

1. 实验名称及目的

1.1 实验名称

基于最小系统模板搭建的差动无人车模型介绍

1.2 实验目的

将Simulink文件编译生成差动无人车的DLL模型文件；并对生成的差动无人车模型在PX4官方控制器下进行软硬件在环仿真测试，通过本例程熟悉平台差动无人车模型的使用

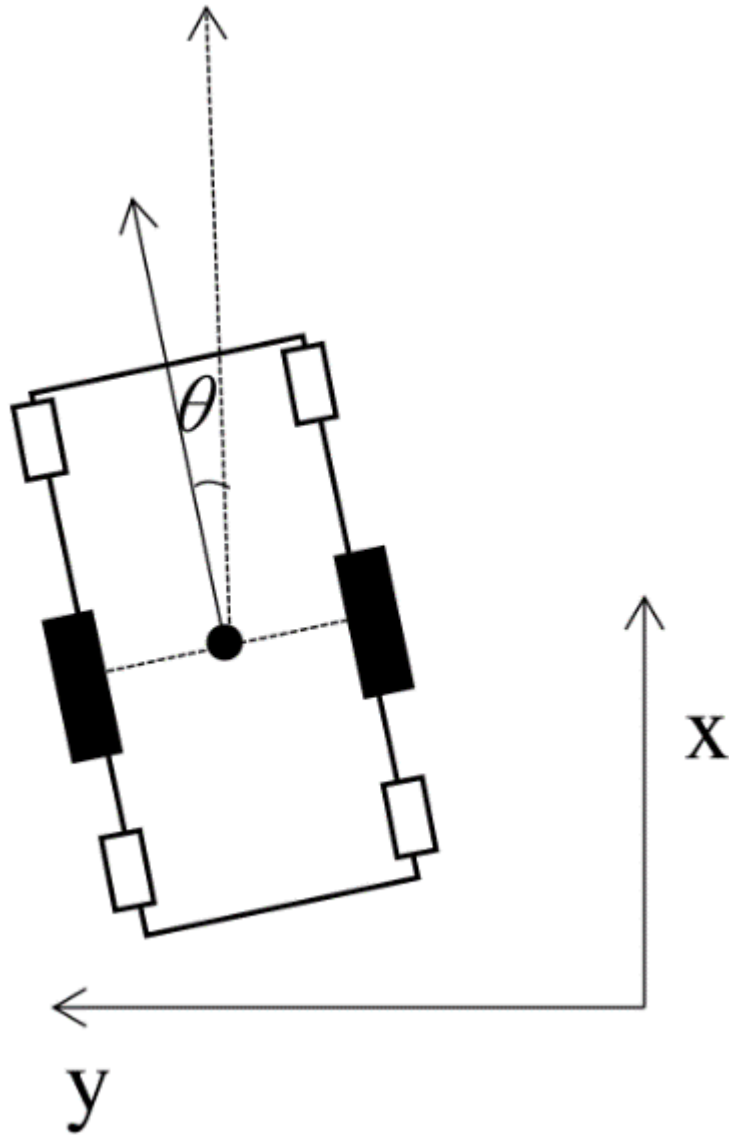
1.3 关键知识点

CarR1Diff.slx是基于最小系统模版建立的差动小车动力学模型。

1. 载具的基本动力学特性

1.1 差速小车运动学约束

所谓差速轮模型，实际上就是两个驱动轮固定在同一根轴上，通过两轮之间的轮速差实现自身的转动，非驱动轮来实现从动及稳定的功能。现在通过运动学来进行一下分析。



假设机器人当前的广义坐标为 $[x, y, \theta]^T$, 代表机器人目前的位置是 $[x, y]$, 车头朝向 θ 方向。机器人可以在世界中自由移动, 但假设它存在无侧滑的约束, 也就是说该机器人的两个主动轮均不可以发生侧向的运动。这是两个约束方程, 实际上, 两个主动轮无侧滑是等效于机器人无侧滑的, 可以等效为一个约束方程。我们来证明这件事

$$p \cdot r \cdot ev_{\perp} = 0 \Leftrightarrow p \cdot car \cdot ev,$$

令驱动轮轴距为 L , 机器人位置为驱动轴的中心。

以左轮为例, 从驱动轴中心到左轮中心的长度即为 $L/2$, 从车辆中心到左轮中心的角度为 $\theta + \pi/2$, 则左轮中心的坐标通过以下公式可以推得

$$\sin(\pi/2 + \theta) = \cos(\theta)$$

$$\cos(\pi/2 + \theta) = -\sin(\theta)$$

$$p_l = [x_l, y_l] = [x, y] + L/2 \cdot [\cos(\pi/2 + \theta), \sin(\pi/2 + \theta)] = [x, y] + L/2 [-\sin(\theta), \cos(\theta)] = [x - L/2 \sin(\theta), y + L/2 \cos(\theta)]$$

同理，右轮中心的坐标为

$$p_r=[x_r,y_r]=[x+L_2\sin(\theta),y-L_2\cos(\theta)]$$

然后我们对左右轮的广义坐标在时间上求导

$$(\dot{\cdot}\sin\theta)=\cos(\theta)$$

$$(\dot{\cdot}\cos\theta)=-\sin(\theta)$$

$$p\cdot l=[x-L_2\cdot\theta\cdot\cos(\theta),y-L_2\cdot\theta\cdot\sin(\theta)]$$

$$p\cdot r=[x+L_2\cdot\theta\cdot\cos(\theta),y+L_2\cdot\theta\cdot\sin(\theta)]$$

上面的式子，描述了在运动中左右轮中心随时间的变化关系。由于左右轮收到了无侧滑的运动约束，我们将上面求出的 $p\cdot l$ 和 $p\cdot r$ 带入约束方程

$$\{p\cdot l\cdot e_{v,\perp}=0, p\cdot r\cdot e_{v,\perp}=0, e_{v,\perp}=[\sin\theta, -\cos\theta]^T$$

两个式子做加法，就得到了如下的关系：

$$x\cdot\sin(\theta)-y\cdot\cos(\theta)=0$$

而 $[x\cdot, y\cdot]$ 就是机器人本身位置的导数。

因此，我们证明了双驱动轮无侧滑约束实际上等价于机器人自身无侧滑约束。

运动学方程

我们证明了差速模型的约束方程只有一个，但事实上我们仍然要对左右驱动轮的轮速分别控制。

机器人的广义速度 $p\cdot$ ，可以表示成左右轮广义速度的平均值

$$p\cdot=(p\cdot l+p\cdot r)/2$$

而左右轮的广义速度，众所周知可以表示为轮子的角速度乘以轮子的半径 $v=w\cdot r, v=w\cdot r$

因此关于速度可以得出如下关系

$$v=(v_l+v_r)/2=r(\omega_l+\omega_r)$$

关于车的航向角 θ ，我们来这么理解，假设如上面示意图所示，车子在向左开，显然右车轮的速度要大于左车轮的速度，才可以让车向左转弯。该航向角的偏航大小，实际上可以理解为右轮多走的那个弧长大小。因此有如下结论

$$\theta\cdot L=(v_r-v_l)t$$

两边对时间 t 求导，就可以得到广义坐标 $\theta \cdot$ 的公式了

$$\theta \cdot = rL(wr - wl)$$

再描述一下广义坐标 $x \cdot$ 和 $y \cdot$

$$x = v \cdot t \cdot \cos\theta$$

$$y = v \cdot t \cdot \sin\theta$$

带入速度 v ，并对时间 t 求导后可以得到

$$x \cdot = r2 \cdot \cos\theta \cdot (wl + wr)$$

$$y \cdot = r2 \cdot \sin\theta \cdot (wl + wr)$$

因此差速轮的运动学模型可以最终表示为

$$[x \cdot \ y \cdot \ \theta \cdot] = [r2\cos\theta \ r2\cos\theta \ r2\sin\theta \ r2\sin\theta - rL \ rL][wl \ wr]$$

在机器人运动控制时，我们通过上层的轨迹跟踪算法，可以得到机器人所需的角速度。通过运动学模型，我们就可以通过控制左右两个驱动轮的轮齿脉冲，来控制机器人的运动轨迹了。

特点

双轮差速模型的优点：结构简单；

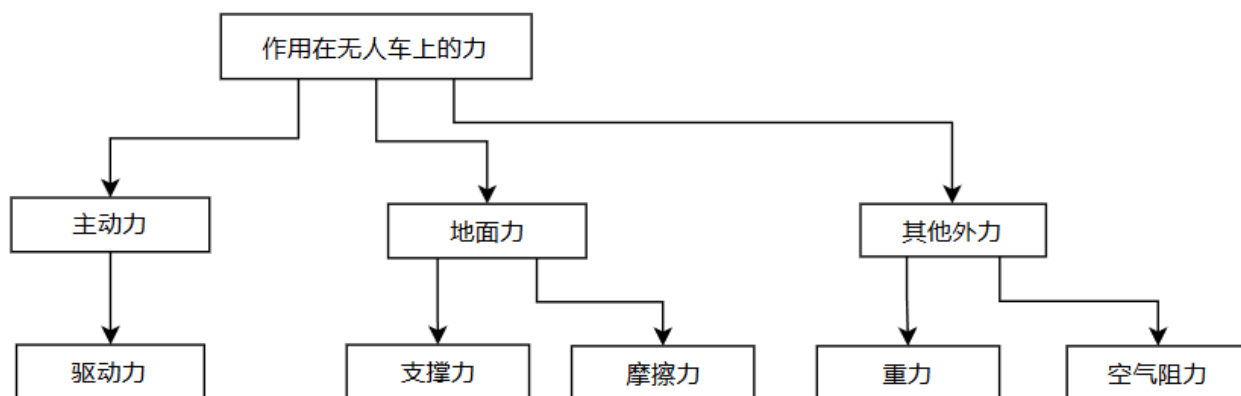
双轮差速模型的缺点：载重小，易侧翻。

在双轮差速模型的基础上，还有四轮差速模型，但相较于双轮差速模型其转弯灵活性更差。市面上大部分机器人也仅仅为双轮差速模型，因此这里暂不做介绍。

力和力矩合成

综合实际的执行器响应、载具运动状态以及环境干扰计算出载具实际受到的力和力矩。

无人车模型的力和力矩都是四个轮胎的传递到整车的，所以在得到轮胎转速后无法和旋翼模型一样，通过结合升力或转矩系数直接得到力和力矩，需要先从轮胎开始分析，得到单个轮胎的力和力矩，再由此得到整车的力和力矩。



载具的控制通道映射

通过混控将归一化的期望力和力矩信号映射到归一化的PWM指令，再根据PWM指令通过电调模块计算出实际的执行器响应。下面结合PX4的混控规则介绍控制通道如何映射到载具模型的执行器输入。

PX4机架对应的混控器

[添加一个新的机型 | PX4 自动驾驶用户指南](#)

- 详细的PX4机架文件配置参考 [\(v1.12\)](#)

本例程所用差速小车的机架型号为aion_robotics_r1_rover，其在\PX4PSP\Firmware\ROMFS\px4fmu_common\init.d\airframes\50003_aion_robotics_r1_rover中定义如下：

```
.$R}etc/init.d/rc.rover_defaults
```

```
param set-default ...
```

执行rc.rover_defaults脚本，它包含了无人车的默认参数设置，可以用来设置一些基本的系统参数和增益。rc.rover_defaults中的关键代码如下：

```
set PWM_OUT 1234
```

```
set MIXER generic_diff_rover
```

设置混控器（mixer）为generic_diff_rover（该混控器用于控制任何差速驱动的载具，即左右电机分别独立驱动，可以原地转向。）

混控通道对应的执行器

[混控器和执行器 | PX4 自动驾驶用户指南](#)

- 详细的PX4混控文件逻辑见：[\(v1.12\)](#)
- 详细的映射过程可参考：[PX4混控器相关知识梳理-CSDN博客](#)

本例的混控文件：在下面的文件夹中可以找到

\PX4PSP\Firmware\ROMFS\px4fmu_common\mixers\generic_diff_rover.main.mix，期望力和力矩信号映射到左右两个电机，适用于差速驱动的载具。

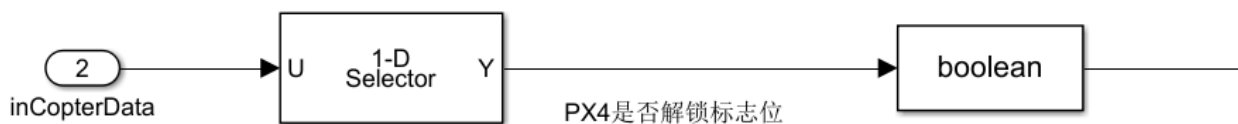
载具模型的整体输入输出和参数

输入输出

- 最小模板的输入输出见：..\..\1.BasicExps\e0_MinModelTemp\Readme.pdf

inCopterData

在最小模板的基础上，多出一个输入接口inCopterData，接收其32维输入的第一位作为执行器解锁标志位



模型参数

参数名	参数	值
三维样式	ModelParam_uavType	int8(50)
初始位置	ModelInit_PosE	[0,0,0]
初始姿态	ModelInit_AngEuler	[0,0,0]
初始速度	ModelInit_VelB	[0,0,0]
初始角速度	ModelInit_RateB	[0,0,0]
初始经纬度	ModelParam_GPSLatLong	[40.1540302 116.2593683]
原点海拔高度	ModelParam_envAltitude	-50
载具质量	ModelParam_uavMass	1.515
转动惯量	ModelParam_uavJ	[0.0211*5,0,0; 0,0.0219*5,0; 0,0,0.0366*5]

参数名	参数	值
阻力系数	ModelParam.uavCd	0.055(N/(m/s)^2)
阻尼力矩系数	ModelParam.uavCCm	[0.0035 0.0039 0.0034] (N/(rad/s)^2)
油门到电机稳态转速曲线斜率	ModelParam.motorCr	842.1
油门到电机稳态转速曲线零点	ModelParam.motorWb	0
电机的初始值	ModelInit_Inputs	[0 0]
电机响应时间常数	ModelParam_motorT	0.0214

2. 实验效果

实现差动无人车DLL模型文件生成，以及完成差动无人车软硬件在环仿真

3. 文件目录

例程目录: [\[安装目录\]](#)\RflySimAPIs\4.RflySimModel\1.BasicExps\e4_CarR1DiffModelCtrl

文件夹/文件名称	说明
CarR1Diff.slx	差动无人车模型文件。
CarR1Diff_HITLRun.bat	硬件在环仿真批处理文件。
CarR1Diff_SITLRun.bat	软件在环仿真批处理文件。
GenerateModelDLLFile.p	DLL格式转化文件。
CarR1Diff_init.m	动力学模型相关参数。
MavLinkStruct.mat	MavLink数据结构体mat文件

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_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；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/>

5. 实验步骤

两小车例程的区别CarR1Diff

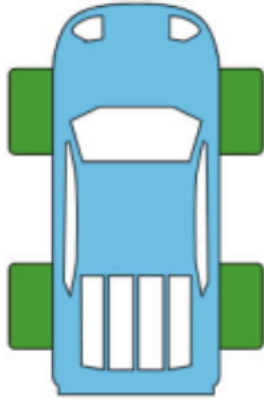
对应了差速小车（控制左右两侧轮子，进行差速转弯，可原地转弯后退），CarAckerman对应了阿克曼底盘小车（控制后轮扭矩+方向盘，不能原地转弯，有转弯半径）

两个小车对应的机架类型。参考PX4的机架配置网

页https://docs.px4.io/main/en/airframes/airframe_reference.html#rover，

可见目前的小车分为两类。其中，CarAckerman对应了常规小车 Generic Ground Vehicle和CarR1Diff对应了差速小车 Aion Robotics R1 UGV。

Rover



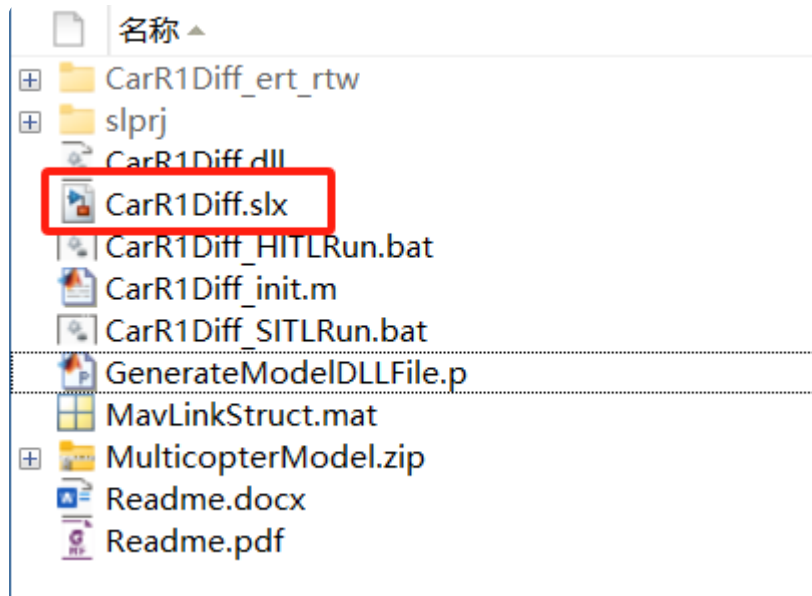
Name	
Generic Ground Vehicle	<p><code>SYS_AUTOSTART</code> = 50000</p> <p>Specific Outputs:</p> <ul style="list-style-type: none">• MAIN2: steering• MAIN4: throttle
Aion Robotics R1 UGV	<p>Maintainer: Timothy Scott</p> <p><code>SYS_AUTOSTART</code> = 50003</p> <p>Specific Outputs:</p> <ul style="list-style-type: none">• MAIN0: Speed of left wheels• MAIN1: Speed of right wheels

5.1. 必做实验：DLL模型生成

Step 1: 编译模型

在MATLAB中打开“CarR1Diff.slx” Simulink 文件，点击“Build Model”按钮。

编译配置可参考 [4.RflySimModel\0.ApiExps\2.UserDefinedC++\2.GenC++\Readme.pdf](#)



注意事项：与多旋翼模型先比，差动小车模型改动如下：

1) 在CarR1Diff.slx的Motor

Model中，硬件在环仿真中设置了只响应1和2号PWM信号，分别给了左侧和右侧轮的速度，遵循Aion

Robotics R1

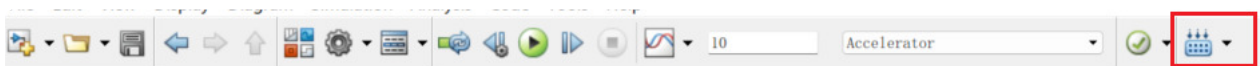
UGV机架模式。

https://docs.px4.io/master/en/airframes/airframe_reference.html#rover。

2) 在CarR1Diff.slx的Force and Moment Model/Ground

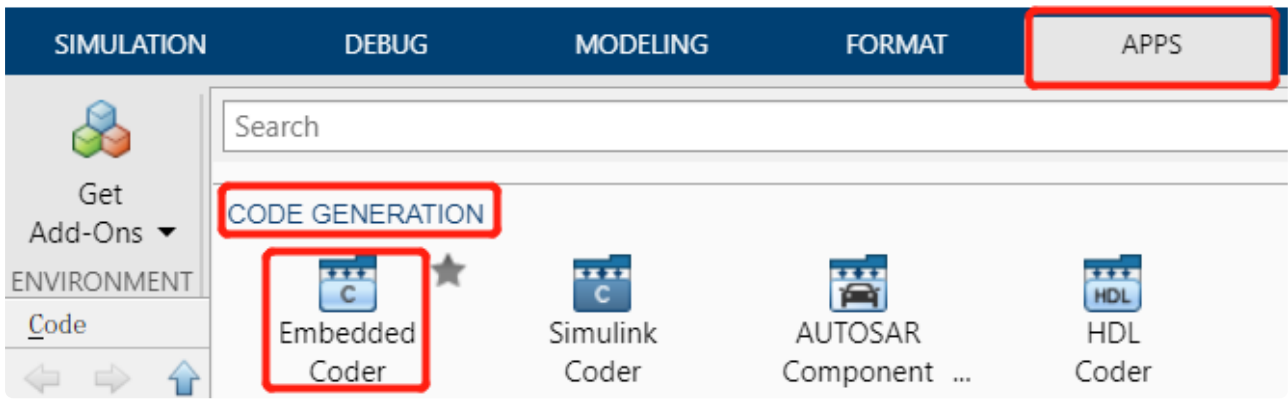
Model模型中个，修改了地面模型，响应四个轮子的摩擦力，来驱动车前进。

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。

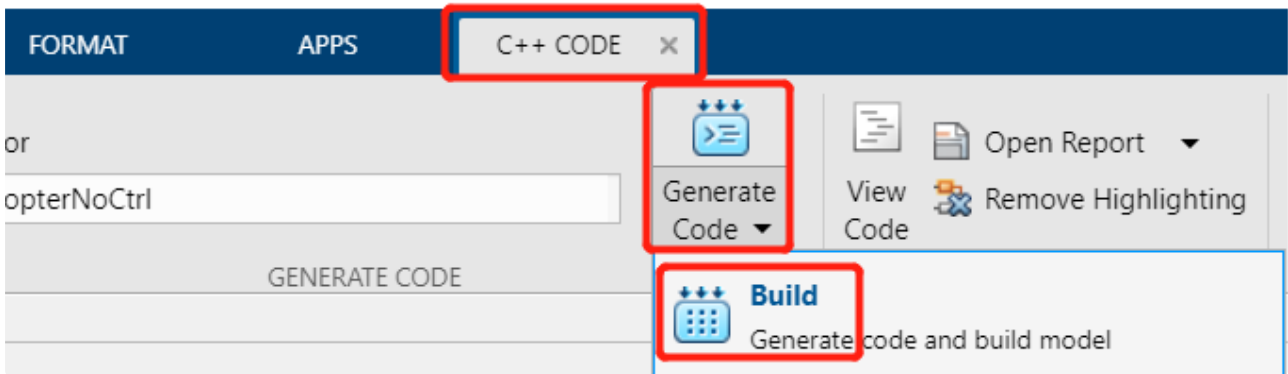


对于2019b及之后版本，点击APPS - CODE GENERATION -Embedded

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。

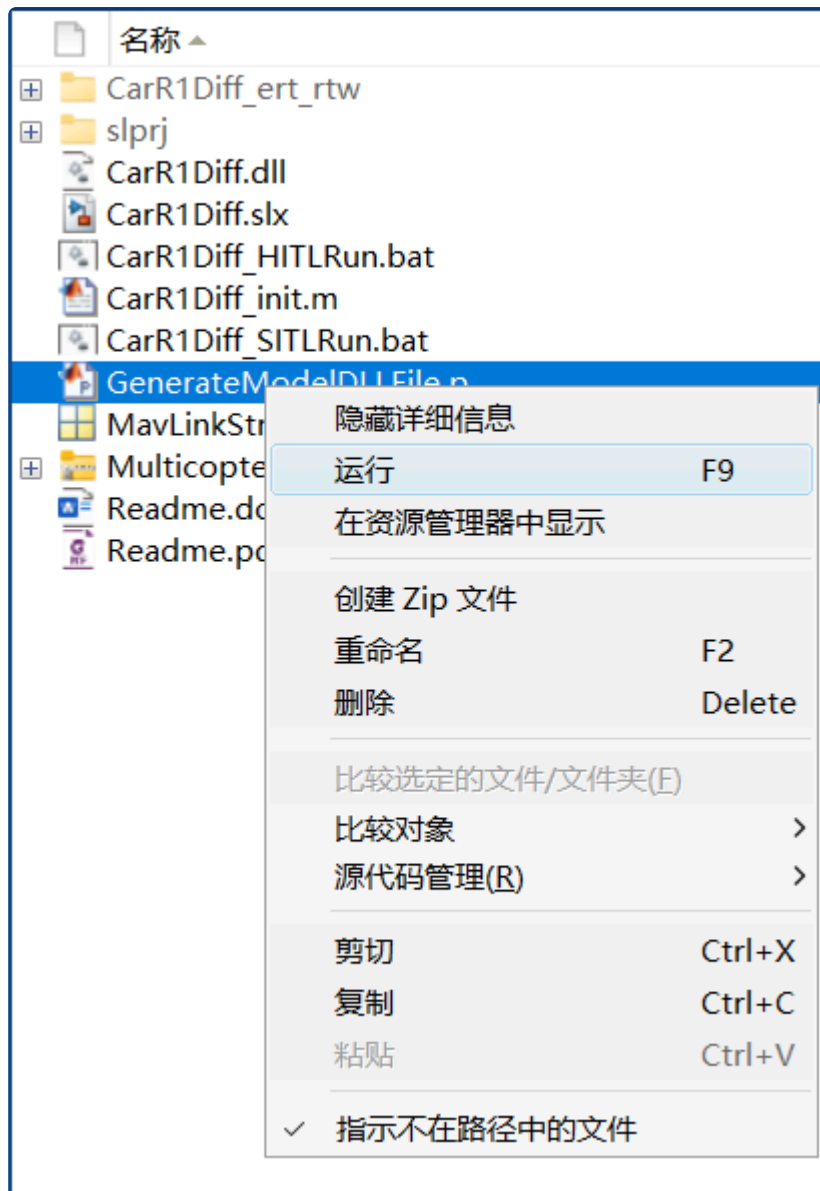


k



Step 2: 生成DLL文件

编译完成后，右键GenerateModelDLLFile.p并点击运行（或者在MATLAB的命令行窗口中输入GenerateModelDLLFile后回车），即可以得到“CarR1Diff.dll”的DLL模型文件。



命令行窗口

```
fx >> GenerateModelDLLFile|
```

命令行窗口

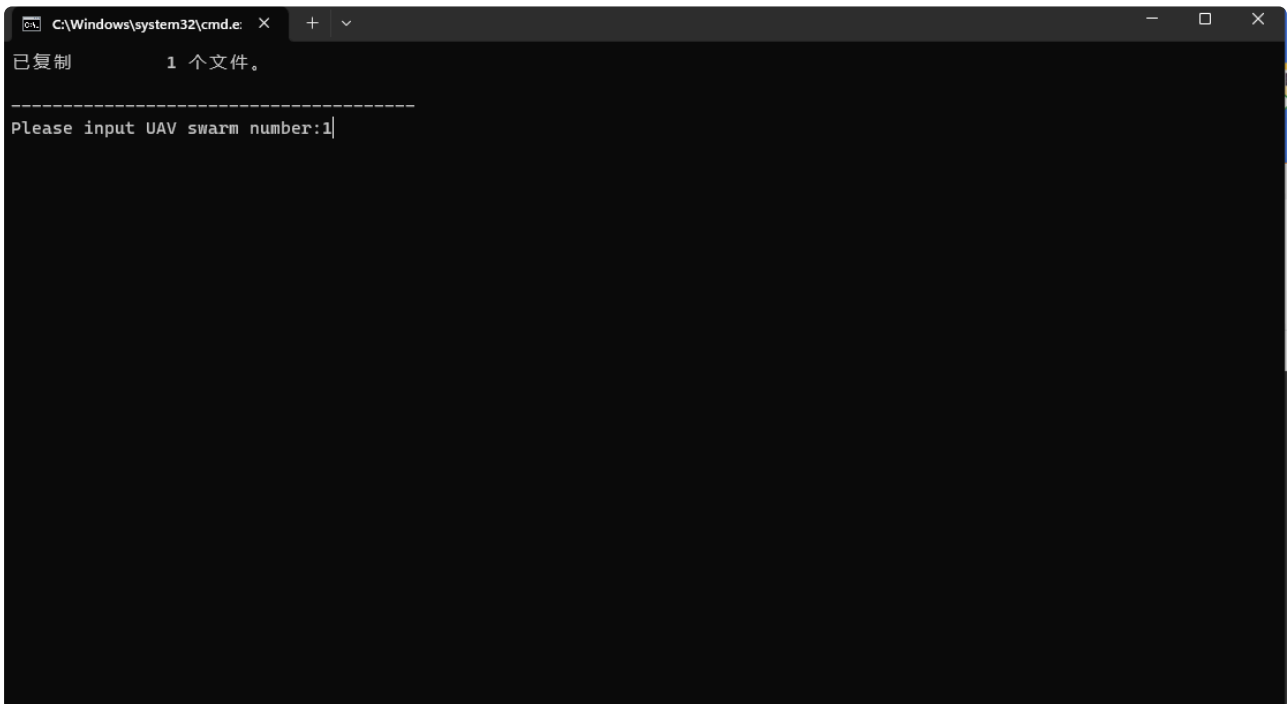
```
Microsoft (R) Incremental Linker Version 14.16.27051.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:modeldllgen.exe
/DLL
/out:CarR1Diff.dll
modeldllgen.obj
CarR1Diff.obj
    正在创建库 CarR1Diff.lib 和对象 CarR1Diff.exp |
Compiling successfully, the CarR1Diff.dll has been generated.
fx >> |
```

5.2. 必做实验：软件在环仿真

Step 1: 启动仿真

双击运行“[CarR1Diff_SITLRun.bat](#)”批处理文件，在弹出的终端窗口中输入1，启动一辆车的软件在环仿真。



注：在“[CarR1Diff_SITLRun.bat](#)”软件在环的脚本文件中，需要设置对应无人车的DLL名：

```
REM Set use DLL model name or not, use number index or name string
REM This option is useful for simulation with other types of vehicles instead of multicopters
set DLLModel=CarR1Diff
```

在SimMode处选择CopterSim中对应的软件在环仿真模式：

```
REM Set the simulation mode on CopterSim, use number index or name string
REM e.g., SimMode=2 equals to SimMode=PX4_SITL_RFLY
set SimMode=2
```

在机架设置处设置差动无人车的对应机架，若不设置对应机架则仿真的默认机架为四旋翼：

```
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 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=rover
```

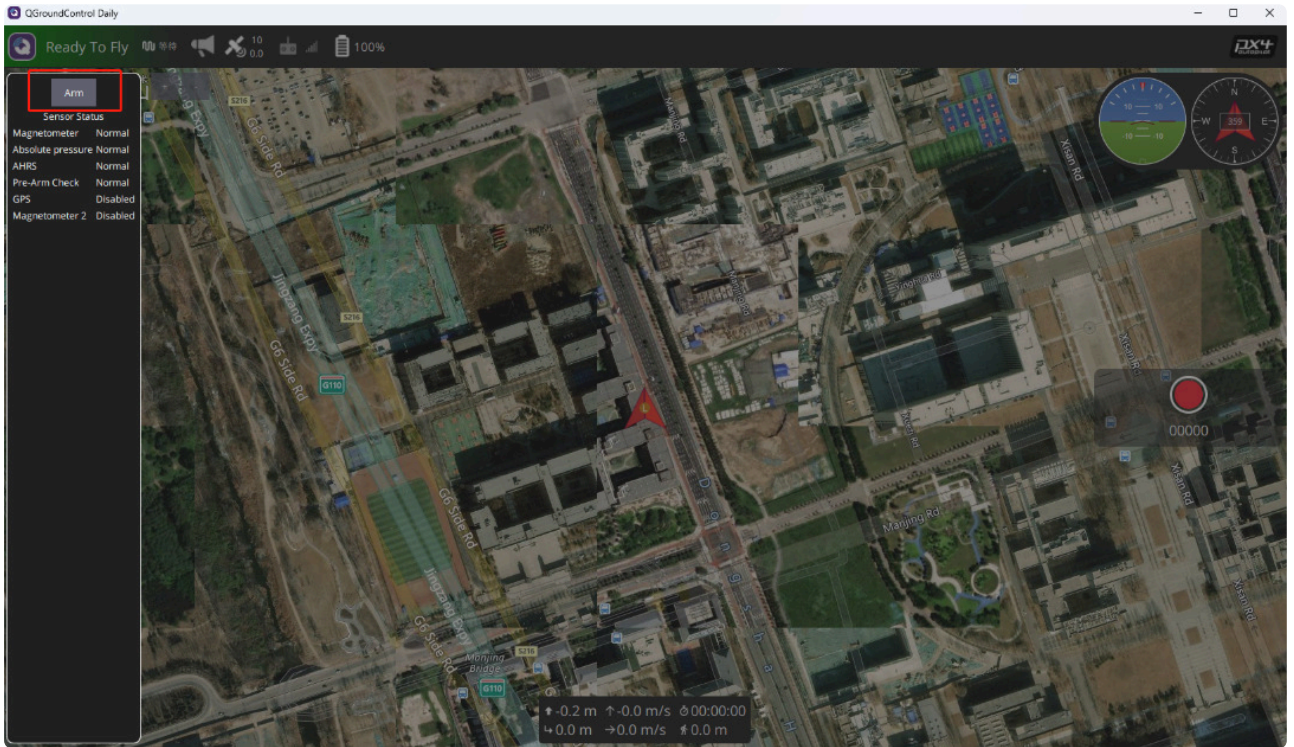
Step 2: 等待初始化完成

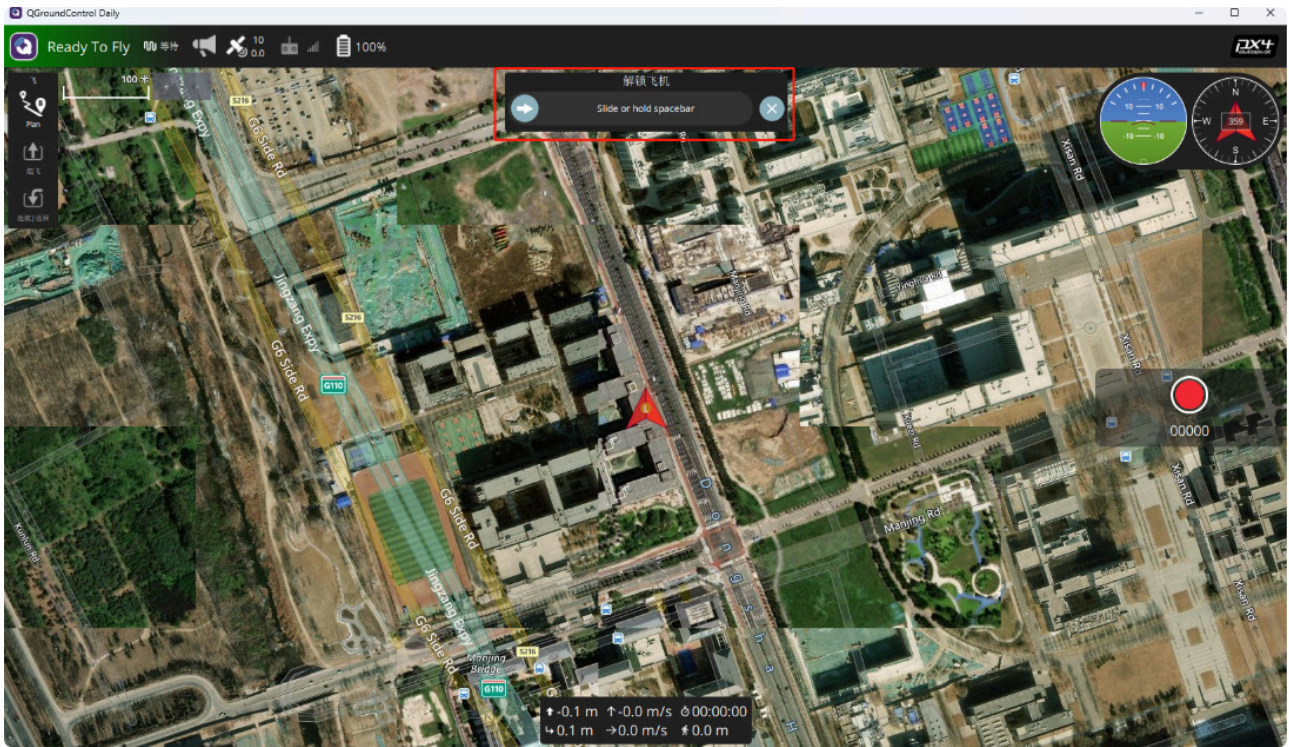
当RflySim3D显示“CopterSim/PX4 EKF 3D Fixed:1/1”，CopterSim显示“PX4: GPS 3D fixed & EKF initialization finished.”时，表明RflySim平台已完成初始化，可以进行软件在环仿真。



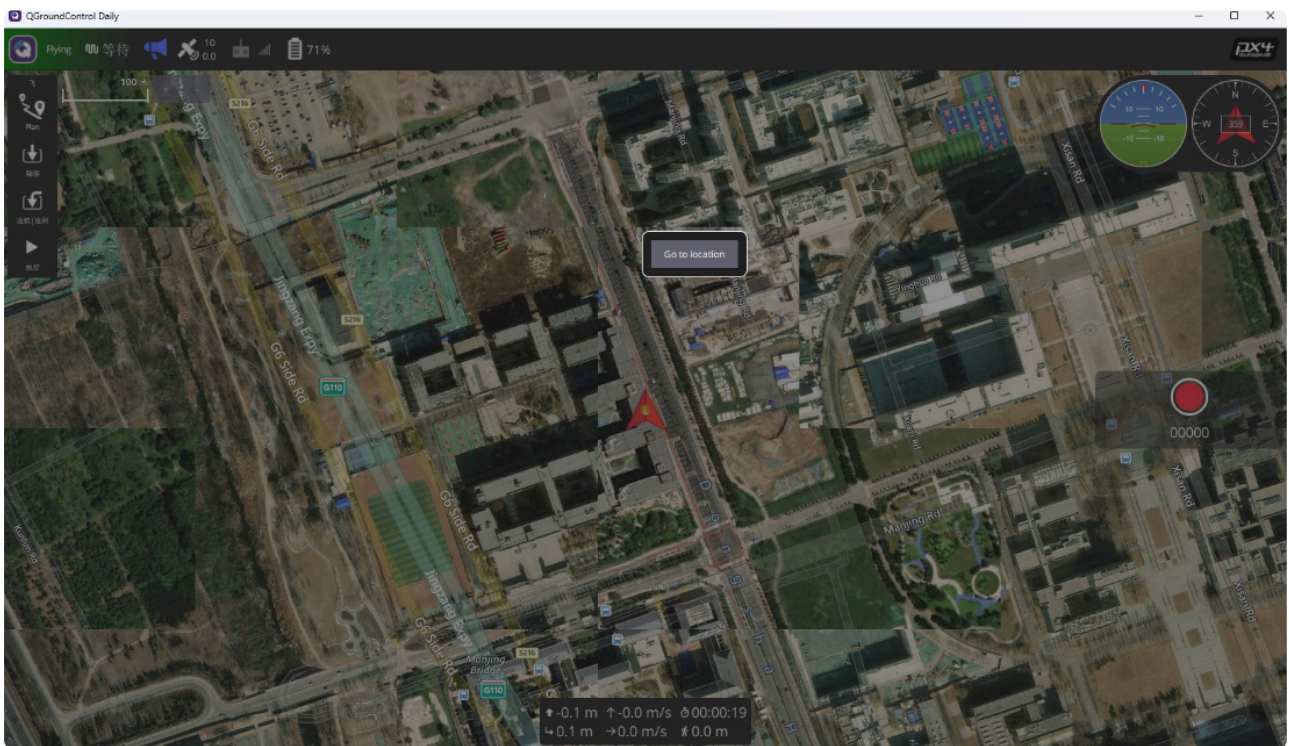
Step 3: 观察结果

1、在QGC界面左上角点击“Ready To Fly”处进行解锁，最后滑动QGC上方解锁飞机进度条。





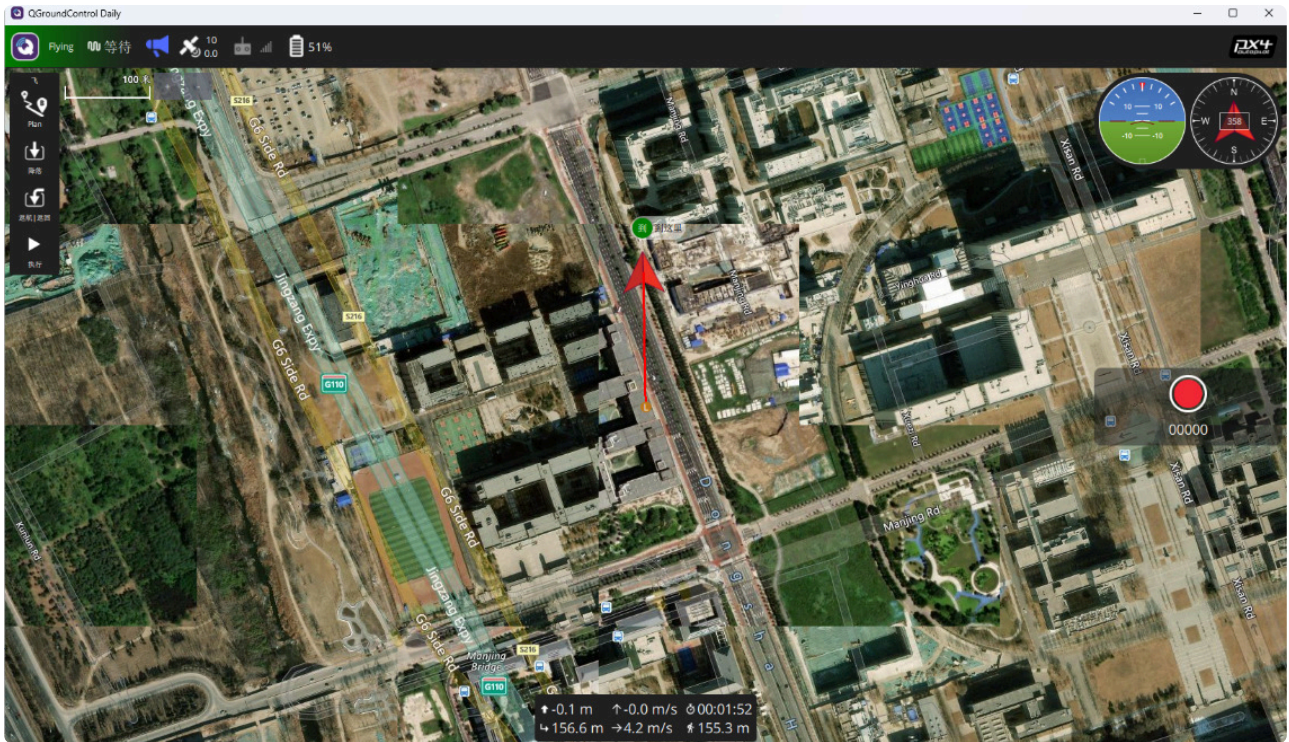
2、在QGC中点击地图后确认“Go to location”，滑动上方“将飞机移动到指定位置”完成目标位置确定。





3、在UE4与QGC中观察无人车的运动状态与运动轨迹。





5.3. 选做实验：硬件在环仿真












Step 1: 连接飞控

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）。

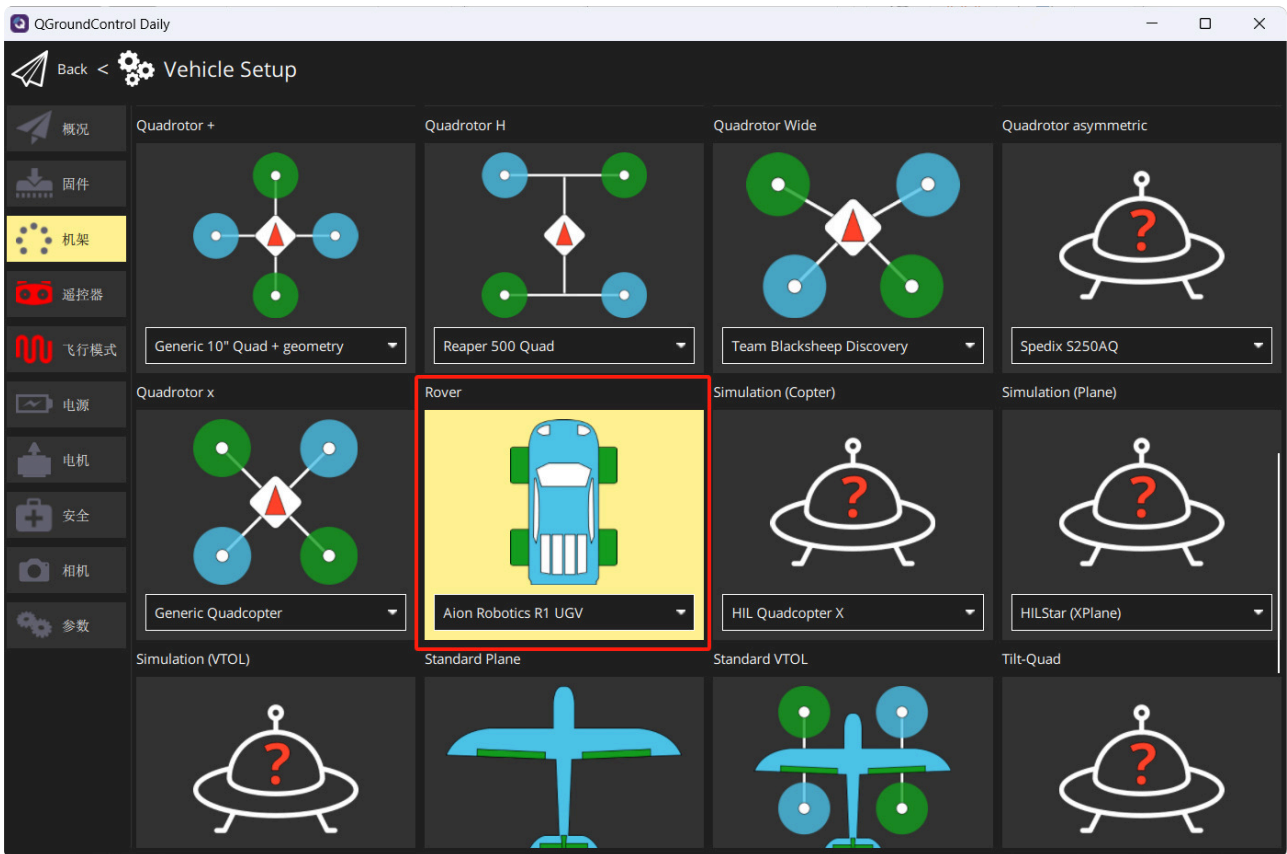


Step 2: 设置硬件在环机架

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

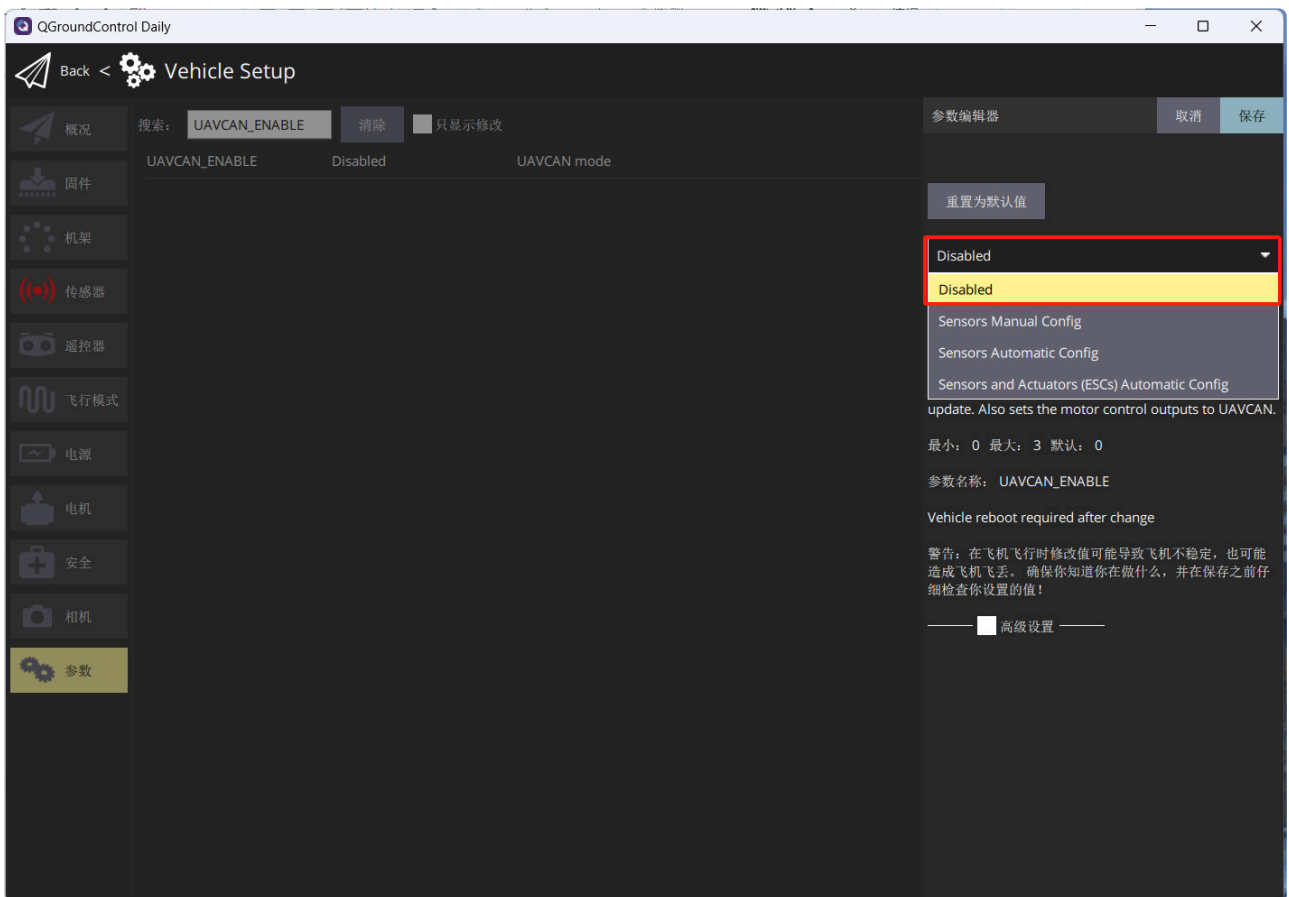
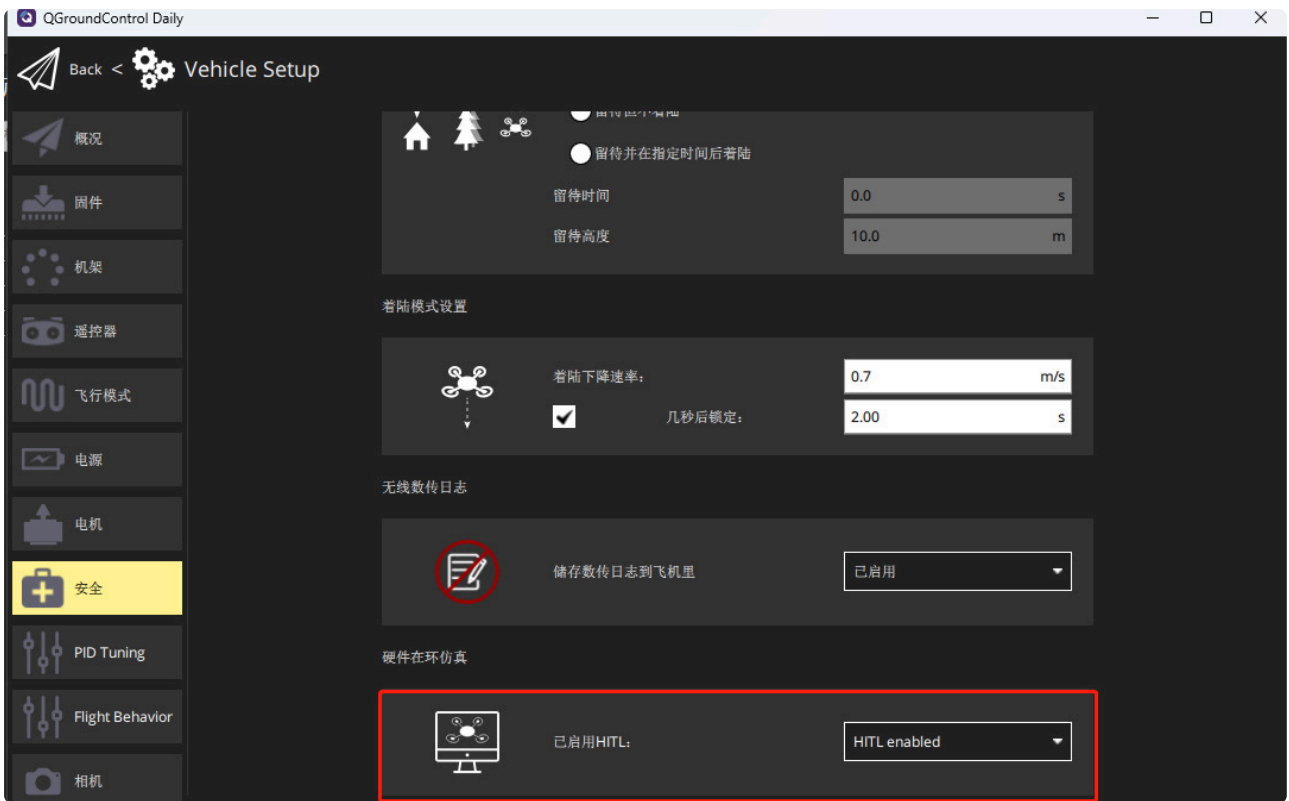
机架设置为“Aion Robotics R1 UGV”，点击QGC右上角的“应用并重启”。

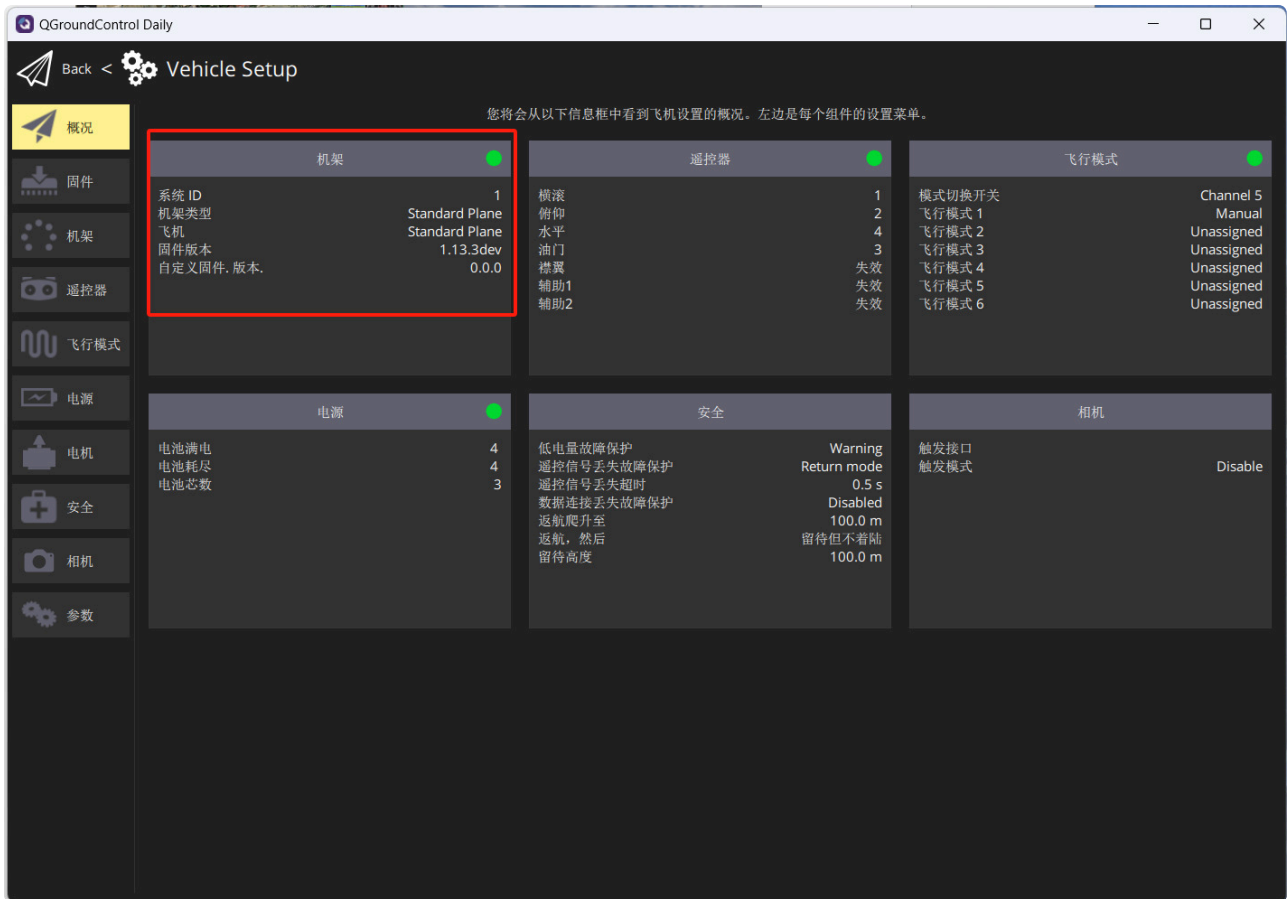


Step 3: 配置硬件在环参数

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。

注：使用1.13版本固件，还需要点击“参数”，在搜索栏中输入“UAVCAN_ENABLE”，在弹出框中设置为“Disabled”，保存。





Step 4: 启动仿真

右键以管理员身份运行“[CarR1Diff_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，启动一辆无人车的硬件在环仿真。

CarR1Diff_ert_rtw	2023/11/7 14:43	文件夹	
slprj	2023/11/7 14:32	文件夹	
CarR1Diff.dll	2023/11/7 14:43	应用程序扩展	224 KB
CarR1Diff.slx	2023/11/7 15:05	Simulink Model	77 KB
CarR1Diff_HITLRun.bat	2023/11/7 14:37	Windows 批处理...	6 KB
CarR1Diff_init.m	2023/10/24 15:33	Objective C 源文件	4 KB
CarR1Diff_SITLRun.bat	2023/11/7 14:37	Windows 批处理...	6 KB
GenerateModelDLLFile.p	2023/10/24 15:33	MATLAB.p.9.14.0	6 KB
MavLinkStruct.mat	2023/10/24 15:33	MATLAB Data	5 KB
MulticopterModel.zip	2023/11/7 14:43	压缩(zipped)文件...	99 KB
Readme.docx	2023/11/7 15:09	Microsoft Word ...	14,358 KB
Readme.pdf	2023/10/24 15:33	Foxit PhantomP...	1,998 KB

```
C:\Windows\system32\cmd.e: X + v
已复制      1 个文件。

-----
Please input the Pixhawk COM port list for HITL
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|
```

注：在“[CarR1Diff_HITLRun.bat](#)”硬件在环的脚本文件中，同样需要设置对应无人车的DLL名：

```
REM Set use DLL model name or not, use number index or name string
REM This option is useful for simulation with other types of vehicles instead of multicopters
set DLLModel=CarR1Diff
```

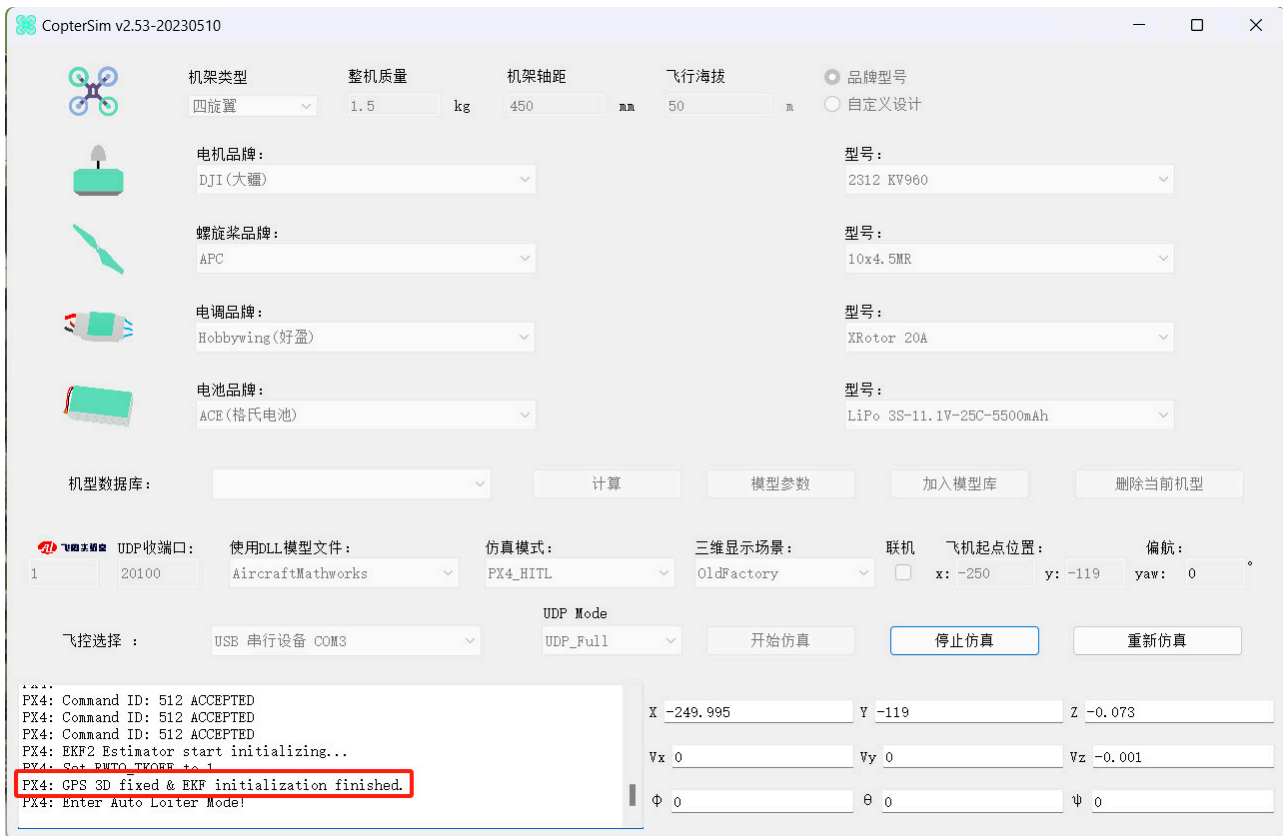
在SimMode处选择CopterSim中对应的硬件在环仿真模式：

```
REM Set the simulation mode on CopterSim, use number index or name string
REM e.g., SimMode=0 equals to SimMode=PX4_HITL
set SimMode=0
```

与软件在环仿真不同的是，在之前的配置准备环节中已经在QGC中设置了对应机架，所以在该脚本文件中不用设置机架。

Step 5: 等待初始化完成

等待 CopterSim 中显示连接上 RflySim3D，完成初始化。



Step 6: 仿真过程

之后测试步骤与软件在环的相同，运行之后观察差动无人车能否准确响应指令。

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. 机器人常见的差速轮模型与阿克曼模型运动学方程解析 - 古月居 (guyuehome.com)。

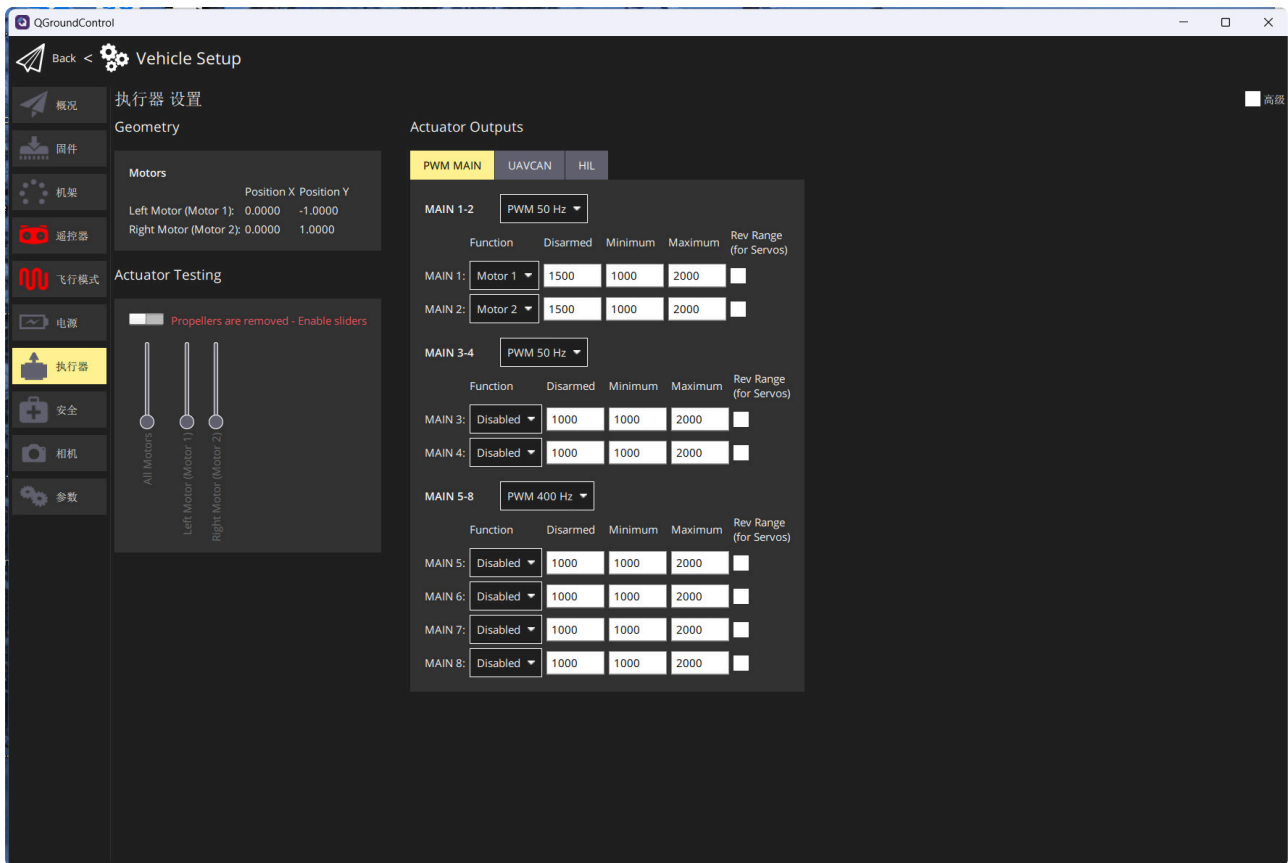
7. 常见问题

1. PX41.14版本固件全面启用了动态混控的规则，体现在使用1.14版本固件进行软硬件在环仿真时，QGC车辆设置页面会新增“执行器”页面，具体内容见

[控制分配](#)

[\(混控\) | PX4 自动驾驶用户指南](#)

[\(main\)](#)



控制分配 (混控)

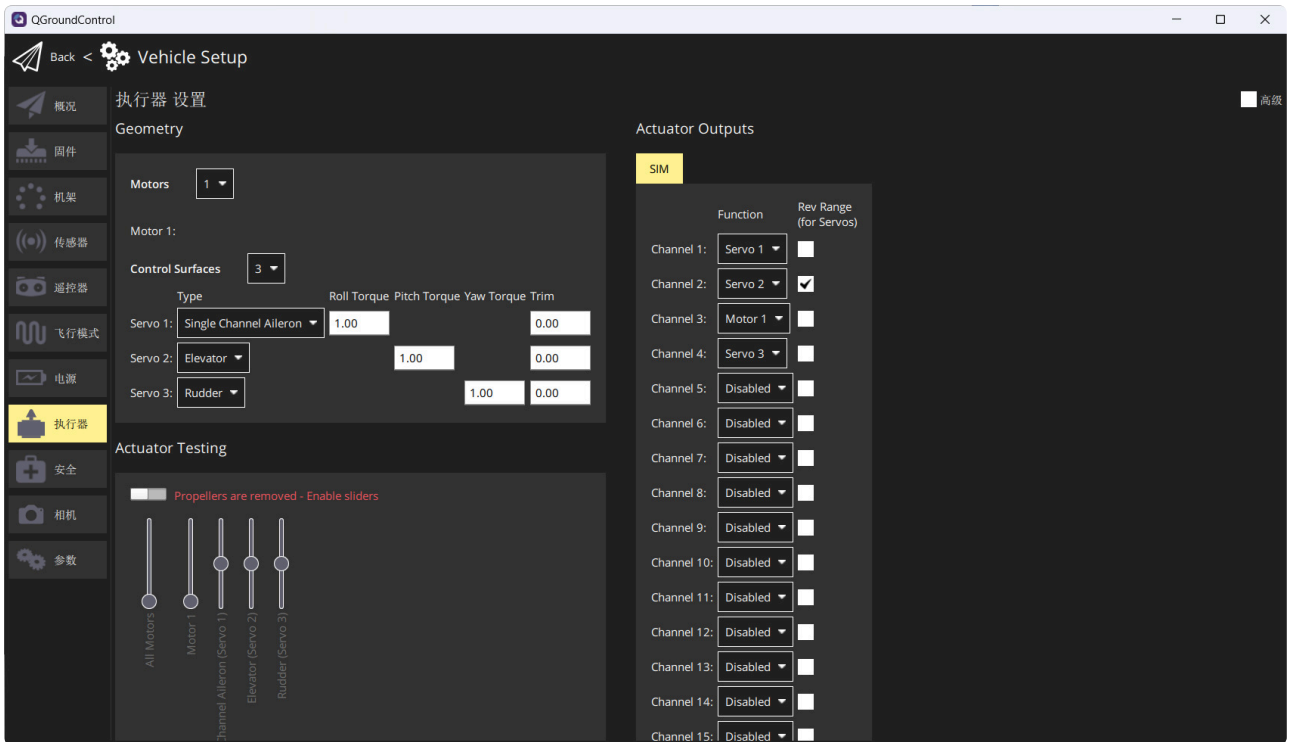
注解

控制分配取代了在 PX4 v1.13 中使用的旧的混控方法。PX4 v1.13 文档见：[混控& 驱动器](#)，[构型文件](#) 和 [添加一个新的机型配置](#)。

如果需要以1.14版本固件进行固定翼模型软硬件在环仿真，那么需要进行以下设置：

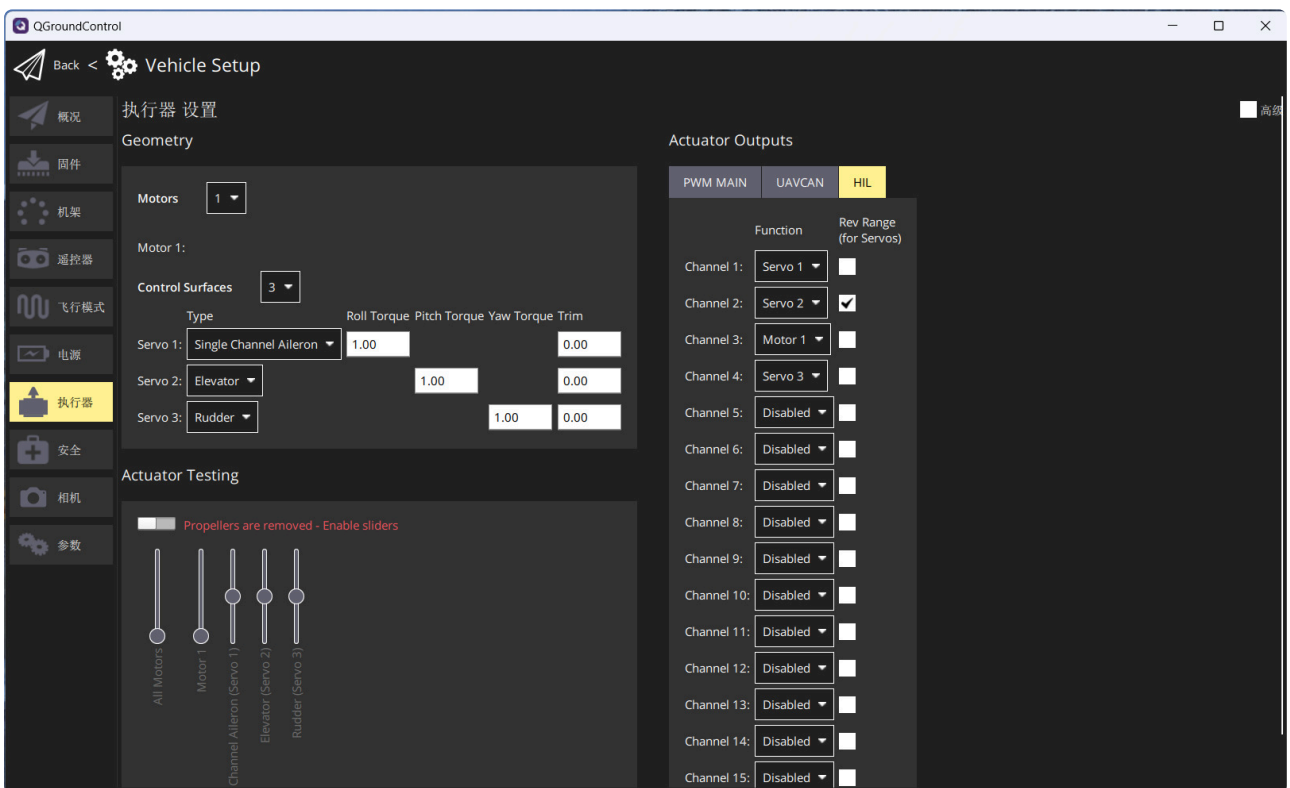
1. 软件在环仿真：

通过AircraftMathworks_SITLRun.bat启动仿真并完成初始化后，在QGC执行器页面中按如下图所示完成设置，即可正常仿真。

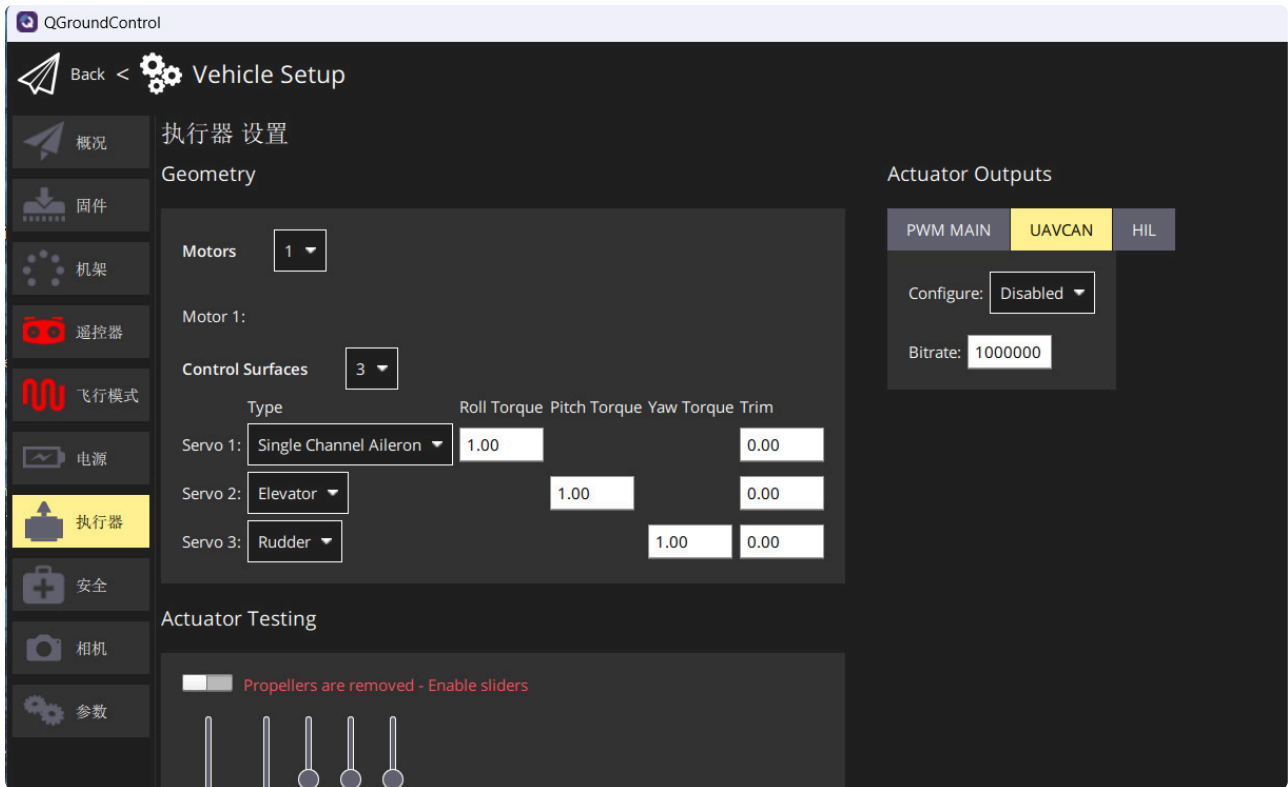


2) 硬件在环仿真:

除了硬件在环通常要设置的选项，还需要在QGC执行器页面中按如下图所示完成Actuator Outputs HIL设置:



并且需要设置UAVCAN为Disabled，完成设置后，重新插拔飞控即可正常仿真。



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