

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阿克曼底盘无人车模型介绍

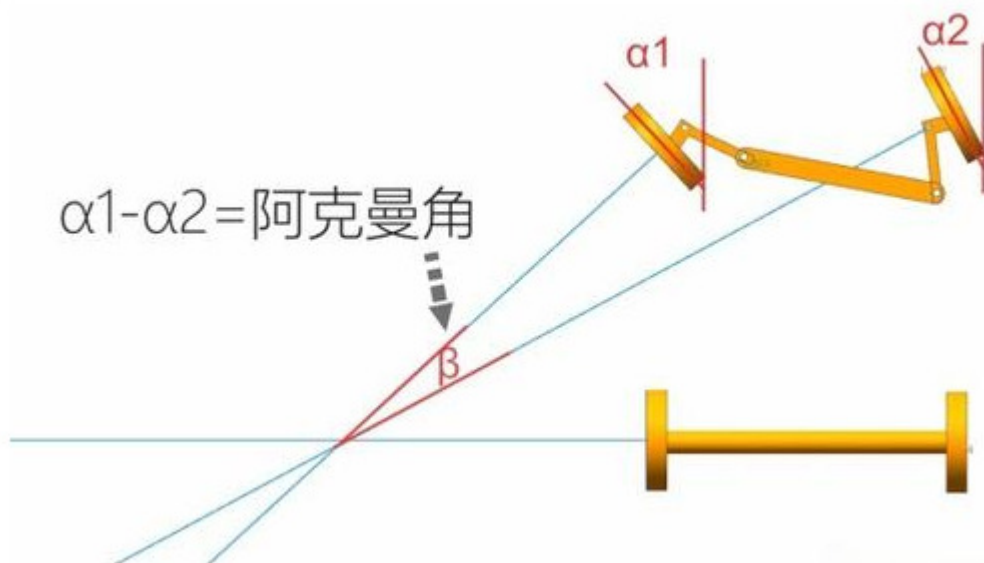
1.2 实验目的

将Simulink文件编译生成阿克曼底盘无人车的DLL模型文件；并对生成的阿克曼底盘无人车模型在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阿克曼底盘无人车模型的使用。

1.3 关键知识点

阿克曼转向（Ackermann Steering）是由英国机械师乔治·艾克曼在1817年发明的。这一概念离我们并不遥远。当前世界的汽车，使用的就是阿克曼转向模型。

阿克曼模型中很关键的概念就是阿克曼角。如果没有阿克曼角，车辆转弯时两个转向轮的转角就会相等，此时外侧车轮和内侧车轮会各自绕着不同的圆心运动，反馈到驾驶员处的感觉就是车辆有点拐不过来。如下图。



在此刻，前轮和后轮会呈现一个稳固的三角形，这时车辆别说转弯了，能不能动都是问题

基本原理：

在汽车转弯的时候，内外轮转过的角度不一样，内侧轮胎转弯半径小于外侧轮胎，内侧车轮的转向角度大于外侧车轮，使得车辆能够围绕一个圆心进行转弯。这个角度差就是“阿克曼角”。这种转向方式能够使得车辆所有车轮垂线均能指向圆心，车身过弯姿态更加顺畅稳定，并且不会造成轮胎损耗。它描述了前轮转向角度、车轮转角、转向杆长度和转向半径之间的关系。

实际实现：

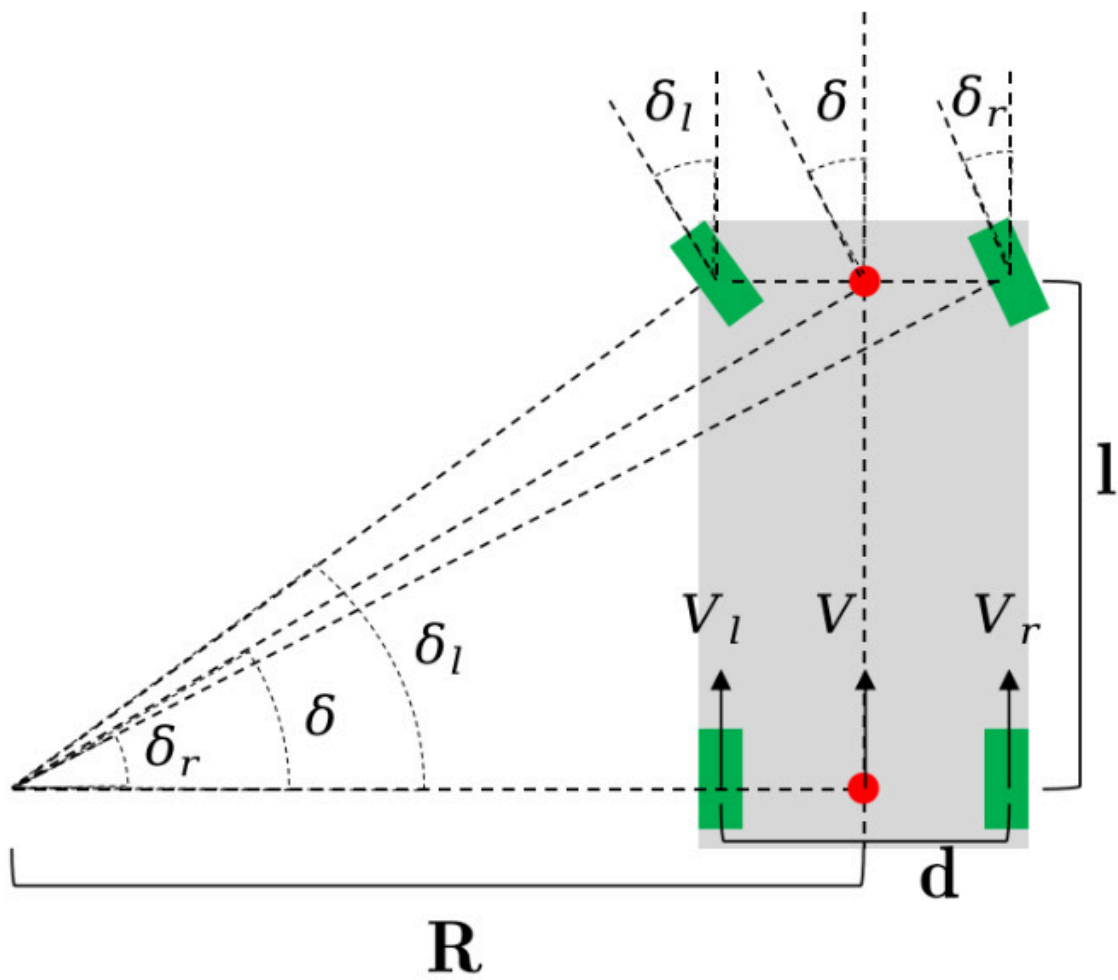
车辆在转弯时驱动轮应该有不同的轮速。利用阿克曼模型的汽车的驱动方式分为：前驱、后驱和四驱。利用“差速器”这样一个机械结构，将发动机的转速自动分配到两个驱动轮上，使两个驱动轮的速度不同并满足转弯需求。前驱的差速器显然在前轮上，后驱的差速器在后轮上，四驱的差速器在前后轮均有。

■ 载具的基本动力学特性

CarAckerman.slx是基于最小系统模版建立的阿克曼底盘无人车动力学模型。

■ 阿克曼车的运动学模型

前驱和后驱的运行学分析大体相同，这里以一个后驱车辆为例进行分析。（补充知识：正向运动学指的是机器人控制空间描述计算出末端位姿，逆向运动学指的是知道机器人末端位姿反推各个控制量。在机械臂、移动机器人中这些控制量指的都是不同器件，这里不再赘述。）



我们需要知道前轮的转角，根据图我们可以得出如下关系

$$\tan \delta = \frac{l}{R}$$

而车辆转弯半径

$$R$$

实际上就是由车速和车角速度决定的

$$R = \frac{v_{car}}{\omega_{car}}$$

因此前轮的角度值为

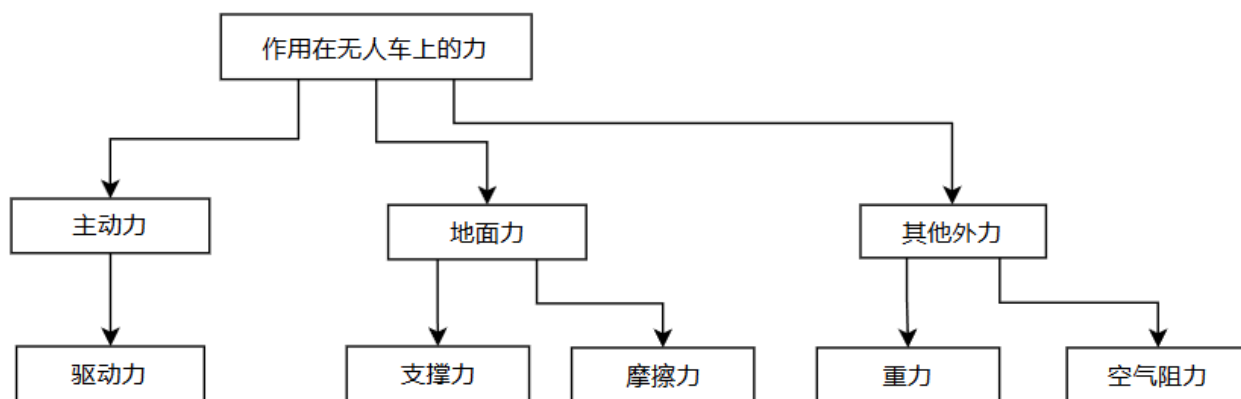
$$\delta = \arctan\left(l * \frac{v_{car}}{\omega_{car}}\right)$$

这样的阿克曼模型可以进一步简化为一个自行车模型：前面一个轮子提供旋转，后面一个轮子提供速度。

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 [\(v1.12\)](#)

本例程中无人车的机架在

`\PX4PSP\Firmware\ROMFS\px4fmw_common\init.d\airframes\50000_generic_ground_vehicle`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`中的部分代码如下：

```
set PWM_OUT 1234
```

设置混控器（mixer）为`rover_generic`

set MIXER rover_generic

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\
rover_generic.main.mix其中只用到了PWM2和PWM4；PWM1，PWM3为空混控器。
PWM2输出偏航到伺服电机来转向，PWM4输出油门给主动轮电机来变速。

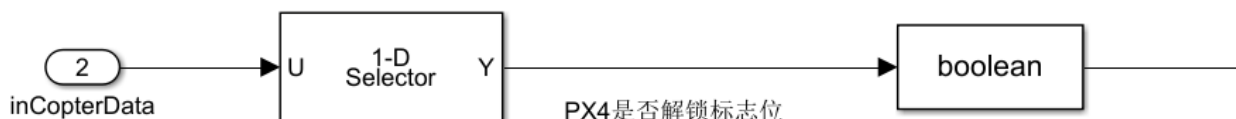
载具模型的整体输入输出和参数

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	int16(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]

参数名	参数	值
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]
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)

2. 实验效果

实现阿克曼底盘无人车DLL模型文件生成，以及完成阿克曼底盘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\e3_CarAckermanModeCtrl

文件夹/文件名称	说明
CarAckerman.slx	阿克曼底盘无人车模型文件。
CarAckerman_HITLRun.bat	硬件在环仿真批处理文件。
CarAckerman_SITLRun.bat	软件在环仿真批处理文件。
GenerateModelDLLFile.p	DLL格式转化文件。
CarAckerman_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

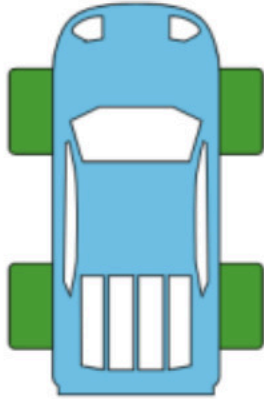
对应了差速小车（控制左右两侧轮子，进行差速转弯，可原地转弯后退），CarAckerman对应了阿克曼底盘小车（控制后轮扭矩+方向盘，不能原地转弯，有转弯半径）

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 data-bbox="767 808 1155 853">SYS_AUTOSTART = 50000</p> <p data-bbox="767 904 1050 949">Specific Outputs:</p> <ul data-bbox="767 999 1067 1095" style="list-style-type: none"><li data-bbox="767 999 1067 1043">• MAIN2: steering<li data-bbox="767 1055 1067 1095">• MAIN4: throttle
Aion Robotics R1 UGV	<p data-bbox="767 1189 1169 1234">Maintainer: Timothy Scott</p> <p data-bbox="767 1279 1155 1323">SYS_AUTOSTART = 50003</p> <p data-bbox="767 1375 1050 1420">Specific Outputs:</p> <ul data-bbox="767 1469 1283 1568" style="list-style-type: none"><li data-bbox="767 1469 1283 1514">• MAIN0: Speed of left wheels<li data-bbox="767 1525 1283 1568">• 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“CarAckerman.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](#)

CarAckerman_ert_rtw	2023/10/27 16:37	文件夹	
slprj	2023/10/27 15:46	文件夹	
CarAckerman.dll	2023/10/27 16:37	应用程序扩展	218 KB
CarAckerman.slx	2023/11/7 9:45	Simulink Model	68 KB
CarAckerman_HITLRun.bat	2023/10/27 16:38	Windows 批处理...	6 KB
CarAckerman_init.m	2023/10/27 16:36	Objective C 源文件	2 KB
CarAckerman_SITLRun.bat	2023/10/27 16:37	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	2023/10/27 16:37	压缩(zipped)文件...	94 KB
Readme.docx	2023/10/27 16:58	Microsoft Word ...	14,476 KB
Readme.pdf	2023/10/24 15:33	Foxit PhantomP...	2,082 KB

与多旋翼模型先比，小车模型改动如下：

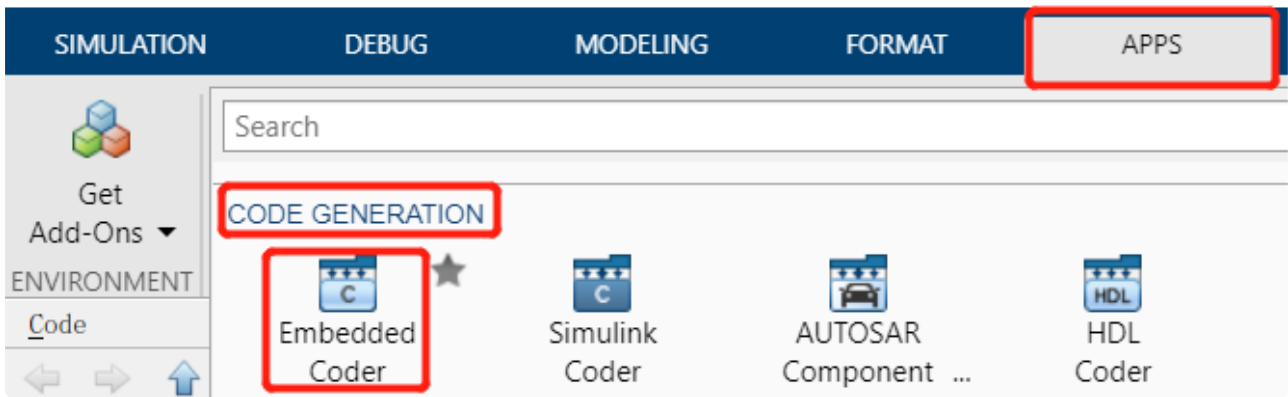
1) 在CarAckerman.slx的Motor Model中，设置了只响应2和4号PWM信号，分别给了方向舵和驱动力矩，遵循Generic Ground Vehicle机架的混控通道。
https://docs.px4.io/master/en/airframes/airframe_reference.html#rover。

2) 在CarAckerman.slx的Force and Moment Model/Ground Model模型中，修改了地面模型，响应四个轮子的摩擦力，来驱动车前进。

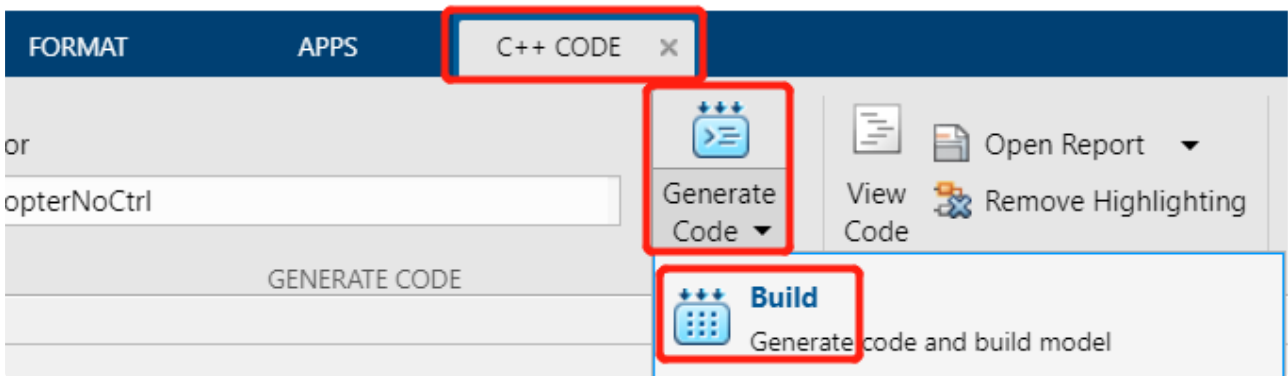
对于MATLAB 2019a及之前版本，工具栏样式见下图，直接点击它的编译按钮“Build”即可。



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。

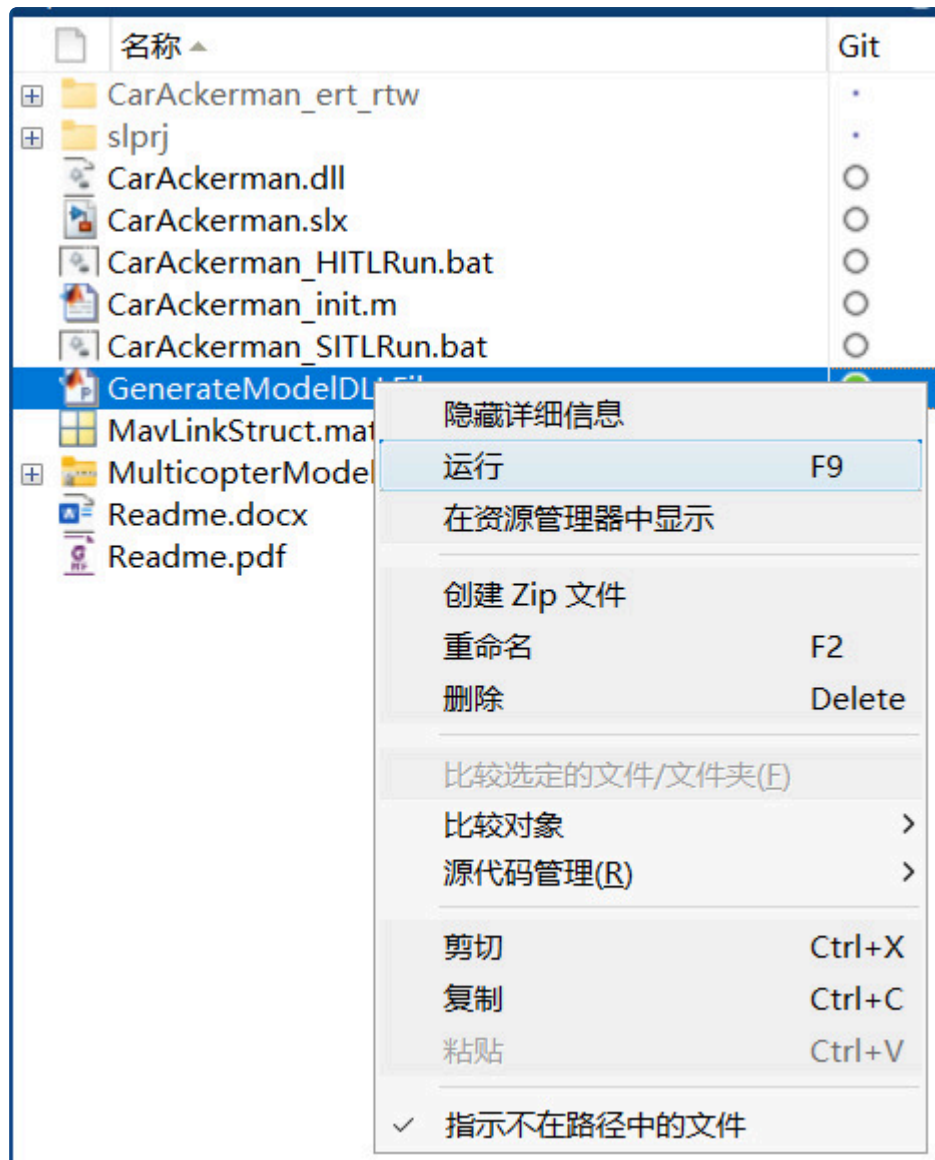


k



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“AircraftMathworks.dll”的DLL模型文件。



命令行窗口

```
fx >> GenerateModelDLLFile|
```

```
命令行窗口
版权所有 (C) Microsoft Corporation。保留所有权利。

modeldllgen.cpp
    cl modeldllgen.obj AircraftMathworks.obj /link /DLL /out:AircraftMathworks.dll
用于 x64 的 Microsoft (R) C/C++ 优化编译器 19.16.27051 版
版权所有 (C) Microsoft Corporation。保留所有权利。

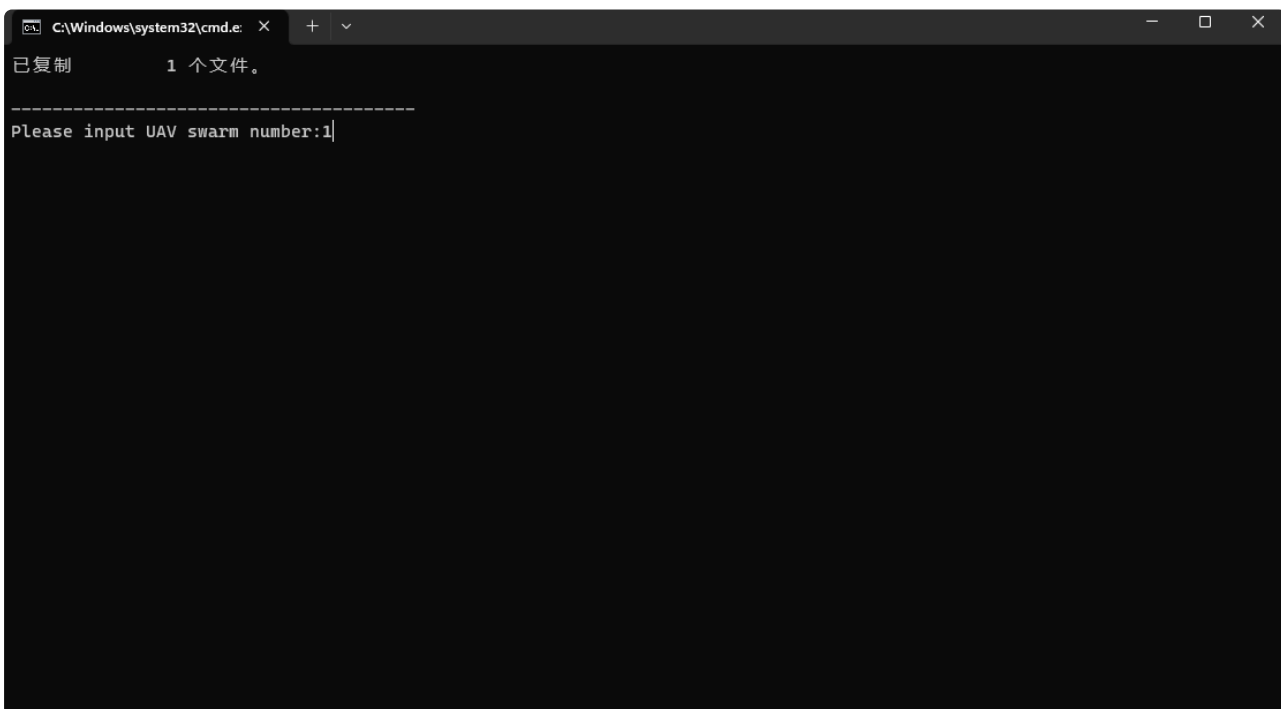
Microsoft (R) Incremental Linker Version 14.16.27051.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:modeldllgen.exe
/DLL
/out:AircraftMathworks.dll
modeldllgen.obj
AircraftMathworks.obj
    正在创建库 AircraftMathworks.lib 和对象 AircraftMathworks.exp
Compiling successfully, the AircraftMathworks.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“[CarAckerman_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“[CarAckerman_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=CarAckerman
```

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=generic_ground_vehicle
```

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

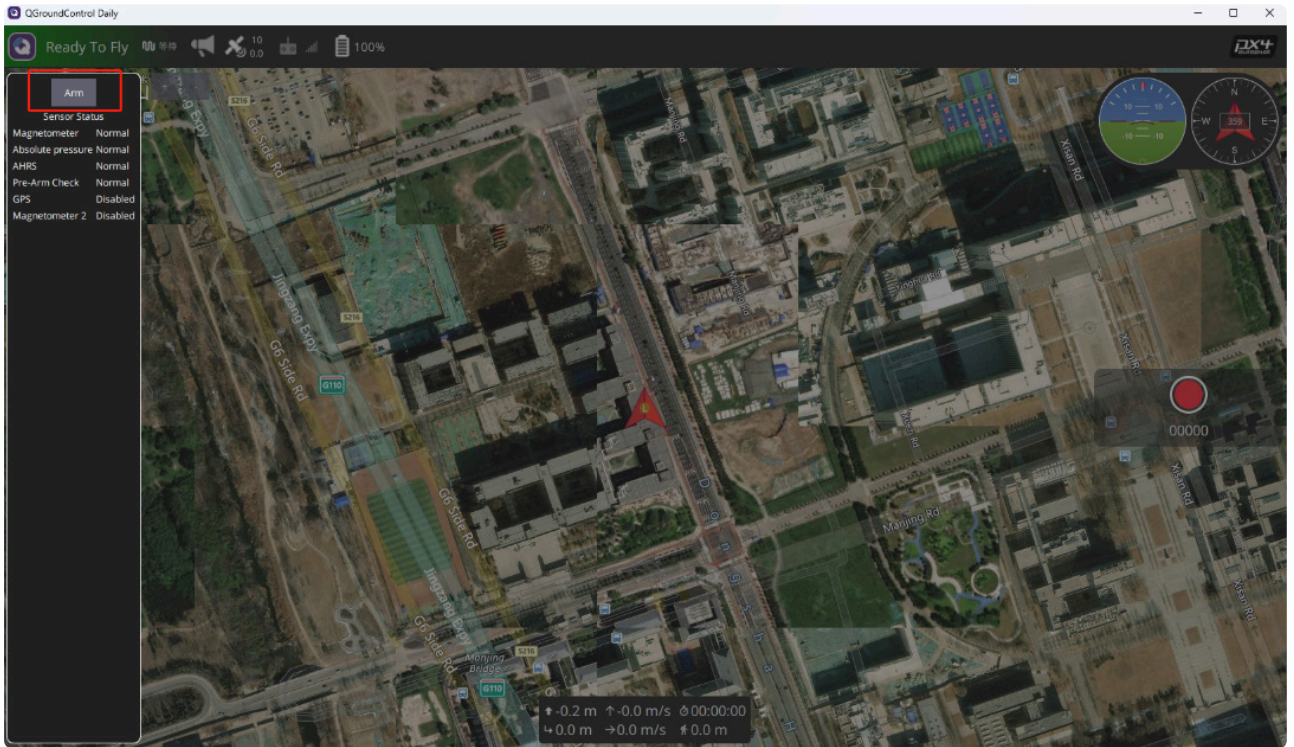
& EKF initialization

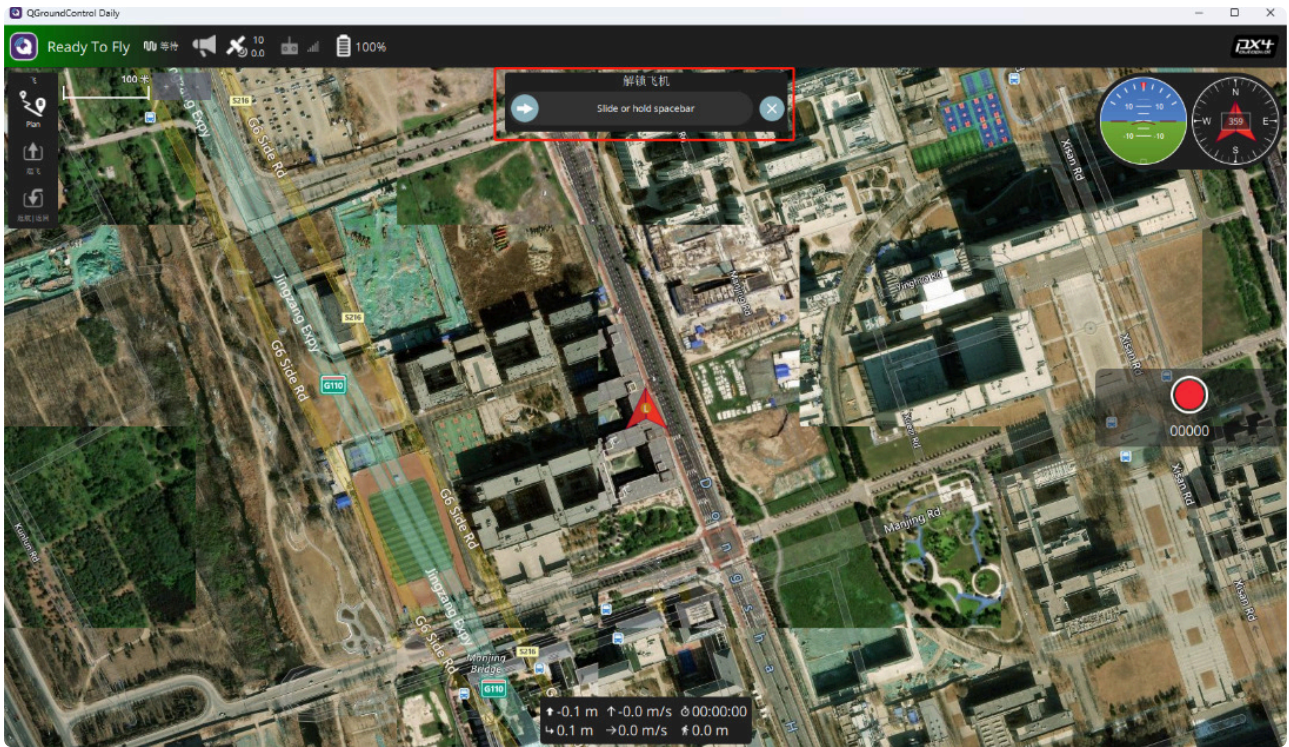
finished.”时，表明RflySim平台已完成初始化，可以进行软件在环仿真。



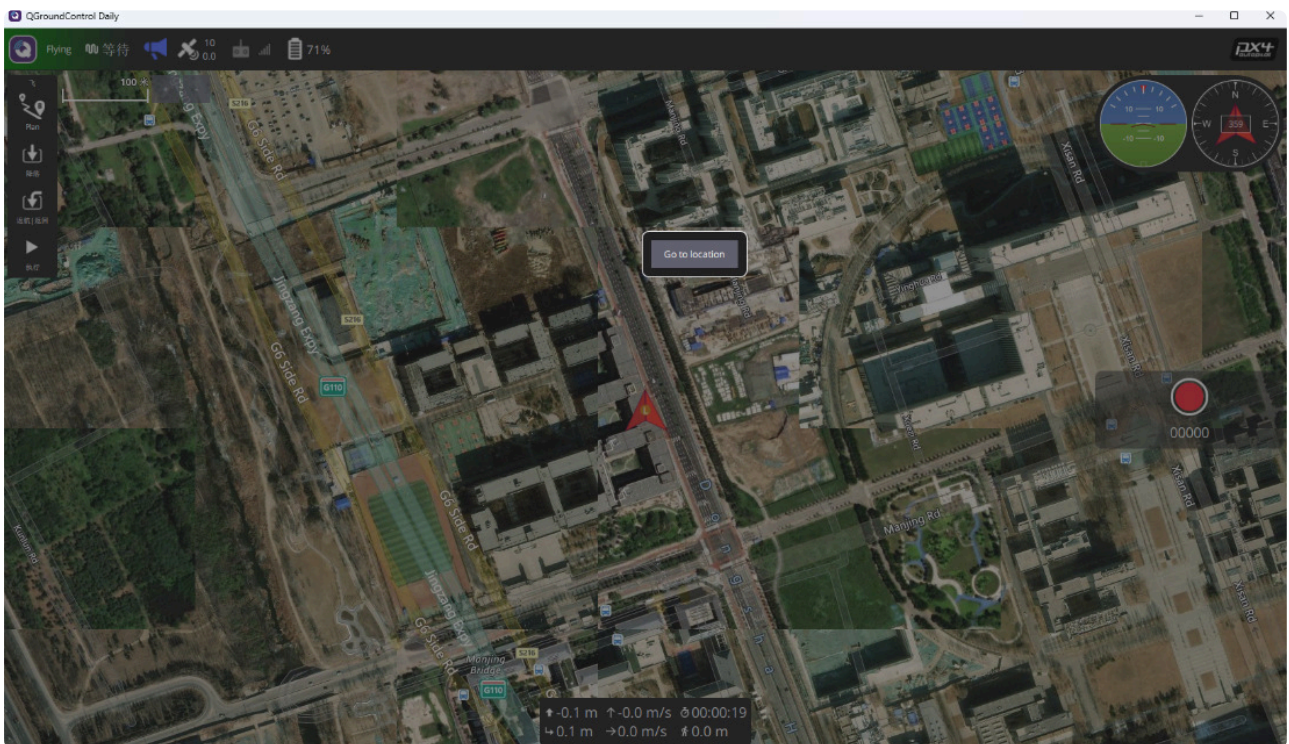
Step 3: 观察结果

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）。

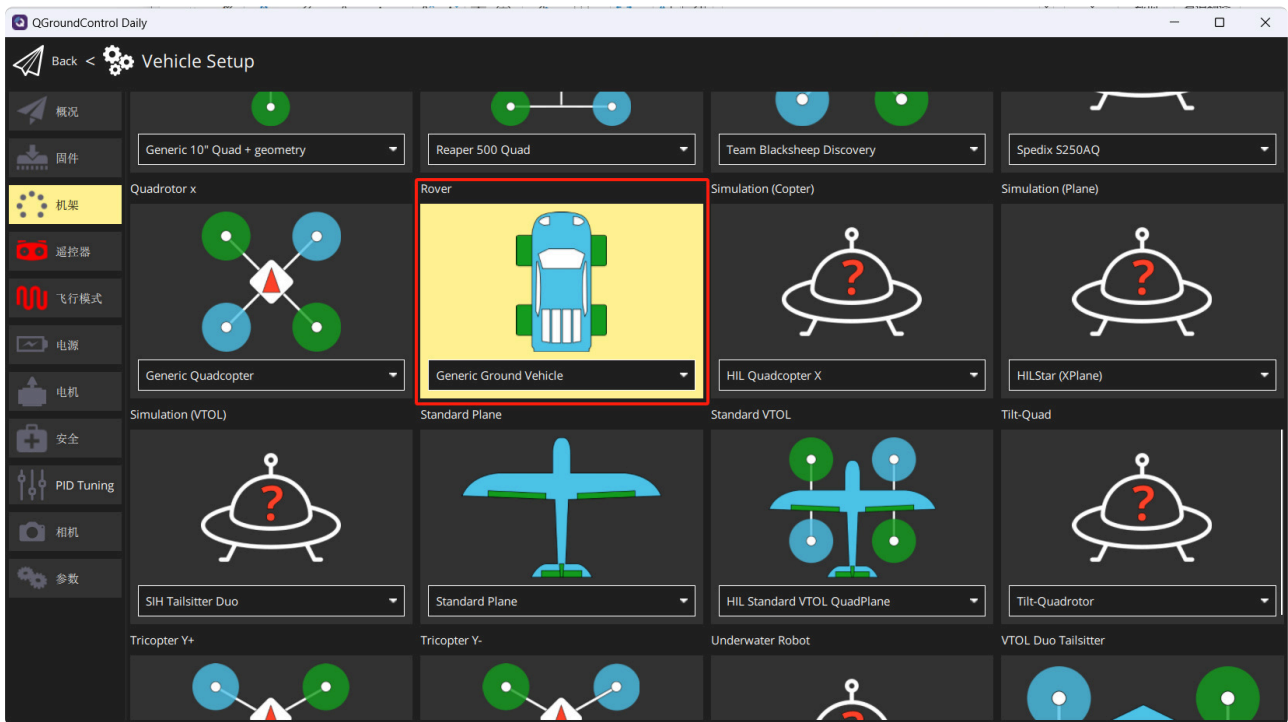


Step 2: 设置硬件在环机架

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

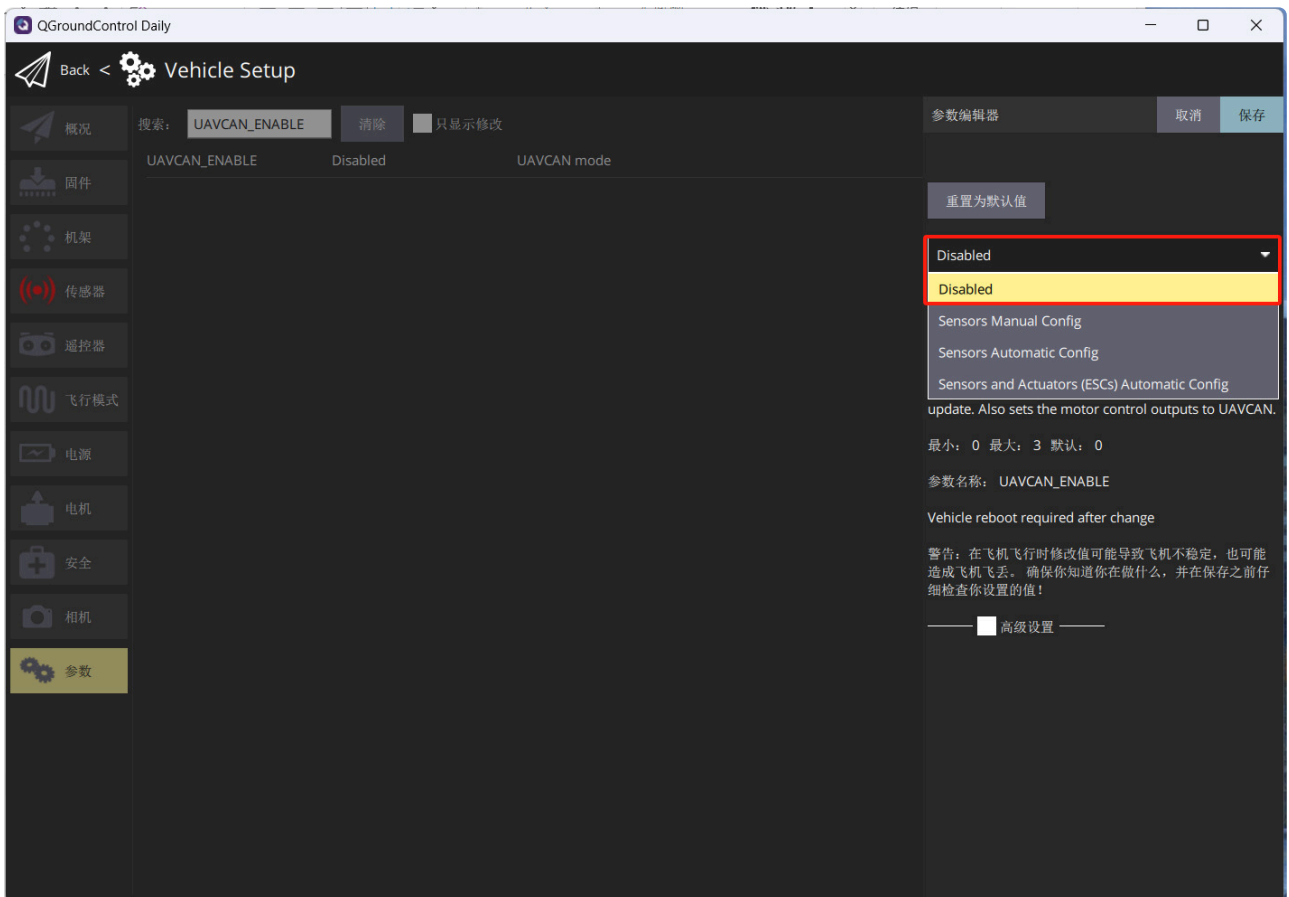
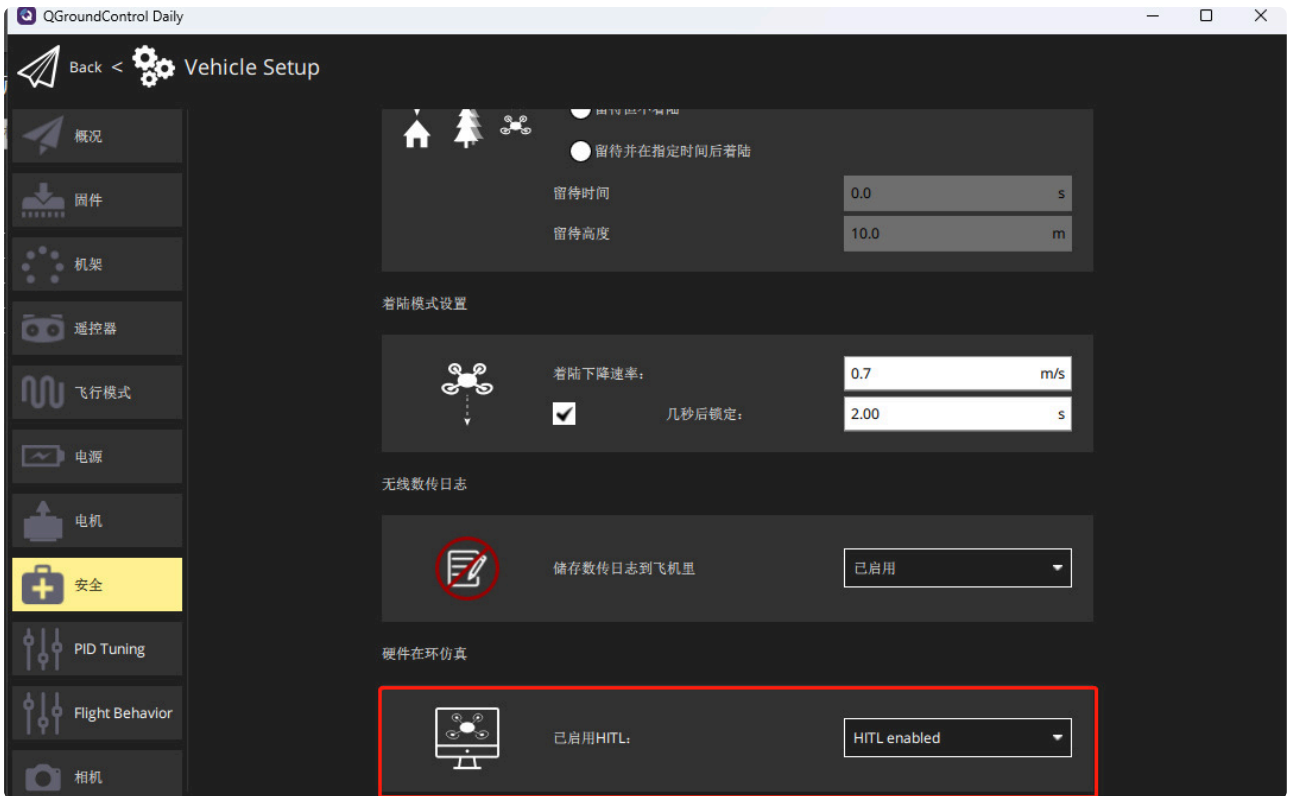
机架设置为“Generic Ground Vehicle”，点击QGC右上角的“应用并重启”。

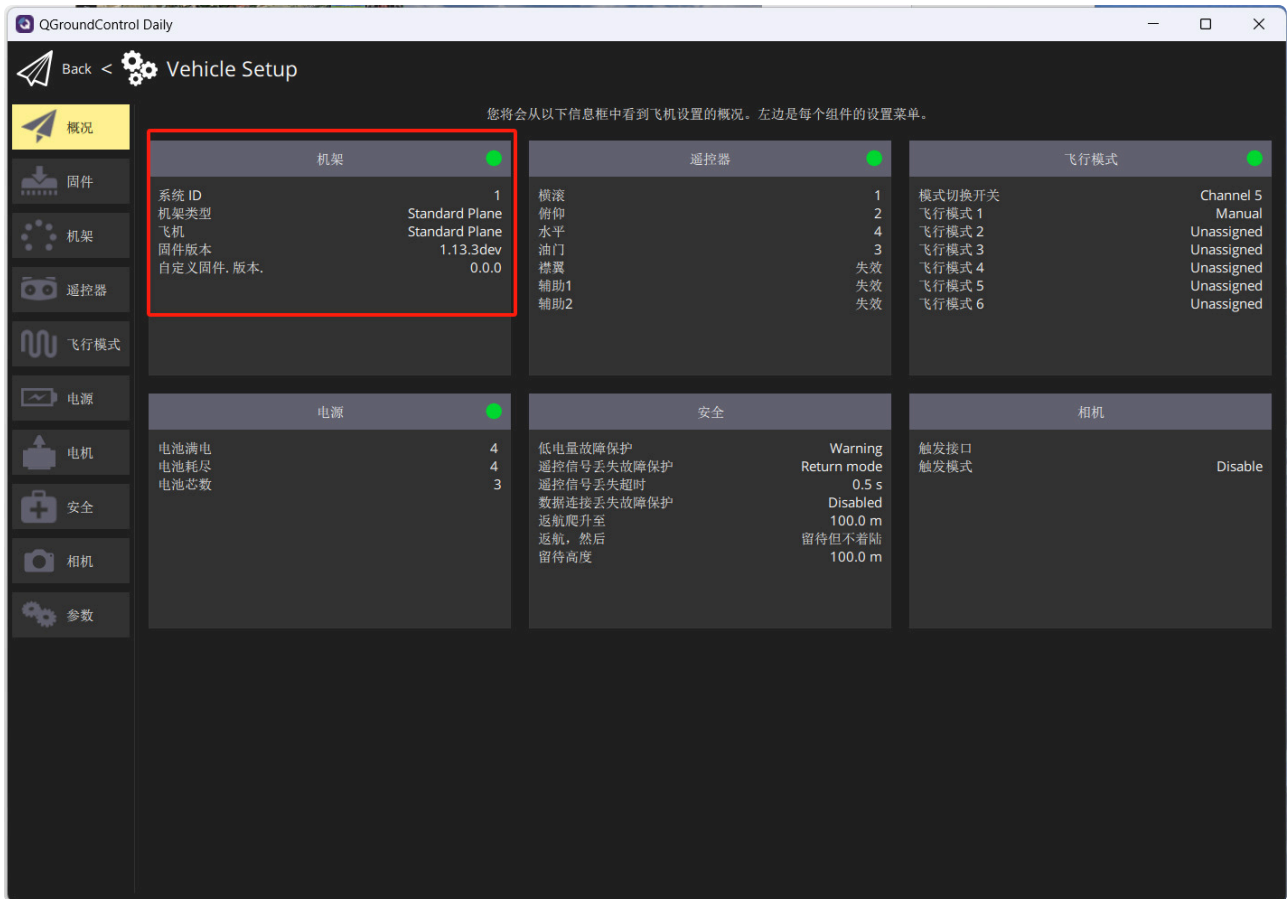


Step 3: 配置硬件在环参数

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“[CarAckerman_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。

CarAckerman_ert_rtw	2023/10/27 16:37	文件夹	
slprj	2023/10/27 15:46	文件夹	
CarAckerman.dll	2023/11/7 9:50	应用程序扩展	218 KB
CarAckerman.slx	2023/11/7 9:45	Simulink Model	68 KB
CarAckerman_HITLRun.bat	2023/10/27 16:38	Windows 批处理...	6 KB
CarAckerman_init.m	2023/10/27 16:36	Objective C 源文件	2 KB
CarAckerman_SITLRun.bat	2023/10/27 16:37	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	2023/10/27 16:37	压缩(zipped)文件...	94 KB
Readme.docx	2023/10/27 16:58	Microsoft Word ...	14,476 KB
Readme.pdf	2023/10/24 15:33	Foxit PhantomP...	2,082 KB

```
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

注：在“CarAckerman_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=CarAckerman
```

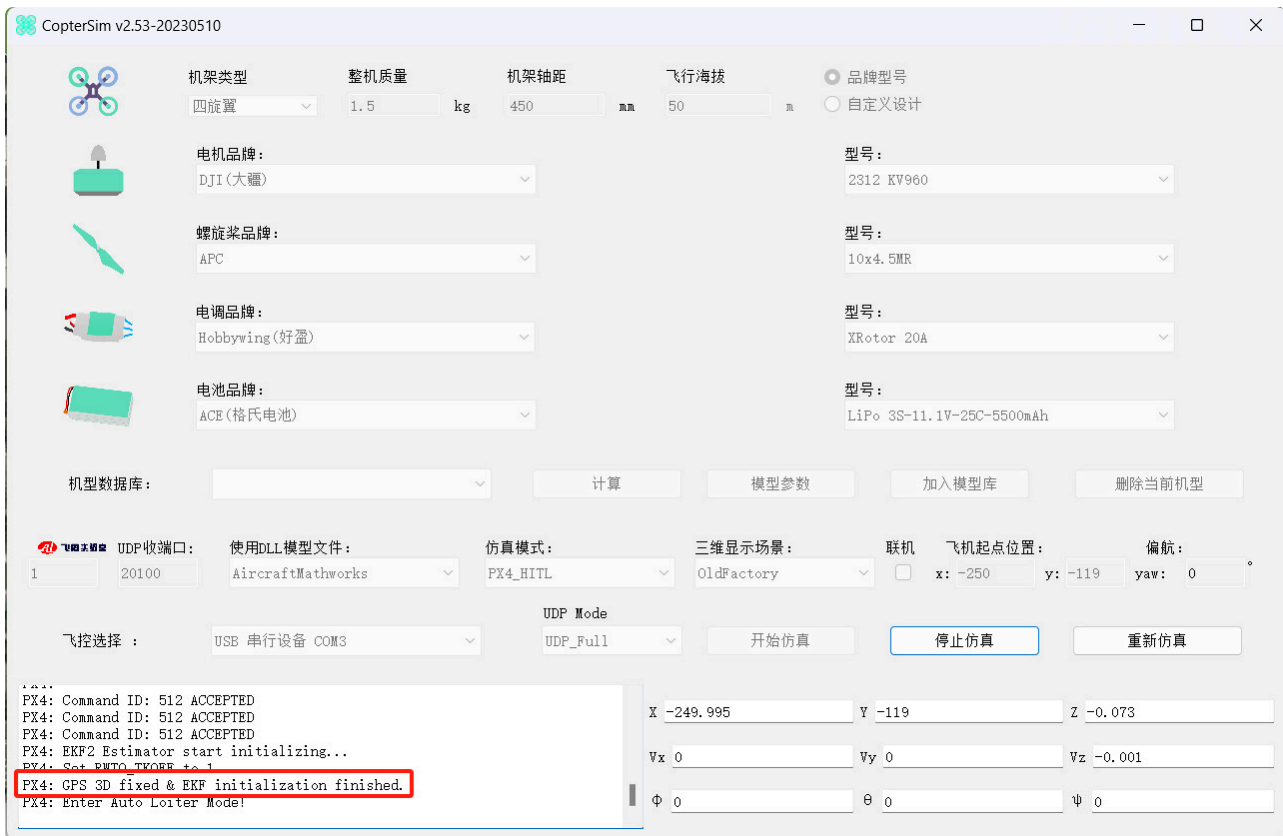
在SimMode处选择CopterSim中对应的硬件在环仿真模式：

```
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，运行之后观察阿克曼底盘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)。

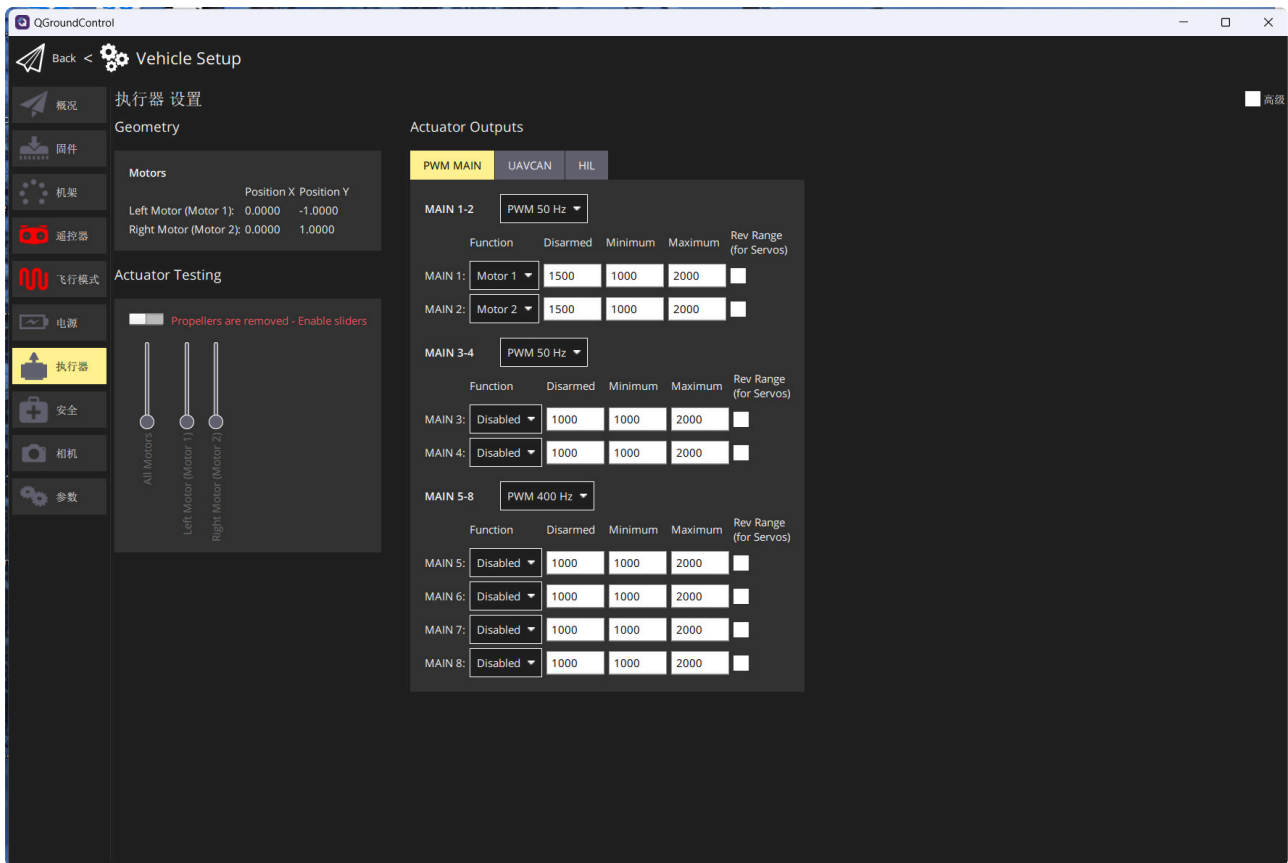
7. 常见问题

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)

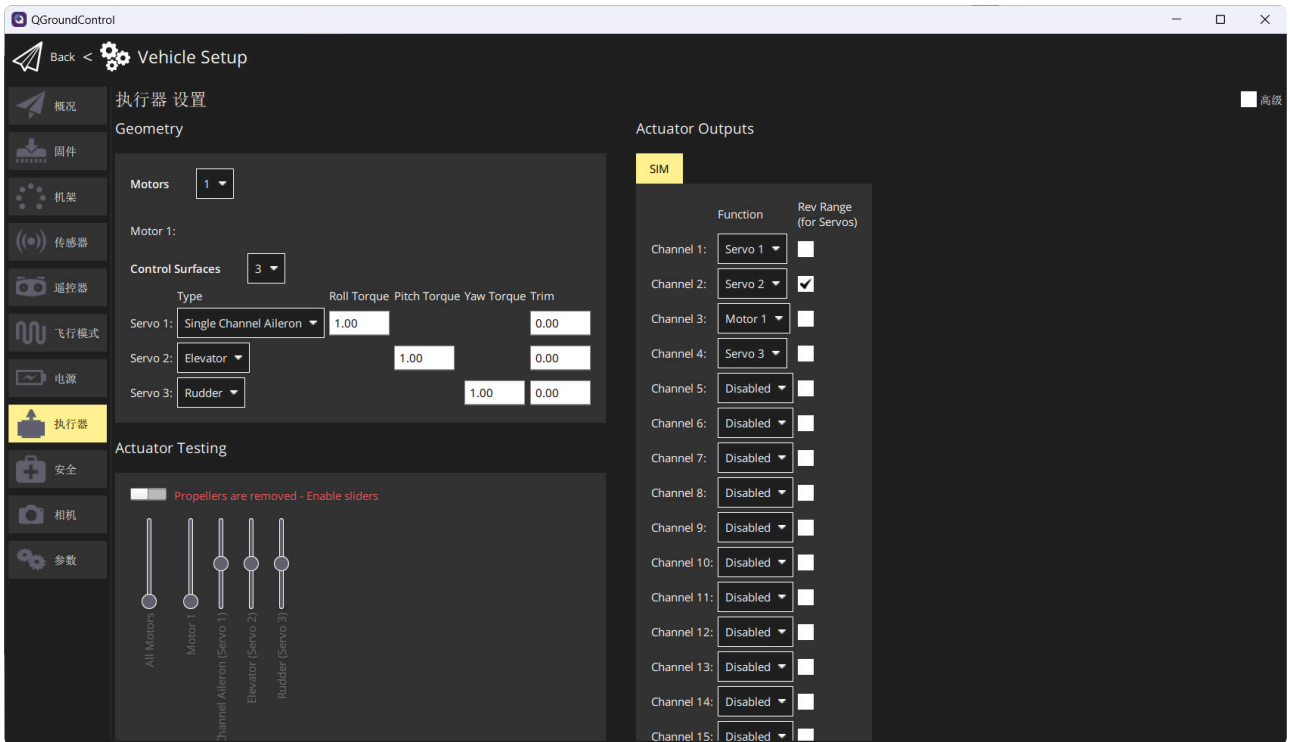
注解

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：

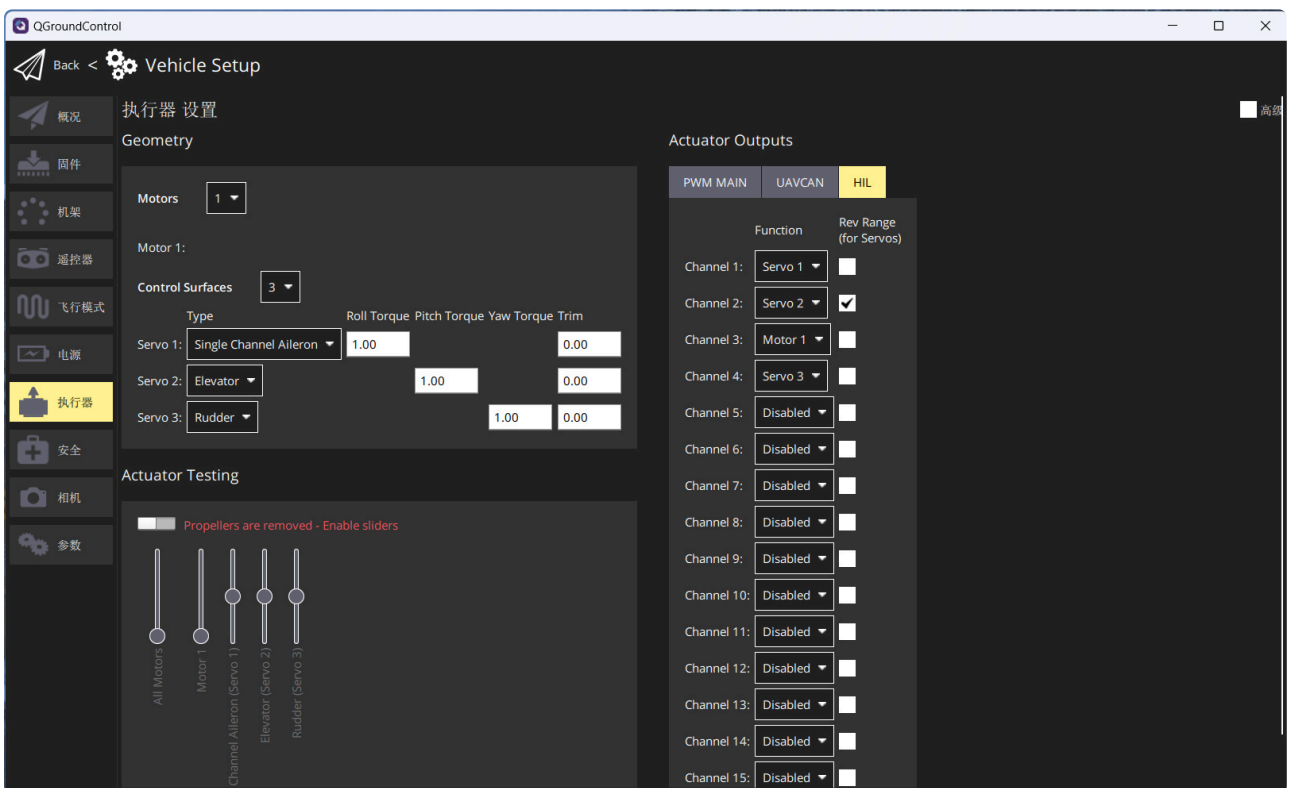
1. 软件在环仿真：

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。

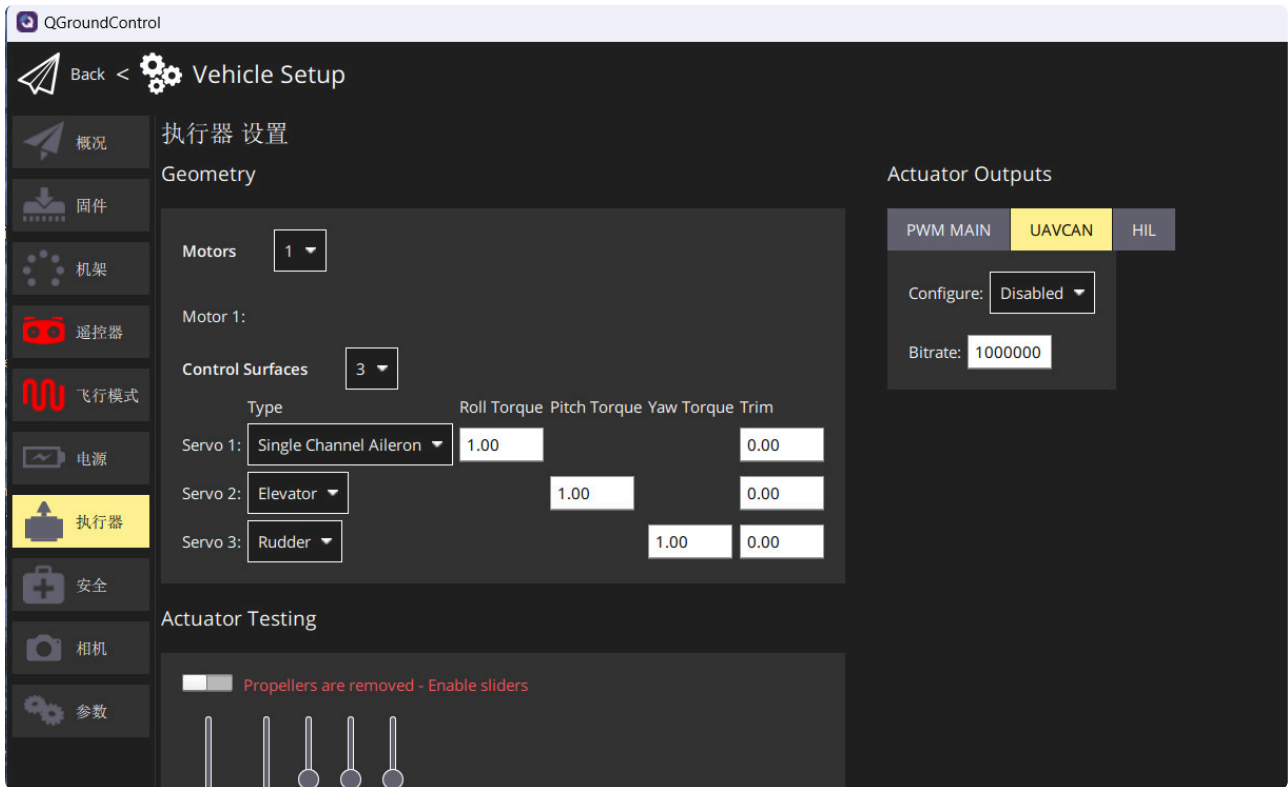


2) 硬件在环仿真:

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