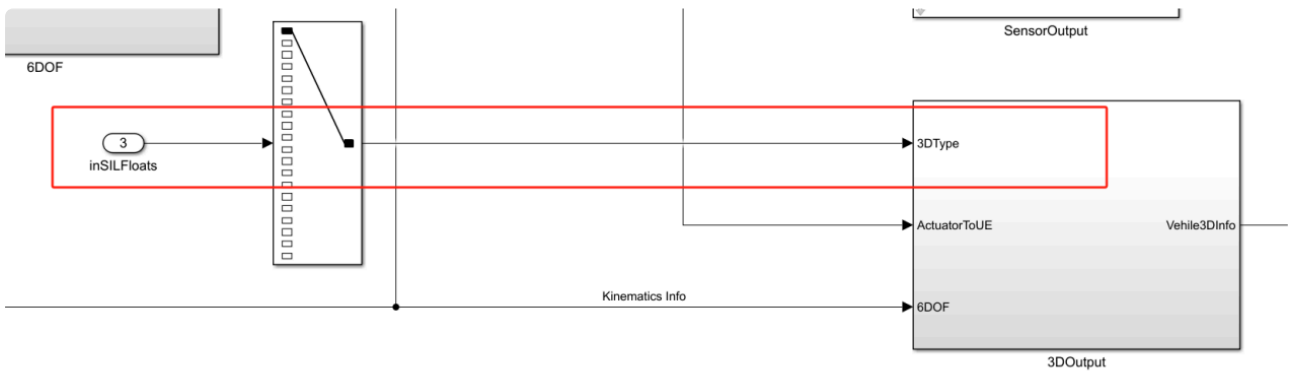
打开“MulticopterNoCtrl.slx”文件，然后新增inSILFloats输入接口，在模型中增加该接口后需要严格按照既定协议命名，并且需要对数据类型与接口维度进行对应的修改。

20-dimensional external input float signals that can be transferred from other programs through UDP network.

4
inSILFloats



该接口为20维的浮点型数据接口，选取20维输入中的第一维作为UE的显示ID与3DType连接。

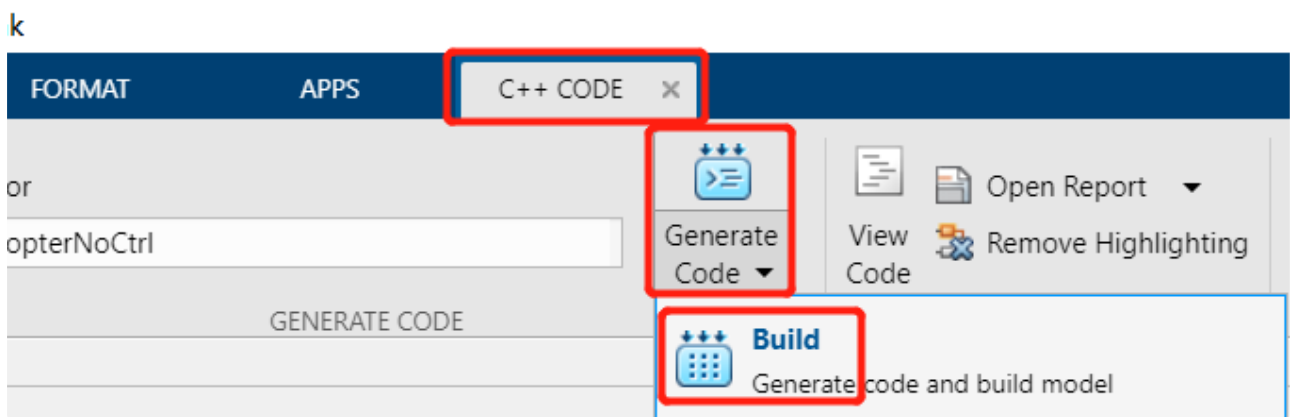
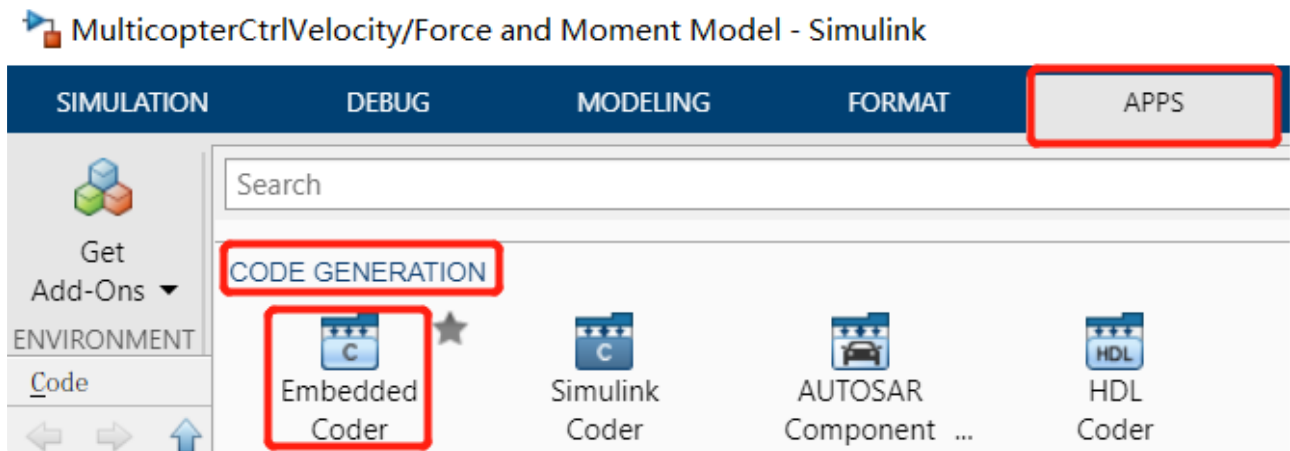


Step 2: 编译模型

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。

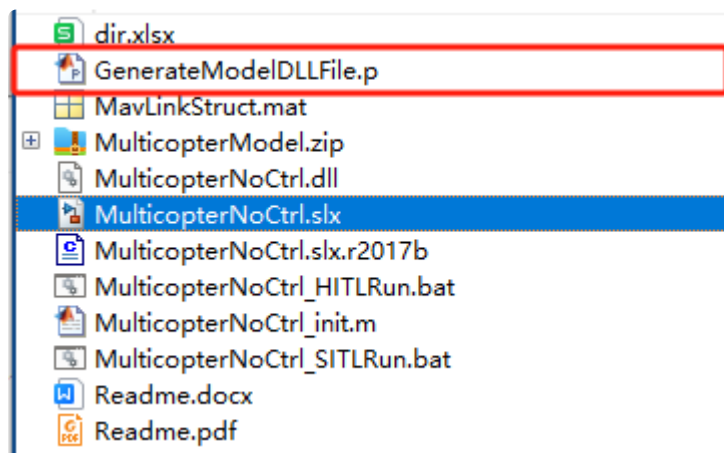


对于2019b及之后版本，点击APPS - CODE GENERATION -Embedded Coder才能弹出代码生成工具栏，在其中如下图所示点击“C++CODE” - “Generate Code” - “Build” 按钮就能编译生成代码。



Step 3: 生成DLL文件

右键运行“GenerateModelDLLFile.p”文件，将上一步的代码生成动态链接库dll文件。



Step 4: 启动仿真

右键以管理员身份运行“MulticopterNoCtrl_SITLRun.bat”批处理文件，开启软件在环仿真。

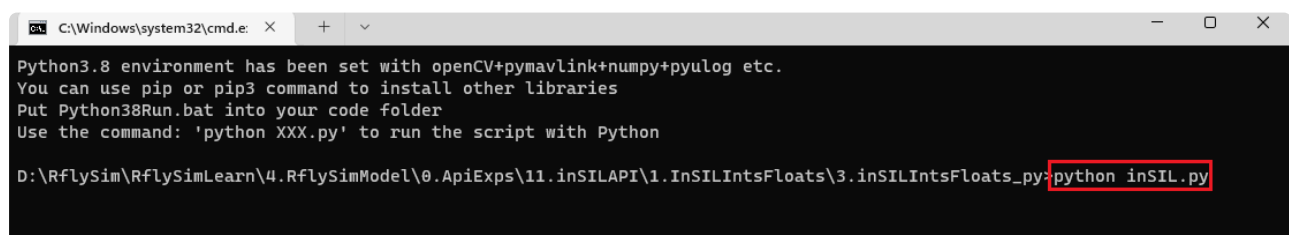
名称	修改日期	类型	大小
 dir.xlsx	2024/1/17 10:04	XLSX 工作表	6 KB
 GenerateModelDLLFile.p	2024/1/17 10:04	MATLAB.p.9.14.0	6 KB
 MavLinkStruct.mat	2024/1/17 10:04	MATLAB.mat.9.1...	5 KB
 MulticopterModel.zip	2024/1/17 10:04	360压缩 ZIP 文件	100 KB
 MulticopterNoCtrl.dll	2024/1/17 10:04	应用程序扩展	226 KB
 MulticopterNoCtrl.slx	2024/1/19 15:48	Simulink Model	135 KB
 MulticopterNoCtrl.slx.r2017b	2024/1/17 15:01	R2017B 文件	75 KB
 MulticopterNoCtrl_HITLRun.bat	2024/1/17 10:04	Windows 批处理...	6 KB
 MulticopterNoCtrl_init.m	2024/1/17 15:00	MATLAB Code	3 KB
 MulticopterNoCtrl_SITLRun.bat	2024/1/17 10:04	Windows 批处理...	6 KB
 Readme.docx	2024/1/17 16:48	DOCX 文档	7,242 KB
 Readme.pdf	2024/1/17 16:53	Foxit PDF Reade...	3,174 KB

Step 5: 运行Python程序

使用记事本或者vscode打开 [inSIL.py](#) 文件将 `silFloat[0]` 分别修改为5、8、10、50。修改完成后，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 [inSIL.py](#) 文件，输入

```
python inSIL.py
```

```
# inSILFloats接口发送
silFloat=np.zeros(20).astype(float).tolist()
silFloat[0]=8
dll1.sendSILIntFloat(silFloat)
print('发送完成')
```



```
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\0.ApiExps\11.inSILAPI\1.InSILIntsFloats\3.inSILIntsFloats_py>python inSIL.py
```

Step 6: 观察结果

根据机型定义文档可知，在UE中可以看到，在修改过后三维显示模型分别转变为六旋翼、八旋翼、四轴八旋翼和小车，由此也证明了 `inSILFloats` 输入接口通过动态的修改3D显示参数实现了仿真中的显示模型变换。





5.3 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 `inSIL.py` 时，可使用VS Code（或Pycharm等工具）来打开 `inSIL.py` 文件，并阅读代码，修改代码，调试执行等。

I 扩展实验

- 请自行使用VS Code阅读 `inSIL.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。

```
# import required libraries
import numpy as np

# import RflySim APIs
# import PX4MavCtrlV4 as PX4MavCtrl
from DllSimCtrlAPI import DllSimCtrlAPI

dll1 = DllSimCtrlAPI()

# inSILInts接口发送
silInt = np.zeros(8).astype(int).tolist()
silInt[0] = 5
dll1.sendSILIntFloat(silInt)
print("发送完成")

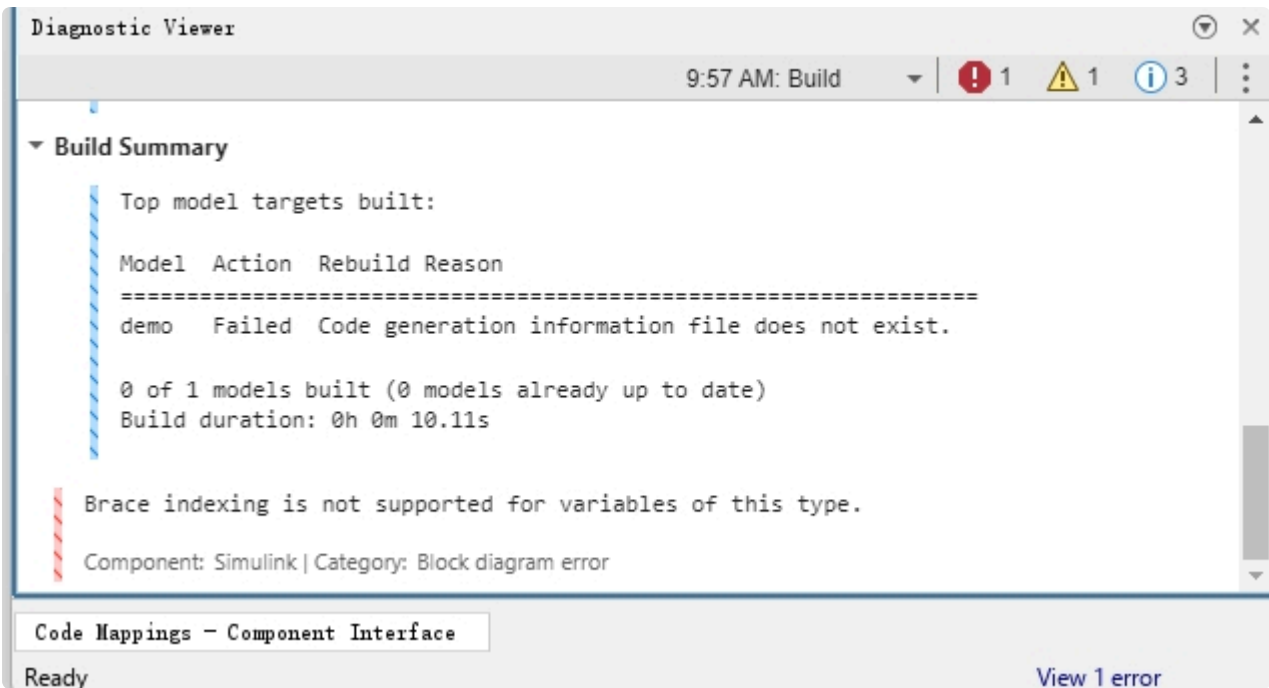
# inSILFloats接口发送
silFloat = np.zeros(20).astype(float).tolist()
silFloat[0] = 5
dll1.sendSILIntFloat(silInt, silFloat)
print("发送完成")
```

I 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](#)

I 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