

FSJ150四旋翼系统辨识模型DLL生成及SIL/HIL实验（仅限完整版及以上版本）

1. 实验目的

在Matlab将Simulink文件编译生成FSJ150辨识模型的DLL模型文件；并对生成的FSJ150辨识模型进行软硬件在环仿真测试，通过本例程熟悉平台FSJ150辨识模型的使用。

2. 实验要求

2.1 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_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>

2.2 硬件要求

笔记本/台式电脑^① 1台；Pixhawk 6X或其它飞控^② 1台；数据线 1台。

①：推荐配置请见：<https://rflysim.com/>

3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\4.RflySimModel\3.CustExps\6_IdentificationModel\FSJ150](#)

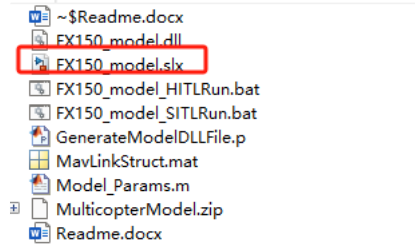
文件夹/文件名称	说明
FSJ150_H_model.slx	四旋翼飞机模型文件。
FSJ150_model_HITLRun.bat	硬件在环仿真批处理文件。
FSJ150_model_SITLRun.bat	软件在环仿真批处理文件。
GenerateModelDLLFile.p	DLL格式转化文件。
FSJ150_H_Model_init.m	动力学模型相关参数。
MavLinkStruct.mat	MavLink数据结构体mat文件

4. 实验内容或步骤

4.1. 必做实验：DLL模型生成

Step 1: 编译模型

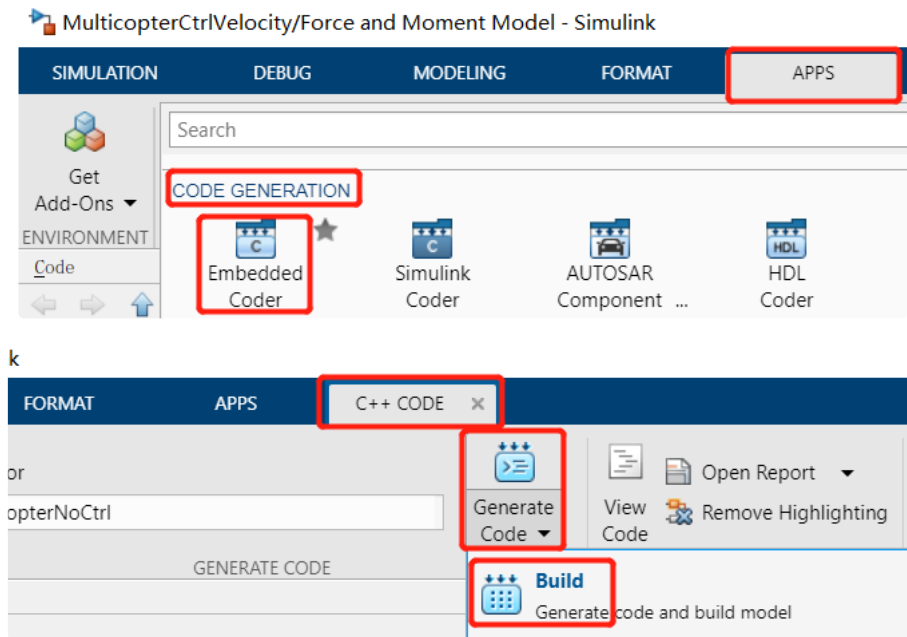
在Matlab中打开“FSJ150_model.slx” Simulink 文件，点击Build Model 按钮生成代码。编译配置可参考 [4.RflySimModel\0.ApiExps\2.UserDefinedC++\2.GenC++\Readme.pdf](#)



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。

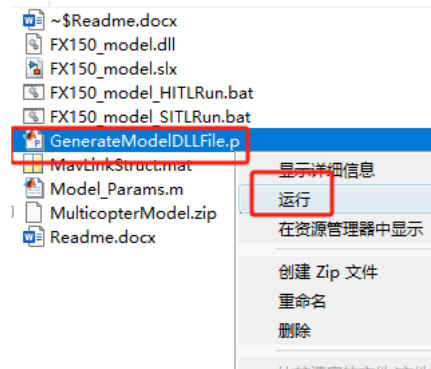


对于2019b及之后版本，点击APPS - CODE GENERATION - Embedded Coder才能弹出代码生成工具栏，在其中如下图所示点击“C++CODE” - “Generate Code” - “Build”按钮就能编译生成代码。



Step 2: 生成DLL文件

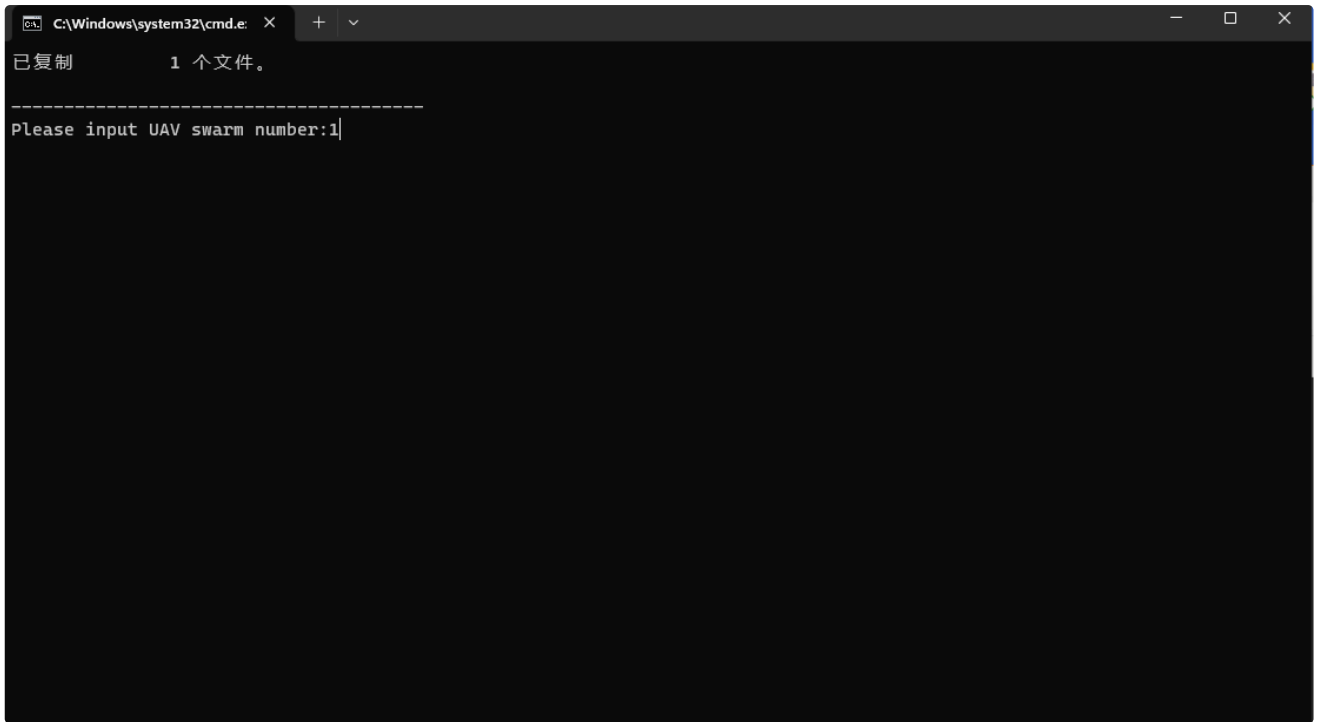
模型编译完成后，在 matlab 中右键“GenerateModelDLLFile.p”文件，点击运行，生成 DLL 文件。



4.2. 必做实验：软件在环仿真

Step 1: 启动仿真

双击运行“FSJ150_model_SITLRun.bat”批处理文件，在弹出的终端窗口中输入 1，启动一架飞机的软件在环仿真。

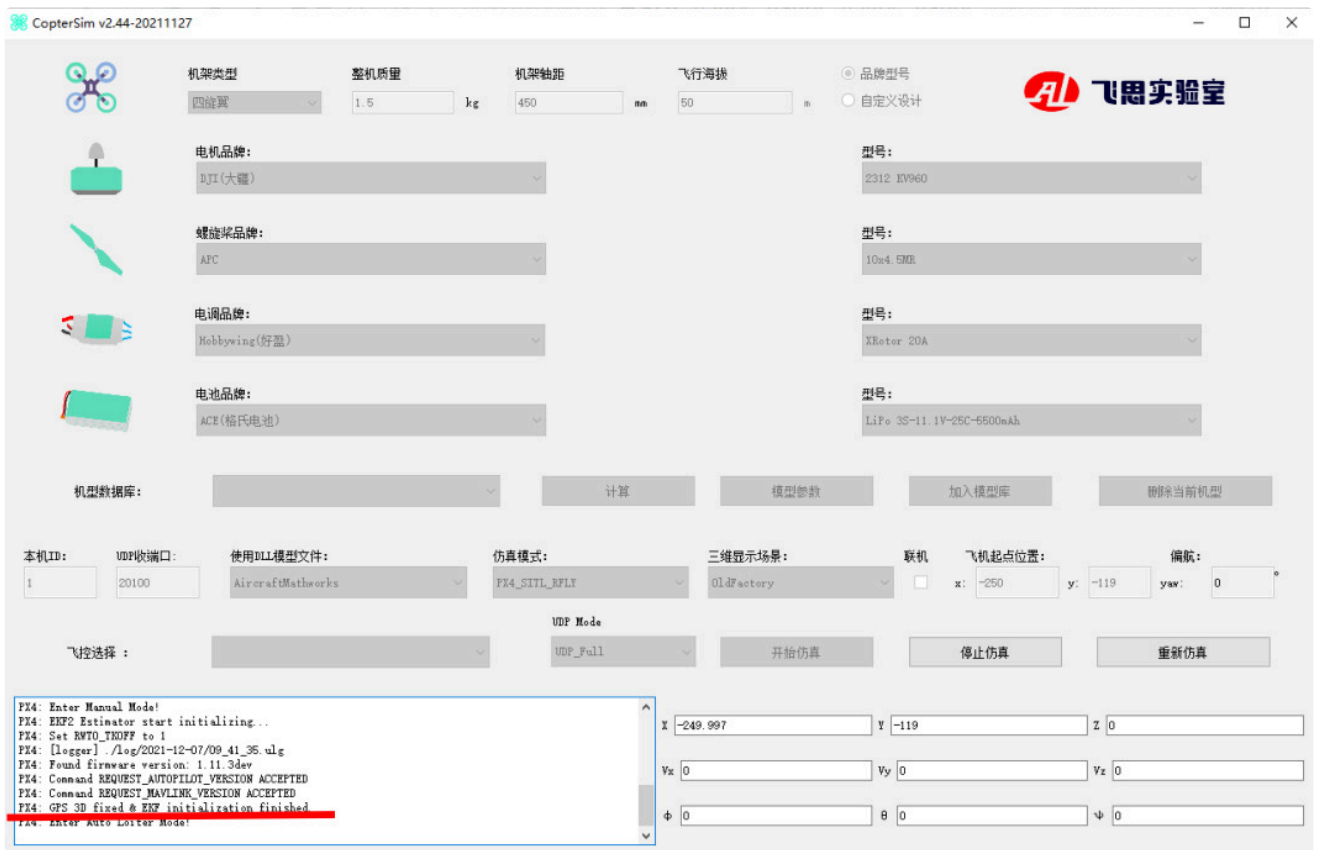


注意，在“FSJ150_model_SITLRun.bat”文件中，机架设置处确认为FSJ150对应的机架。

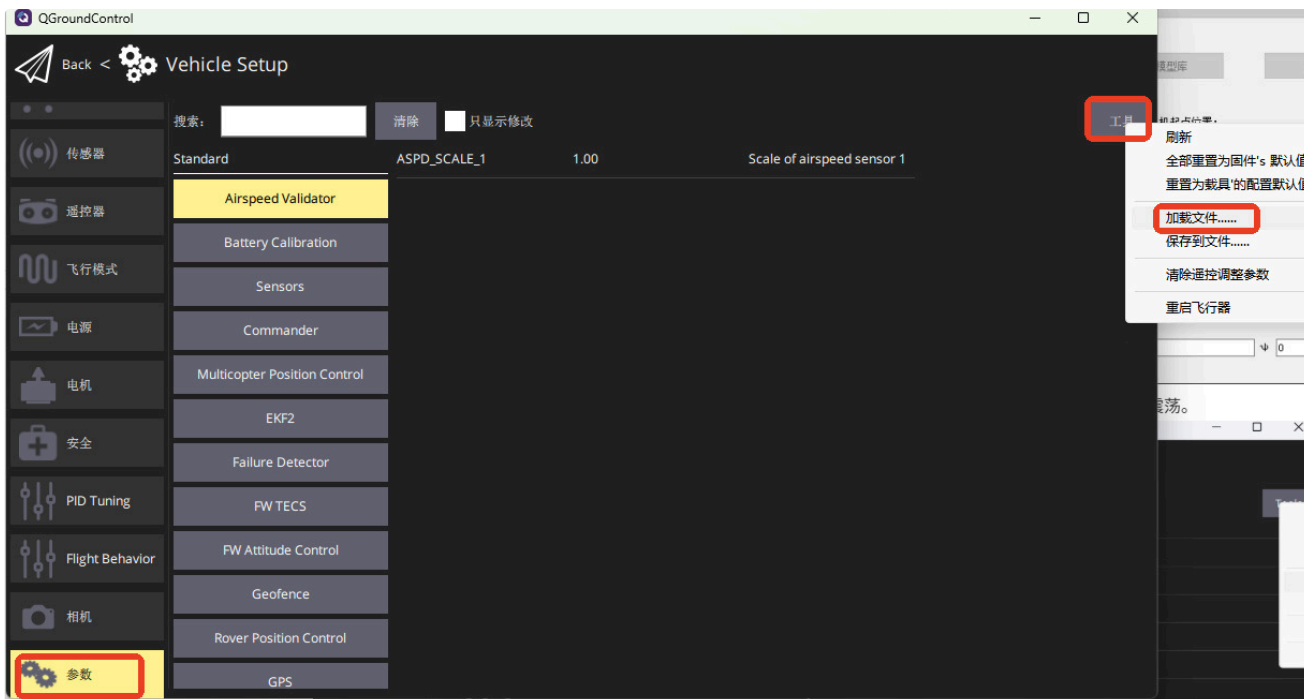
```
REM Set the vehicle-model (airframe) of PX4 SITL simulation, the default airframe is a quadcopter: iris
REM Check folder Firmware\ROMFS\px4fmu_common\init.d-posix (or init.d/airframes) for supported airframes (Note: You can also create your airframe file here)
REM E.g., fixed-wing aircraft: PX4SITLFrame=plane; small cars: PX4SITLFrame=rover
set PX4SITLFrame=iris
```

Step 2: 重新上传飞控参数并等待初始化完成

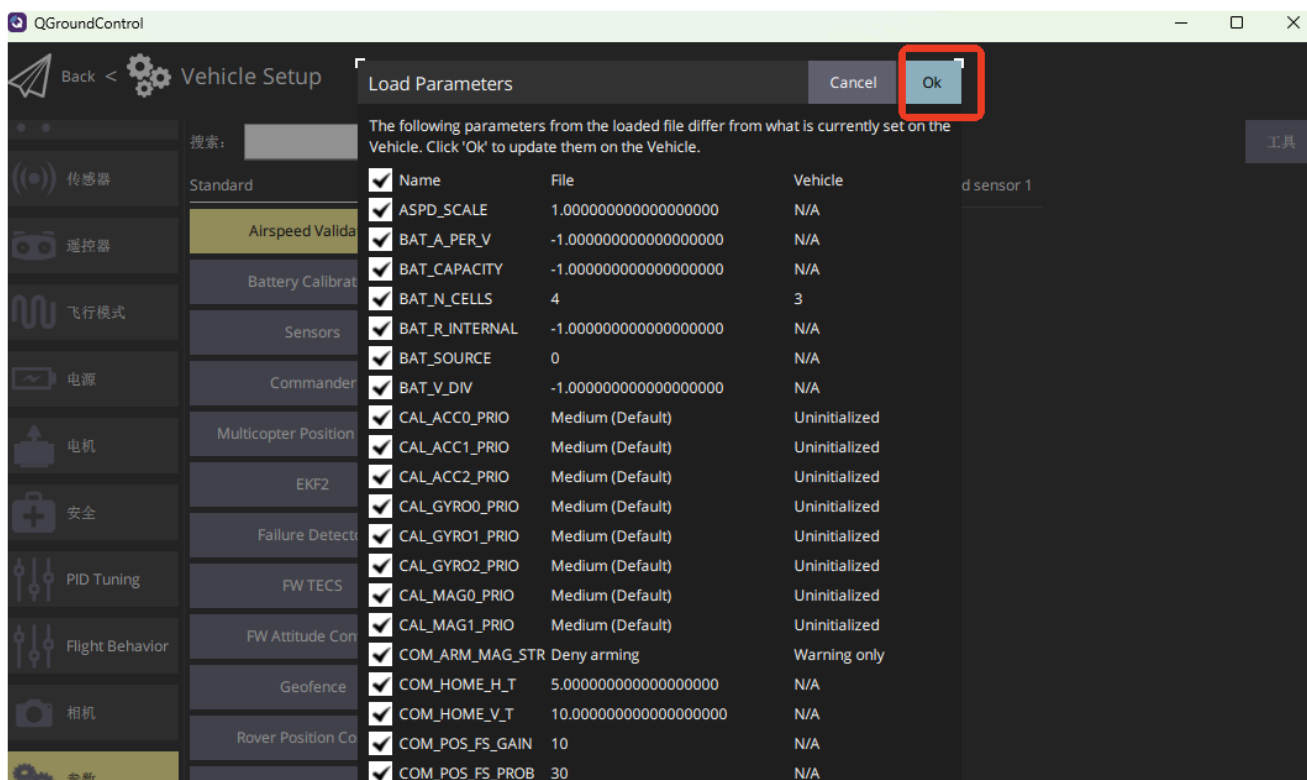
等待 CopterSim 中显示初始化完成。



默认的控制参数pitch角速度的D过大导致飞机震荡
进入QGC的设置页面，依次点击QGC的“参数”、“工具”、加载文件。



定位到例程文件夹，选择Modify.params参数文件并打开后，点击确认修改参数。

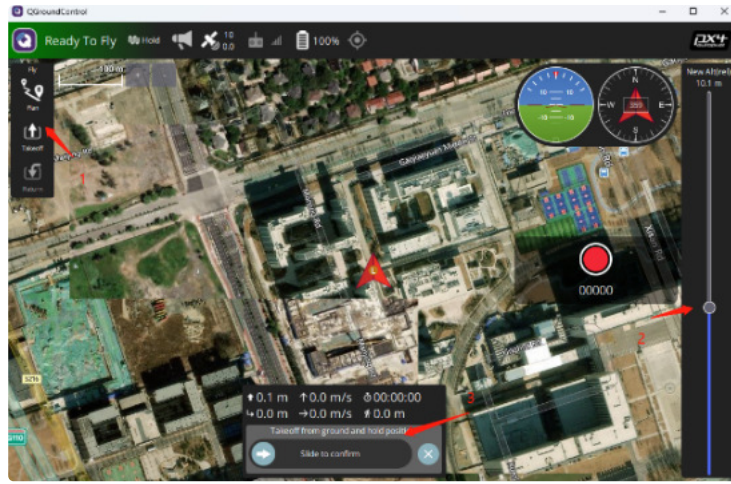


可能提示部分参数上传失败不影响仿真。

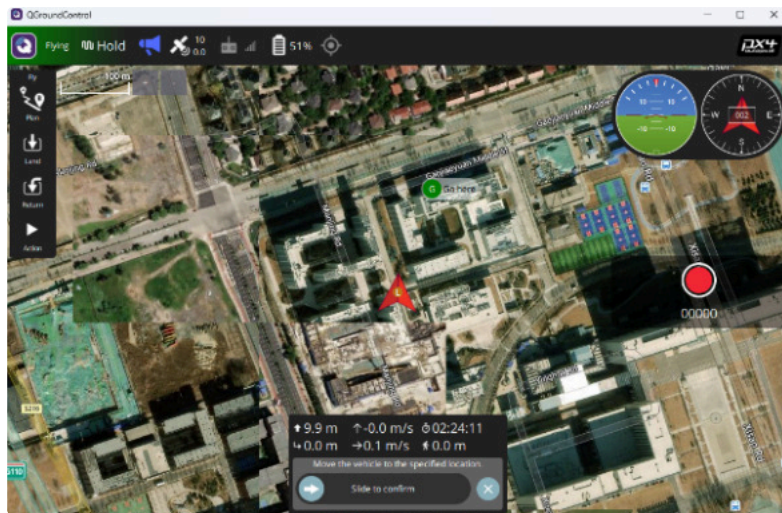
Step 3: 观测结果

在 RflySim3D中观察是否正常起飞、降落以及按照指令飞行。

1. 起飞

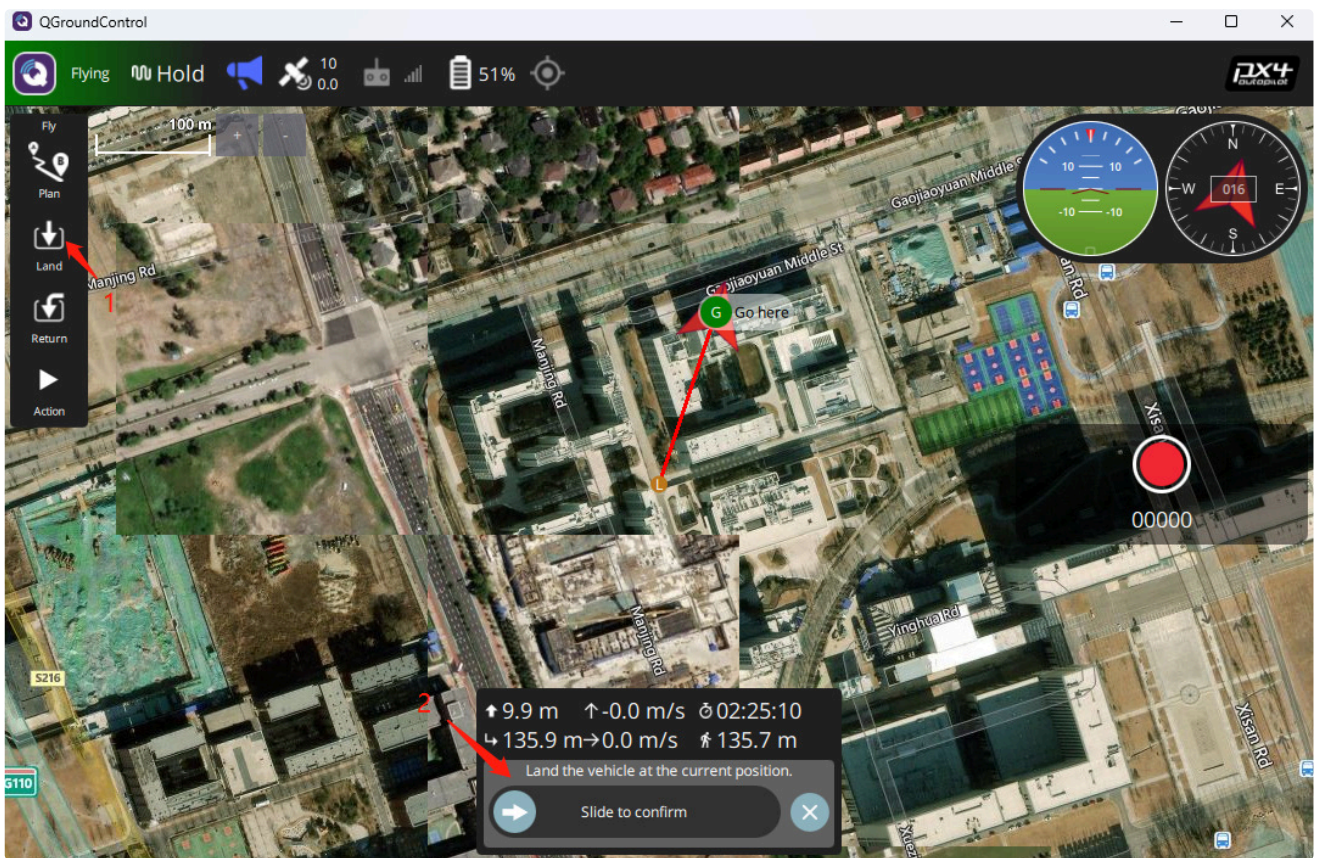


2. 飞行





3. 着陆



4.3. 选做实验：硬件在环仿真

Step 1: 连接飞控

硬件在环仿真需要准备一个飞控，如下图所示，将飞控通过USB线连接电脑，并确保完成硬件在环仿真配置。注意，本图使用Pixhawk6x飞控，其他飞控配置方法类似（推荐使用Pixhawk飞控）。

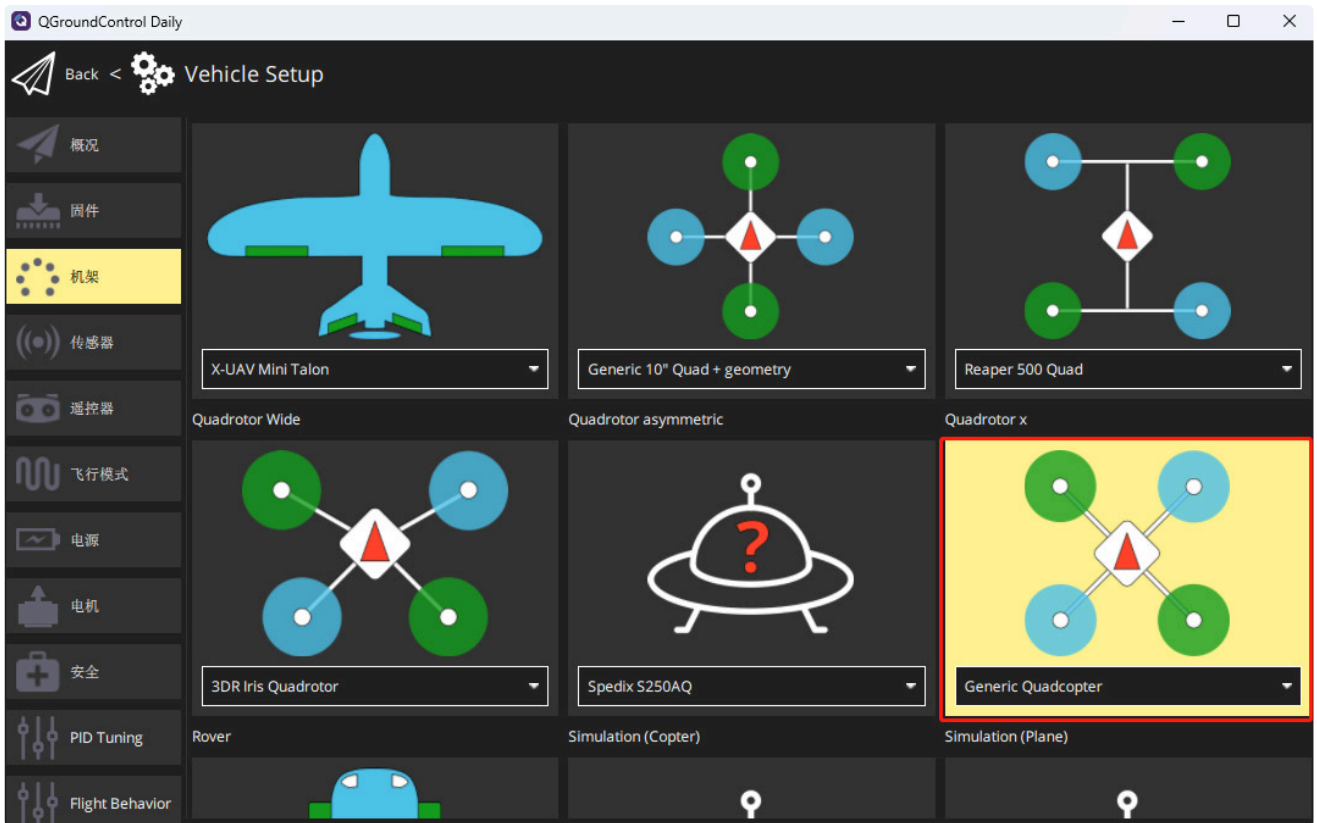


Step 2: 设置硬件在环机架

在 Rflytools 文件夹中打开 QGC 地面站。

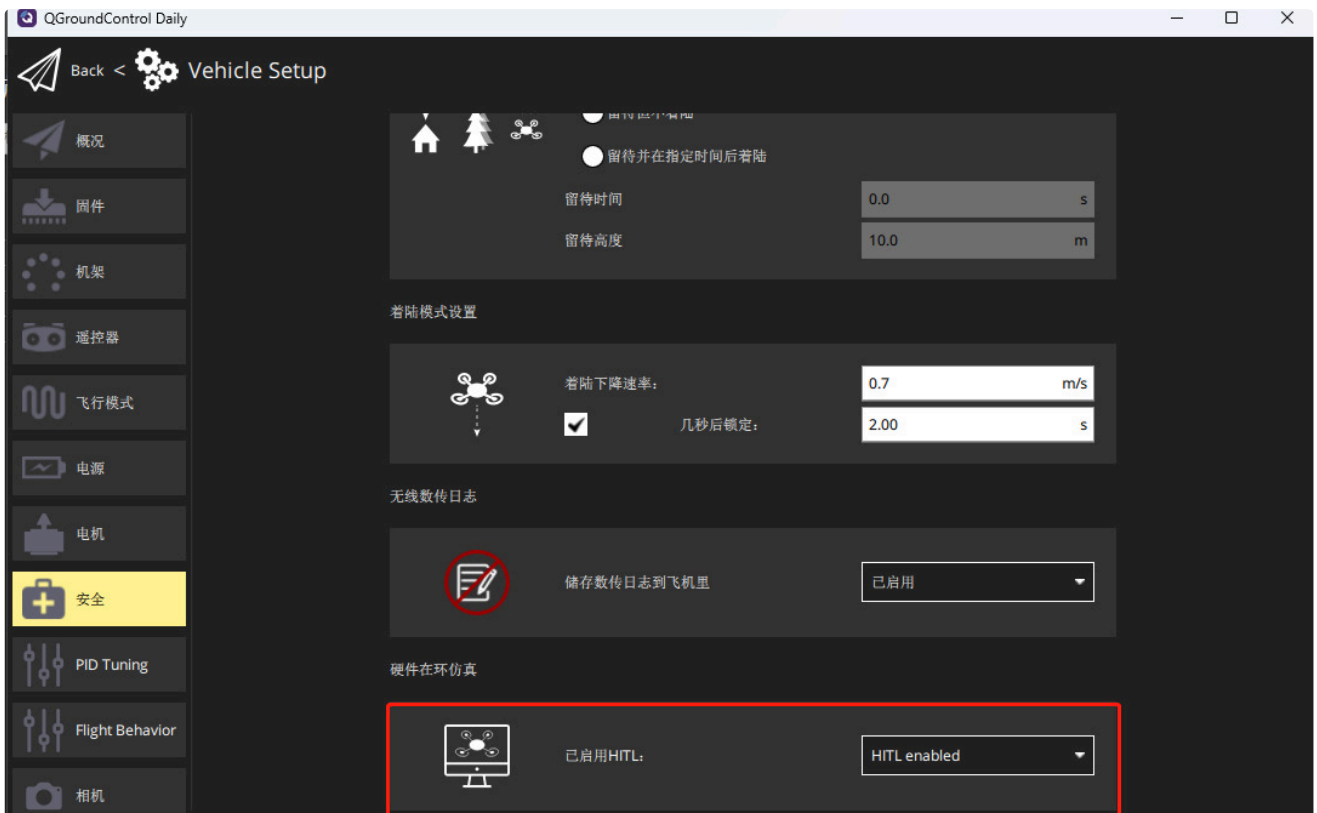
3DDisplay	2023/7/27 15:02	快捷方式	1 KB
CopterSim	2023/7/27 15:02	快捷方式	1 KB
FlightGear-F450	2023/7/27 15:02	快捷方式	2 KB
HITLRun	2023/7/27 15:02	快捷方式	2 KB
Python38Env	2023/7/27 15:02	快捷方式	2 KB
QGroundControl	2023/7/27 15:02	快捷方式	1 KB
RflySim3D	2023/7/27 15:02	快捷方式	1 KB
RflySimAPIs	2023/7/27 15:02	快捷方式	1 KB
RflySimUE5	2023/7/27 15:02	快捷方式	1 KB
SITLRun	2023/7/27 15:02	快捷方式	2 KB
Win10WSL	2023/7/27 15:02	快捷方式	2 KB

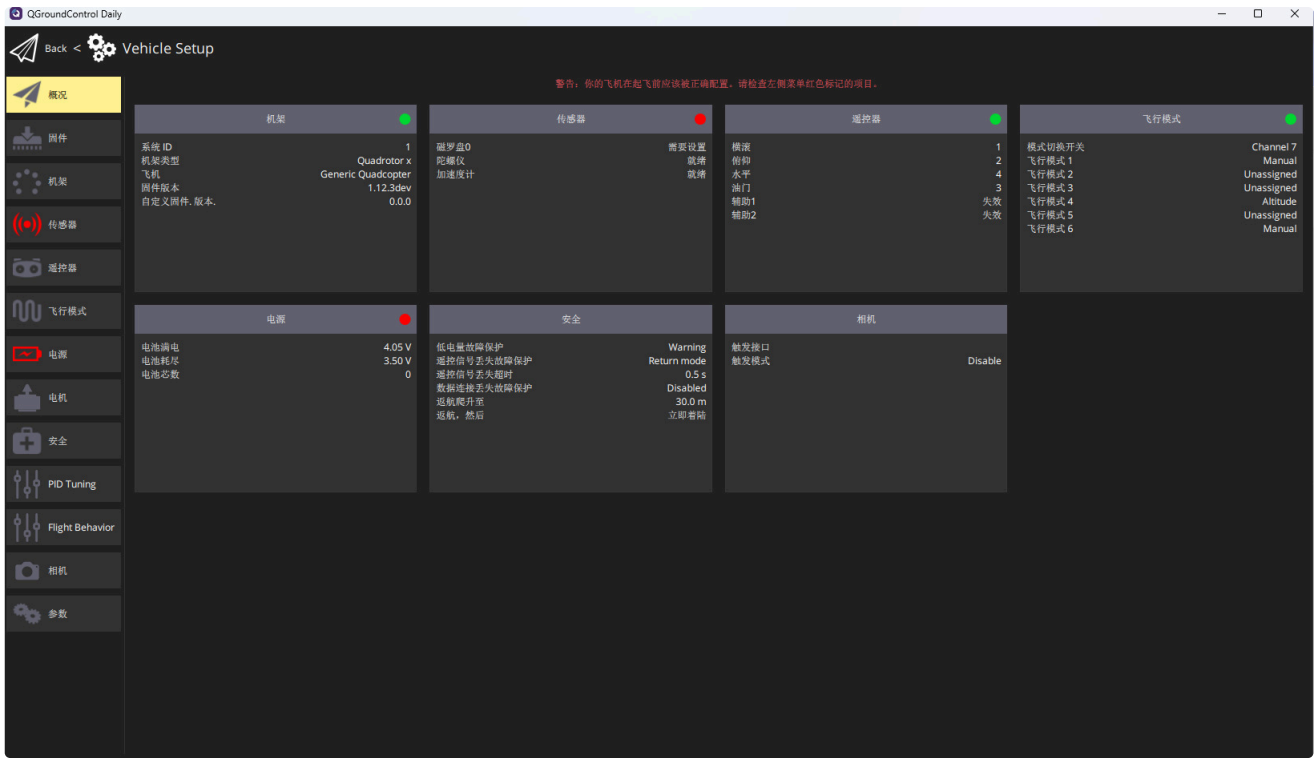
在机架界面设置机架型号为“Generic Quadcopter”，设置完毕后点击右侧“应用并重启”。



Step 3: 配置硬件在环参数

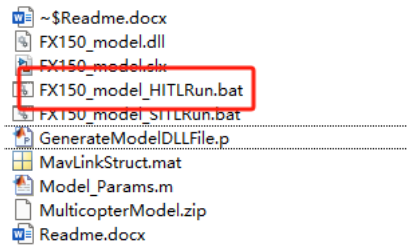
在“安全”界面，选择“HITL enabled”启动硬件在环仿真，之后在概况界面中确认配置完成后，重新插拔飞控完成设置。





Step 4: 启动仿真

右键以管理员身份运行“FSJ150_model_HITLRun.bat”批处理文件，在弹出的终端窗口中根据串口提示输入串口号，启动一架飞机的硬件在环仿真。



```
C:\Windows\system32\cmd.e: X + v
已复制 1 个文件。
-----
Please input the Pixhawk COM port list for HIL
Use ',' as the separator if more than one Pixhawk
E.g., input 3 for COM3 of Pixhawk on the computer
Input 3,6,7 for COM3, COM6 and COM7 of Pixhawks

Available COM ports on this computer are:
COM3: ??????????
COM4: ??????????
COM5: USB ????
-----
Recommended COM list input is: 3,4,5
-----
My COM list for HITL simulation is:5|
```

Step 5: 仿真过程

之后测试步骤与软件在环仿真的Step2到Step3相同，运行之后观察四旋翼能否按照指令飞行。

5. 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](#)

FSJ150_model.slx是基于系统模版构建的FSJ150辨识模型，但本simulink模型中没有用到最大模板相对最小系统模板附加的输入、输出和参数，只是高级版的CopterSim可以读取RflySim3D场景地形高度并传输给DLL模型，可以视为最基本的多旋翼模型。

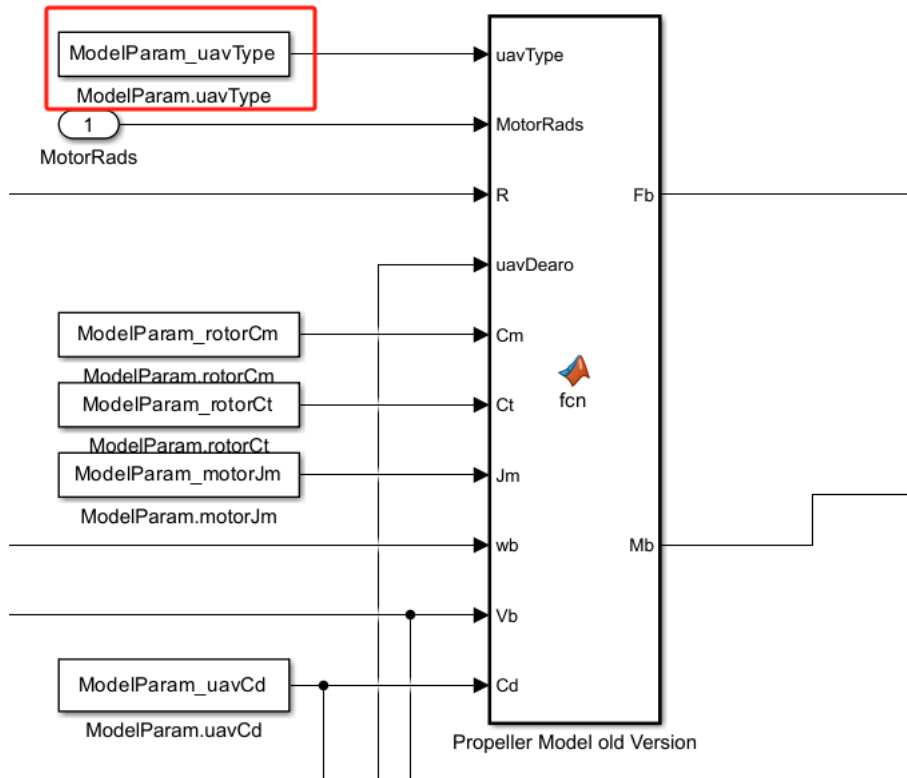
模型参数介绍

重要参数

[FSJ150_H_Model_init.m](#)中定义了FSJ150辨识模型的各种参数，关键数据如下。

飞机的三维显示样式

- 1 | `ModelParam_3DType = int16(251);`定义机型为四旋翼，这个参数决定了飞机的三维显示样式，需要和RflySim3D的XML文件中的ClassID相匹配；
- 2 | `ModelParam_uavType = int16(3);`对于多旋翼飞行器的螺旋桨模型（该模型在力和力矩模块中可见），不同的机型，要对应不同的机架配置以计算力矩分配



飞机的初始位姿参数

```

1 | ModelInit_PosE=[0,0,0];%用于设置飞机的初始位置，对应了CopterSim上的X和Y初始值。Z值利用TerrainZ实现了从CopterSim中读取当前地形高度数据，使得飞机可以
2 | 初始化在复杂地形的地表面（例如Grassland地图）。
   | ModelInit_AngEuler=[0,0,0];%用于设定飞机的初始姿态。飞机姿态角的前两位（俯仰和滚转角）可以通过ModelInit_AngEuler参数来配置，但是偏航角需要在
   | CopterSim中配置。针对导弹等竖直起飞的飞行器，需要设定合适的俯仰和滚转值。

```

QGC中显示的地图坐标和高度原点（在RflySim3D的Cesium大场景中能任意指定飞机在地球三维场景中的坐标）

```

1 | ModelParam_GPSLatLong = [40.1540302 116.2593683];%飞机初始的纬度和经度，单位度。
2 | ModelParam_envAltitude = -50;%原点的海拔高度，竖直向下为正，高于海平面填负值，单位米。

```

执行器的初始参数

```

1 | ModelInit_Inputs = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]; % 十六维输入向量，定义电机PWM初始值，默认全0，对固定翼和小车需要修改，因为它们的油门在初始状态处于最小值 (-1)，见“Motor Model”模块

```

故障接口参数FaultInParams：可通过外部消息动态改变的32维参数向量，在故障注入或者可变形的异构飞行器上有用，也可动态地调整传感器模型噪声等；与inSILInts和inSILFloats 形成功能互补。

```

1 | FaultParamAPI.FaultInParams = zeros(32,1);% 定义了一个名为 FaultInParams的32维向量，该向量被初始化为所有元素都为零。
2 | FaultParamAPI.FaultInParams(3)=1;% 将 FaultInParams 向量的第三个元素设置为1。

```

FSJ150辨识模型关键动力学参数如下：

参数名称	.m文件参数名称	参数值
质量	ModelParam.uavMass	0.319 (kg)
转动惯量矩阵	ModelParam.uavJ	$\begin{bmatrix} 4.603658832003199e - 04 & 0 & 0 \\ 0 & 5.356539710941770e - 04 & 0 \\ 0 & 0 & 8.290048007402470e \end{bmatrix}$
多旋翼机身半径	ModelParam.uavR	0.075(m)
螺旋桨拉力系数	aT	3.8619
	bT	-0.7873

参数名称	.m文件参数名称	参数值
螺旋桨力矩系数	aM	0.0387
	bM	-0.0073
阻力系数xyz	ModelParam.uavCd	[0.6023, 0.6023, 0.9] (N/(m/s)^2)
阻尼力矩系数pqr	ModelParam.uavCCm	[0.0123 0.0067 0.0121] (N/(rad/s)^2)

参数调用过程

FSJ150_model.slx是用于生成DLL模型的simulink动态建模模板，simulink模型启动运行（编译）时会调用FSJ150_H_Model_init.m



FSJ150_H_Model_init.m中包含了模型的参数信息，本脚本会在FSJ150_model.slx编译（simulink模型编译所需环境配置参考PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf中的环境配置）时被调用将参数载入MATLAB工作空间，也可以直接运行FSJ150_H_Model_init.m将参数载入工作空间。Simulink模型会通过参数名称读取工作空间中的参数，故需要保证simulink模型中设置的参数名称与***_init.m中的参数名称相同。



名称	值
FaultParamAPI	1x1 struct
filepath	'F:\d2\4.RflySimMo...
HILGPS	1x1 Bus
MavLinkGPS	1x1 Bus
MavLinkSensor	1x1 Bus
MavLinkStateQuat	1x1 Bus
MavVehileInfo	1x1 Bus
ModelInit_AngEuler	[0,0,0]
ModelInit_Inputs	1x16 double
ModelInit_PosE	[0,0,0]
ModelInit_RateB	[0,0,0]
ModelInit_RPM	0
ModelInit_VelB	[0,0,0]
ModelParam_envAltitude	-50
ModelParam_GPSLatLong	[40.1540,116.2594]
ModelParam_motorCr	842.1000
ModelParam_motorJm	1.2870e-04
ModelParam_motorMinThr	0.0500
ModelParam_motorT	0.0214
ModelParam_motorWb	22.8300
ModelParam_rotorCm	2.7830e-07
ModelParam_rotorCt	1.6810e-05
ModelParam_uavCCm	[0.0035,0.0039,0.00...
ModelParam_uavCd	0.0550
ModelParam_uavDeard	0.1200
ModelParam_uavJ	[0.0211,0,0,0.0219...
ModelParam_uavMass	1.5150
ModelParam_uavMotNumbs	4
ModelParam_uavR	0.2250
ModelParam_uavType	3

GenerateModelDLLFile.p是将slx模型转化为DLL模型文件的脚本，使用RflySim平台进行载具软硬件在环仿真时，需要将DLL(windows下)/SO (Linux下)模型导入到CopterSim，形成运动仿真模型，因此，在Simulink模型编译完成后，需要将模型对应的C++文件打包成DLL/SO模型。

输入信号

输入数据包括电机控制量、地形数据等。

电机数据inPWMs

输入接口inPWMs，16维执行器控制量输入，已归一化到-1到1尺度(通常电机是0-1，舵机是-1~1)，它的数据来自飞控回传的电机控制MAVLink消息mavlink_hil_actuator_controls_t的controls，具体定义如下：

```

1 typedef struct __mavlink_hil_actuator_controls_t {
2
3     uint64_t time_usec; //时间戳，从开机后的时间，单位ms
4
5     uint64_t flags; //标志位，用于显示当前的飞行状态
6
7     float controls[16]; //控制量，16维电机的控制量，发送到模型中，驱动飞机飞行
8
9     uint8_t mode; //模型，用于显示飞机当前的飞行模式和是否上锁等信息})
10 mavlink_hil_actuator_controls_t;

```

软件在环仿真时，电机控制指令从PX4 SITL控制器通过TCP4561系列端口以MAVLink协议发送到运动仿真模型的inPWMs接口，而硬件在环仿真时，该指令是从飞控通过串口以MAVLink协议发送到运动仿真模型的inPWMs接口。

地形高度terrainZ

最大模板可以利用TerrainZ

实现从CopterSim中读取当前地形高度数据，使得飞机初始化为在复杂地形的地表面（例如RflySim3D中的Grassland地图）。这个值是由CopterSim读取DLL模型初始位置参数ModelInit_PosE中的xy坐标，根据地形校准文件及高程信息解算出地形高度TerrainZ，通过Mavlink消息传输给DLL模型的TerrainZ接口，在DLL模型中通过PhysicalCollisionModel/GroundSupportModel/ Ground Model函数中重新定义模型初始位置的高度，最后会通过MavVehile3DInfo接口传给RflySim3D中的三维显示模型。

```

function [F1,M1] = fcn(mg,wb,Xe,Ve,Quet,terrainZ)
% This is a ground model that can provides reaction force and r

% persistent z0;
% if isempty(z0)
%     z0=0;
% end

persistent takeoffFlag;
if isempty(takeoffFlag)
    takeoffFlag=int8(0);
end

persistent landFlag;
if isempty(landFlag)
    landFlag=int8(1);
end

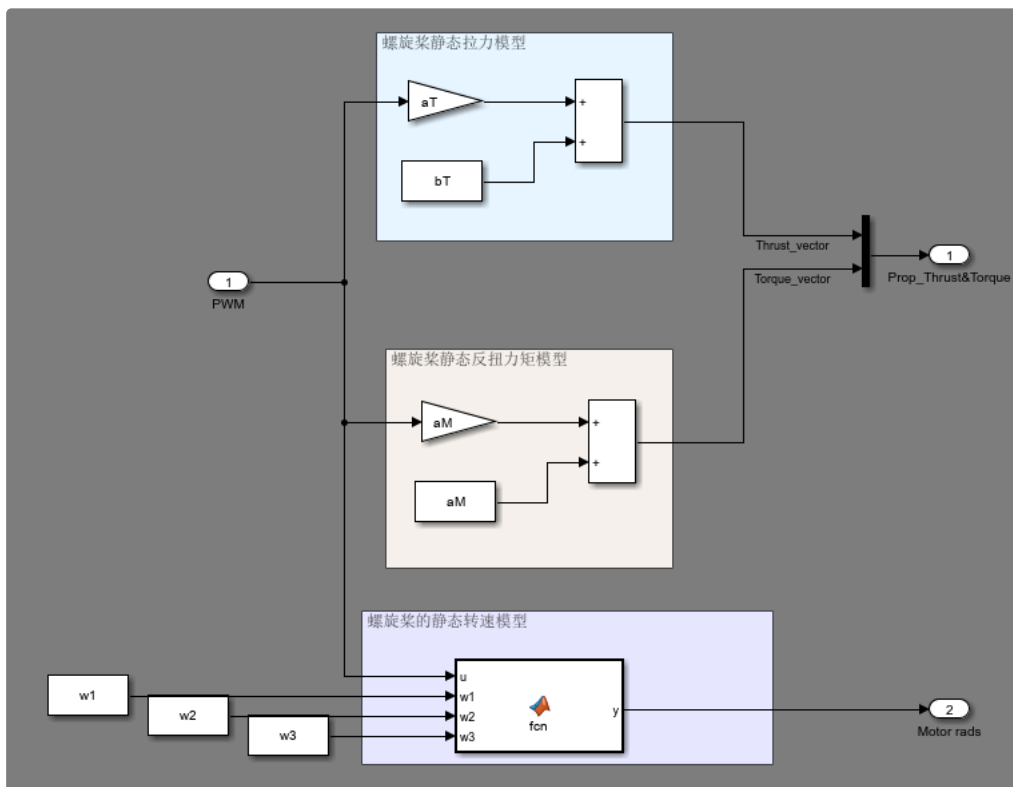
z=Xe(3)-terrainZ;

```

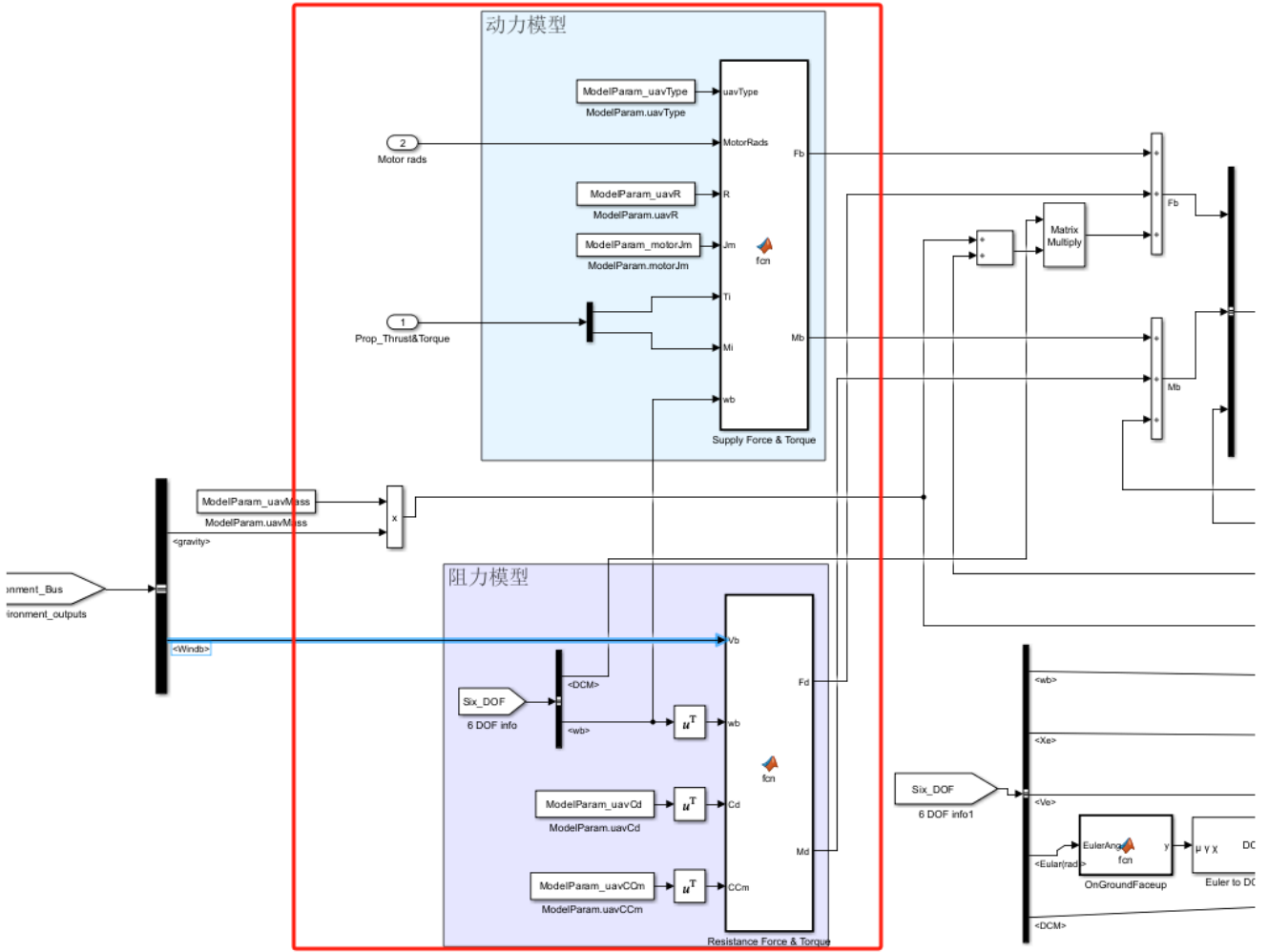
模型模块简介

静态螺旋桨模块

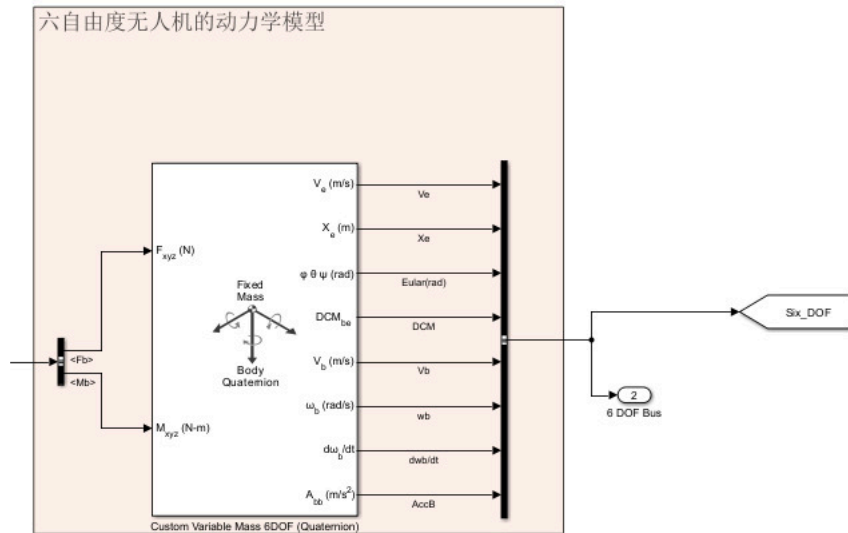
- ① 此处的螺旋桨模型采用的是电机PWM输入到拉力、反扭力矩输出的静态模型，而非通过螺旋桨理论推导的一般形式模型。
- ② 拉力模型为实飞数据辨识所得。反扭力矩、转速模型为动力测试台测量所得。
- ③ T单位为N，Q单位为N.m，radW单位为rad/s。



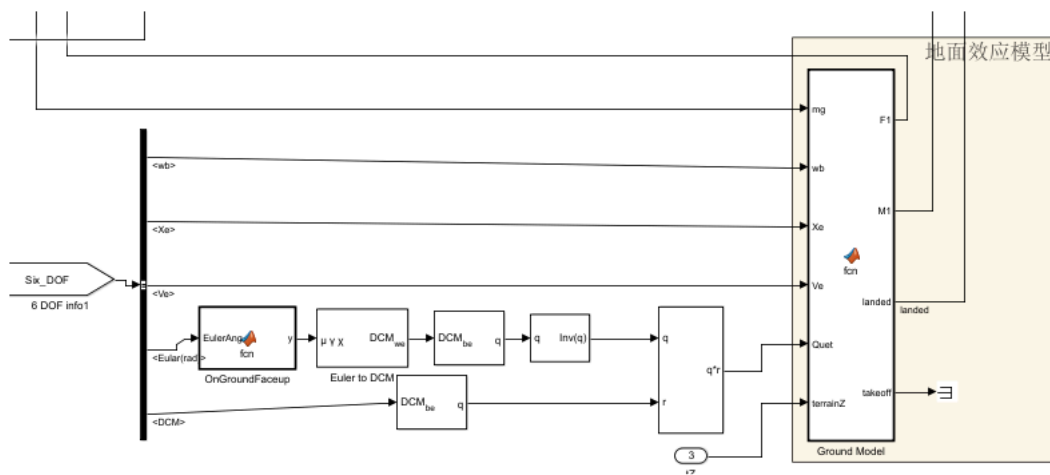
动态力和力矩模块



6DOF六自由度刚体模块



GroundSupportModel地面支撑模块

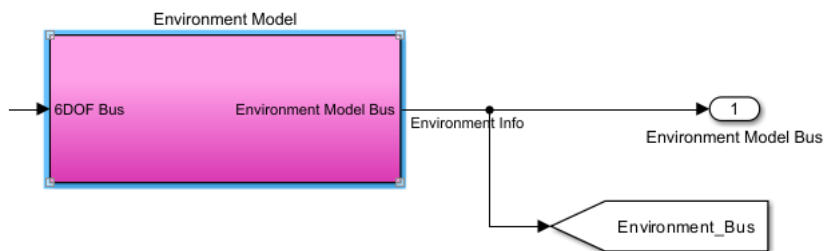


SensorOutput传感器输出模块

该模块中包括了环境模型、传感器模型和GPS模型。

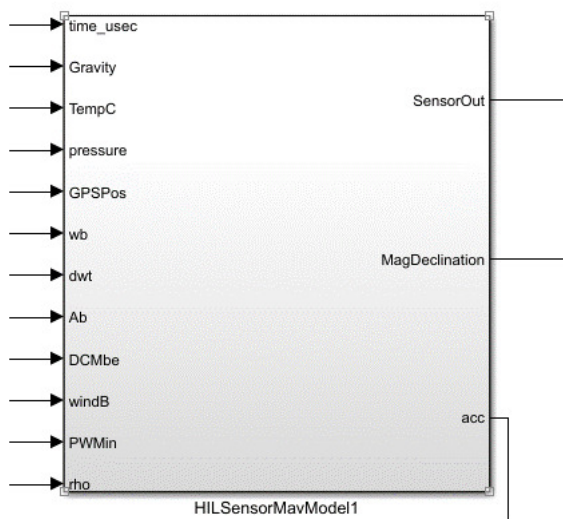
环境模型

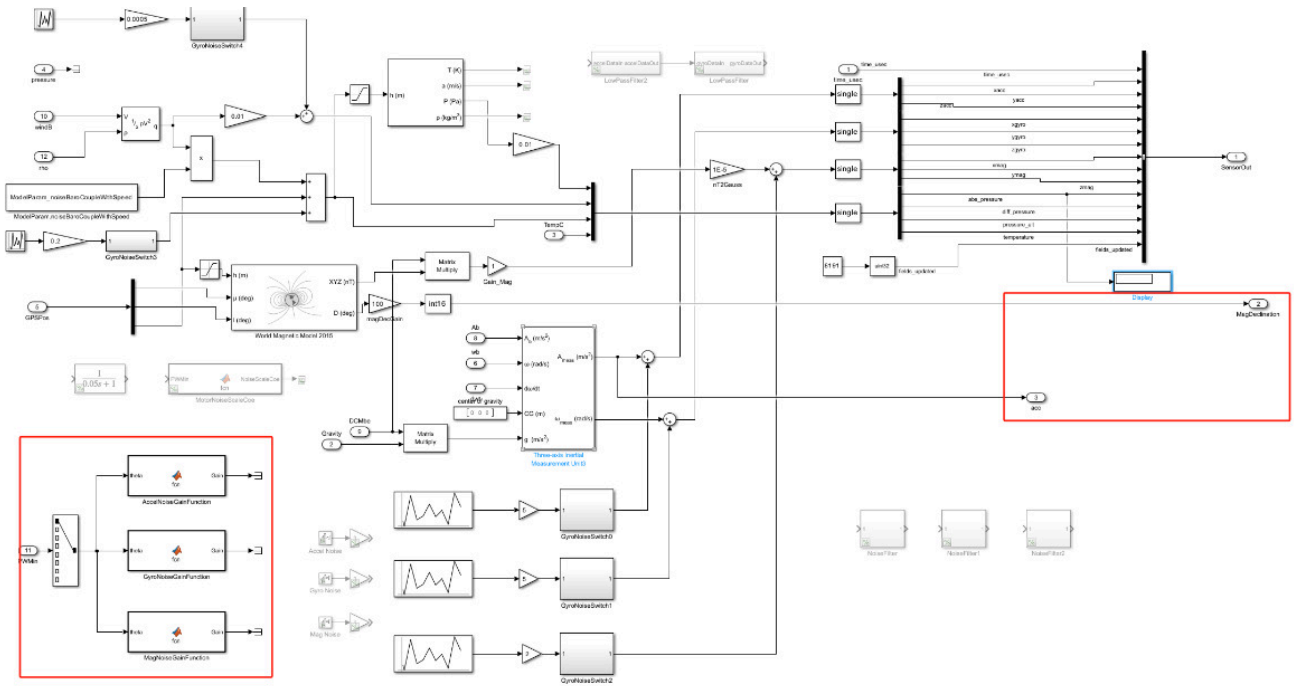
环境模型对重力和大气压强对无人系统飞行产生的影响进行了模拟



传感器模型

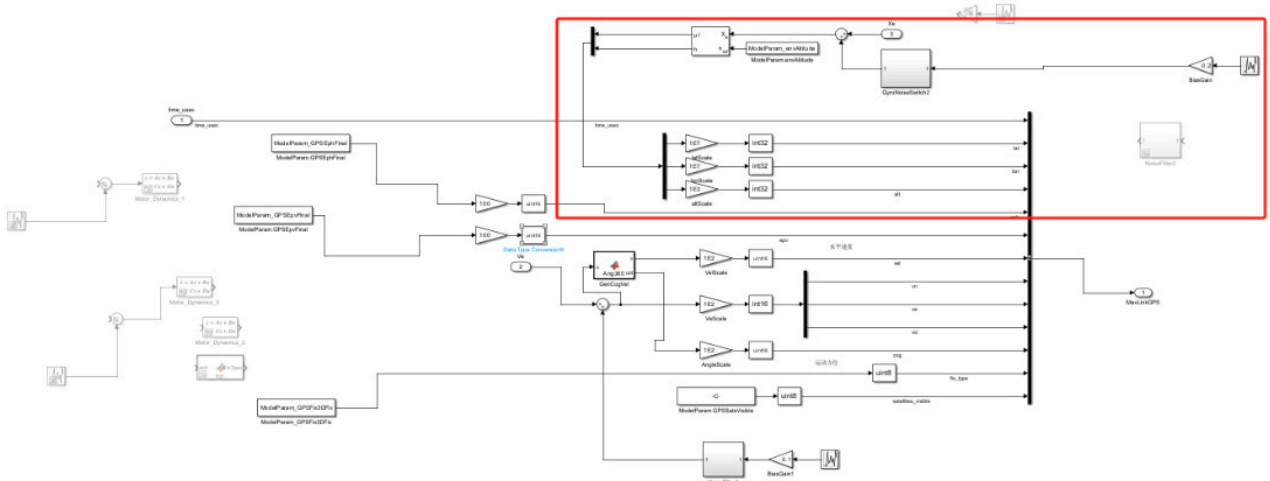
传感器模型中不仅对磁力计、惯性导航进行了建模，同时加入了噪声模拟





GPS模型

GPS模型用于计算GPS数据，在仿真时反馈回PX4控制器



输出信号

输出信号，分别是HILSensor30d、HILGPS30d、VehicleInfo60d。

HILSensor30d (传感器接口集合)

模型发送给飞控的各种传感器数据的集合，对应了MAVLink的mavlink_hil_sensor_t消息，本结构体包含了，加速度传感器的加速度值、陀螺仪传感器的角速度值、磁罗盘传感器的磁场值，气压和空速传感器的气压值等。这些传感器的值在仿真时由我们的模型提供，在真机飞行时由真实传感器芯片提供。

```

1  typedef struct __mavlink_hil_sensor_t {
2
3  uint64_t time_usec; /*时间戳, 单位毫秒*/
4
5  float xacc; /*机体坐标系x方向加速度, 单位m/s^2 */
6
7  float yacc; /*机体坐标系y方向加速度, 单位m/s^2 */
8
9  float zacc; /*机体坐标系z方向加速度, 单位m/s^2 */
10
11 float xgyro; /*机体坐标系x方向角速度, 单位rad/s */
12
13 float ygyro; /*机体坐标系y方向角速度, 单位rad/s */
14
15 float zgyro; /*机体坐标系z方向角速度, 单位rad/s */
16
17 float xmag; /*机体坐标系x方向磁通量, 单位Gauss =T/10000*/
18
19 float ymag; /*机体坐标系y方向磁通量, 单位Gauss =T/10000*/
20
21 float zmag; /*机体坐标系z方向磁通量, 单位Gauss =T/10000*/
22
23 float abs_pressure; /*绝对气压值, 单位 millibar=100Pa*/
24
25 float diff_pressure; /*相对气压值, 单位 millibar=100Pa*/
26
27 float pressure_alt; /*气压解算高度值, 单位m*/
28
29 float temperature; /*温度, 单位摄氏度*/
30
31 uint32_t fields_updated; /*传感器参数初始化标志位, bit 0 = xacc, bit 12:
32 temperature, bit 31:全部重新初始化 */
33
34 } mavlink_hil_sensor_t;

```

■ HILGPS30d (GPS接口)

模型发送给飞控的GPS数据值, 它对应了MAVLink消息的mavlink_hil_gps_t结构体。输出信号中包含了经纬高、水平竖直精度、地速、北东地的速度、偏航角、定位状态和卫星数量等数据。

这些传感器的值在仿真时由我们的模型提供, 在真机飞行时由真实GPS模块提供。

```

1  typedef struct __mavlink_hil_gps_t {
2
3  uint64_t time_usec; /*时间戳, 单位毫秒*/
4
5  int32_t lat; /*纬度(WGS84地球模型), 单位度, 再乘以 1E7*/
6
7  int32_t lon; /*经度(WGS84地球模型), 单位度, 再乘以 1E7*/
8
9  int32_t alt; /*高度 (AMSL地球模型, 而不是 WGS84), 单位m, 再乘以1000
10 (向上为正)*/
11
12 uint16_t eph; /*GPS水平方向定位精度, 单位cm, 如果不知道设为 65535*/
13
14 uint16_t epv; /*GPS竖直方向定位精度, 单位cm, 如果不知道设为 65535*/
15
16 uint16_t vel; /*GPS地速, 单位cm/s, 如果不知道设为 65535*/
17
18 int16_t vn; /*GPS地速朝北方向分量, 单位cm/s */
19
20 int16_t ve; /*GPS地速朝东方向分量, 单位cm/s */
21
22 int16_t vd; /*GPS地速朝下方向分量, 单位cm/s */
23
24 uint16_t cog; /*运动方向, 单位和范围0~359.99度, 再乘以100 degrees *
25 100, 如果不知道设为 65535*/
26
27 uint8_t fix_type; /*定位类型 0-1: no fix, 2: 2D fix, 3: 3D fix. */
28
29 uint8_t satellites_visible; /*可见卫星数, 如果不知道设为255*/
30
31 } mavlink_hil_gps_t;

```

注: GPS数据的发送频率与真实传感器硬件基本相同为10Hz, 因此飞控的实时位置并不能靠GPS直接提供, 需要与IMU等传感器进行融合滤波估计得到。

VehileInfo60d（真实仿真数据输出）

模型发送给RflySim3D的真实仿真数据，是平滑的理想值，这些数据可用于Simulink下的飞控与模型进行软件仿真测试。

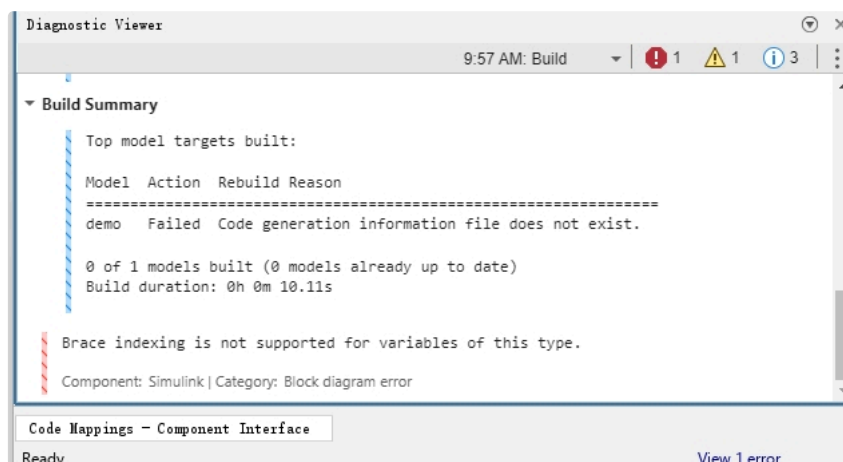
```
1 struct SOut2Simulator {
2
3 int copterID; //飞机ID, 用于区分局域网内不同飞机
4
5 int vehicleType;
6 //飞机样式, 区分同种飞机(如四旋翼)下的不同样式(例如, 大疆、AR.Drone)
7
8 double runnedTime; //时间戳, 当前时刻的时间, 单位毫秒
9
10 float VelE[3]; //速度向量, 地球坐标系的xyz速度(z向下为正), 单位m/s
11
12 float PosE[3];
13 //位置向量, 地球坐标系下的xyz方向(z向下为正, 单位m, 以起飞点为坐标原点)
14
15 float AngEuler[3]; //姿态角, 飞机的欧拉角, 定义于机体坐标系, 单位弧度
16
17 float AngQuatern[4]; //四元数, 飞机姿态的四元数, 定义于机体坐标系
18
19 float MotorRPMS[8]; //电机转速, 飞机的各个旋翼转速, 单位转每分
20
21 float AccB[3]; //加速度, 飞机的运动加速度, 单位m/s^2
22
23 float RateB[3]; //角速度, 飞机的转动角速度, 单位rad/s
24
25 double PosGPS[3]; //GPS坐标, 飞机的经纬高坐标, 单位度、度、米
26
27 };
```

6.参考资料

1. PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf中DLL/SO模型与通信接口的重要参数部分。
2. [\[RflySim安装目录\]/RflySimAPIs/4.RflySimModel/API.pdf](#)
3. [\[RflySim安装目录\]/RflySimAPIs/4.RflySimModel/API.pdf](#)
4. [\[RflySim安装目录\]/RflySimAPIs/4.RflySimModel/API.pdf](#)

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，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

```

命令窗口
>> mex -setup
MEX 配置为使用 'Microsoft Visual C++ 2017 (C)' 以进行 C 语言编译。
警告: MATLAB C 和 Fortran API 已更改, 现可支持
包含 2^32-1 个以上元素的 MATLAB 变量。您需要
更新代码以利用新的 API。
您可以在以下网址找到更多的相关信息:
http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit

要选择不同的 C 编译器, 请从以下选项中选择一种命令:
Microsoft Visual C++ 2013 (C) mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2013.xml C
Microsoft Visual C++ 2015 (C) mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2015.xml C
Microsoft Visual C++ 2017 (C) mex -setup:C:\Users\dream\AppData\Roaming\MathWorks\MATLAB\R

要选择不同的语言, 请从以下选项中选择一种命令:
mex -setup C++
mex -setup FORTRAN
fx >>

```

Q2: 编译报错, 无法加载库文件

```

诊断台日志
-----
Exp1_Motor041Temp 因操作文件完成工作失败。 无法编译。 有关详细信息, 请参见编译日志。 00
编译了 n 个模型, 共 1 个模型(= 个模型已经编译好的)
编译耗时: 0h 0m 3.7699s
无法加载 "zshawk_slib_mbu\CopterSim.mobi" 驱动程序 "zshawk_slib_mbu"。
附件: Simulink (类型: Block diagram 模型)
代码映射 - 应用程序

```

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...

- Software package installation directory
C:\PX4PSP
- PX4 firmware compiling command: firmware versions <= PX4-1.8 use format px4_fmu-v3_default; >= PX4-1.9 use format px4_fmu-v3_default
px4_fmu-v6c_default
- 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
- PX4 firmware compiling toolchain (1: WinWSL[suitable for all versions], 2: Msys2[suitable for <= PX4-1.8], 3: Cygwin[for >=PX4-1.8])
1
- Whether to reinstall PSP toolbox (yes to reinstall and no to remain current installation)
yes
- Whether to reinstall the dependent software packages (CopterSim, QGroundControl, CopterSim, etc. About 5 minutes)
no
- Whether to reinstall the selected compiling toolchain (yes to reinstall and no to remain unchanged, about 5 minutes)
no
- Whether to reinstall the selected PX4 firmware source code (yes to reinstall and no to remain unchanged, about 5 minutes)
no
- Whether to pre-compile the selected firmware with the selected command (yes to compile and no to remain unchanged, about 5 minutes)
no
- 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