

# 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（python差速控制）

## 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软件在环仿真测试，通过本例程熟悉平台差动无人车模型的使用

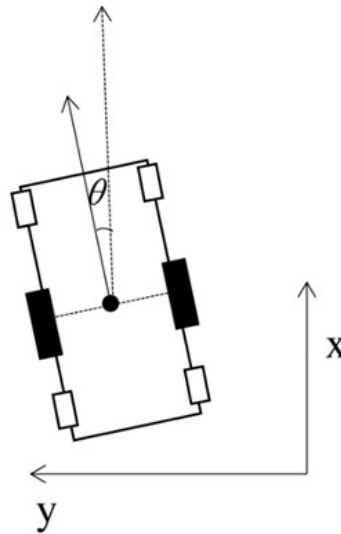
## 1.3 关键知识点

CarR1Diff.slx是基于最小系统模版建立的差动小车动力学模型。

### 载具的基本动力学特性

#### 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  $[x, y, \theta]$ ，代表机器人目前的位置是  $[x, y]$ ，车头朝向  $\theta$  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

$$p^T r \cdot ev, \perp = 0 \Leftrightarrow p^T car \cdot ev,$$

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导出

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

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

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

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

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

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

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

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

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 $p^l$ 和 $p^r$ 带入约束方程

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

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 $[x, y]$ 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

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

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

$$v = (v_l + v_r)/2 = r(w_l + w_r)$$

关于车的航向角 $\theta$ ，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，实际上可以理解为由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

$$\theta^{\cdot} = rL(w_r - w_l)$$

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

$$x^{\cdot} = r^2 \cdot \cos\theta \cdot (w_l + w_r)$$

$$y^{\cdot} = r^2 \cdot \sin\theta \cdot (w_l + w_r)$$

因此差速轮的运动学模型可以最终表示为

$$[\dot{x} \ \dot{y} \ \dot{\theta}] = [r_2 \cos \theta \ r_2 \cos \theta \ r_2 \sin \theta \ r_2 \sin \theta - rLrL][\omega_l \ \omega_r]$$

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；

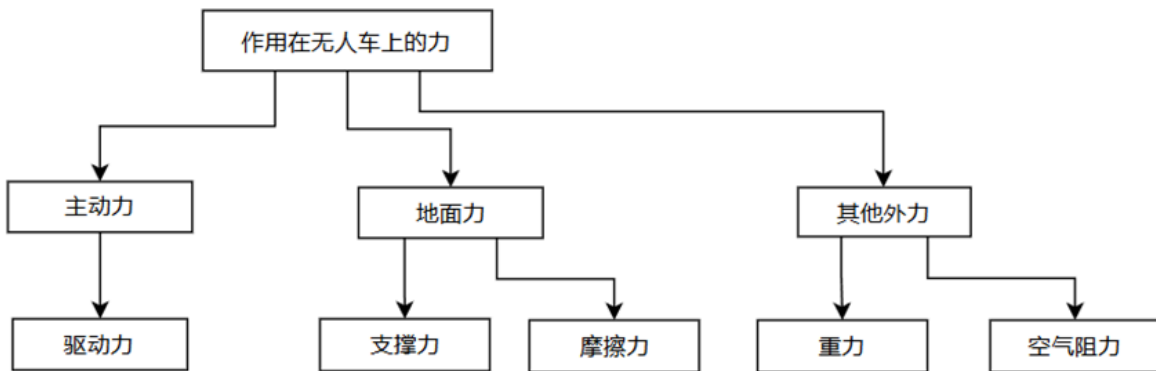
双轮差速模型的缺点：载重小，易侧翻。

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

Ø 详细的PX4机架文件配置参考 [添加一个新的机型 | PX4 自动驾驶用户指南 \(v1.12\)](#)

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：

```
1 | .${R}etc/init.d/rc.rover_defaults
2 |
3 | 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：

```
1 | set PWM_OUT 1234
2 |
3 | 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

Ø 详细的PX4混控文件逻辑见：[混控器和执行器 | PX4 自动驾驶用户指南 \(v1.12\)](#)

Ø 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

Ø 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	[0.02115,0,0; 0,0.02195,0; 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)
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

无人车走到目标点就停止。



## 3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\4.RflySimModel\2.AdvExps\e6\\_CarR1DiffCtrl\5.Car825Posctrl\\_Py](#)

### 4.1 软件要求

Windows 10及以上版本；RflySim工具链；MATLAB2022B以上版本。

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台。

①：推荐配置请见：<https://rflysim.com/>

## 5. 实验步骤

### step1生成DLL文件

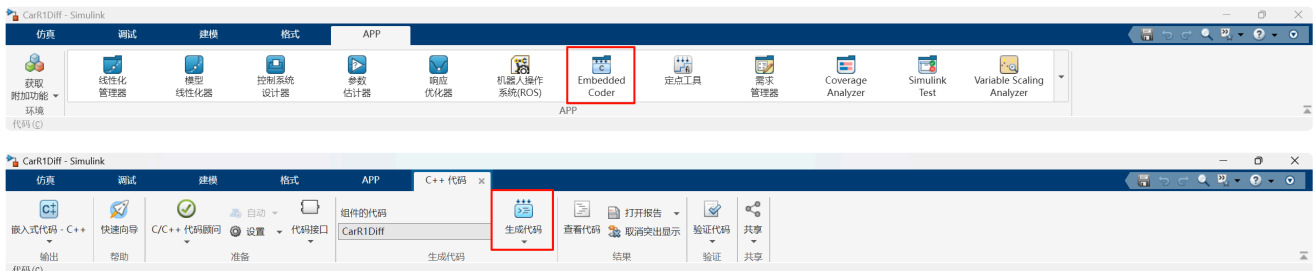
[!!IMPORTANT]

注意：在生成DLL文件之前需要安装VisualStudio，如安装过，忽略此步骤，没安装过参考[\[安装目录\]\PX4PSP\RflySimAPIs\1.RflySimIntro\2.AdvExps\e6.VisualStudioInstall](#)中readme进行安装。

使用matlab运行 [CarR1Diff\\_init.m](#) 参数文件。



打开 CarR1Diff.slx 模型，在APP中找并生成代码。



在诊断查看器中看到编译过程以成功字段，代表编译成功。

```

### Invoking Target Language Compiler on CarR1Diff.rtw
### Using System Target File: F:\matlab\rtw\c\ert\ert.tlc
### Loading TLC function libraries
### Initial pass through model to cache user defined code
### Caching model source code
### Writing header file CarR1Diff.h
### Writing source file CarR1Diff.cpp
### Writing header file rtwtypes.h
.
### Writing source file ert_main.cpp
### TLC code generation complete (took 2.76s).
### 正在保存二进制信息缓存。
.....### 已成功完成 CarR1Diff 的代码生成

```

编译过程已成功完成

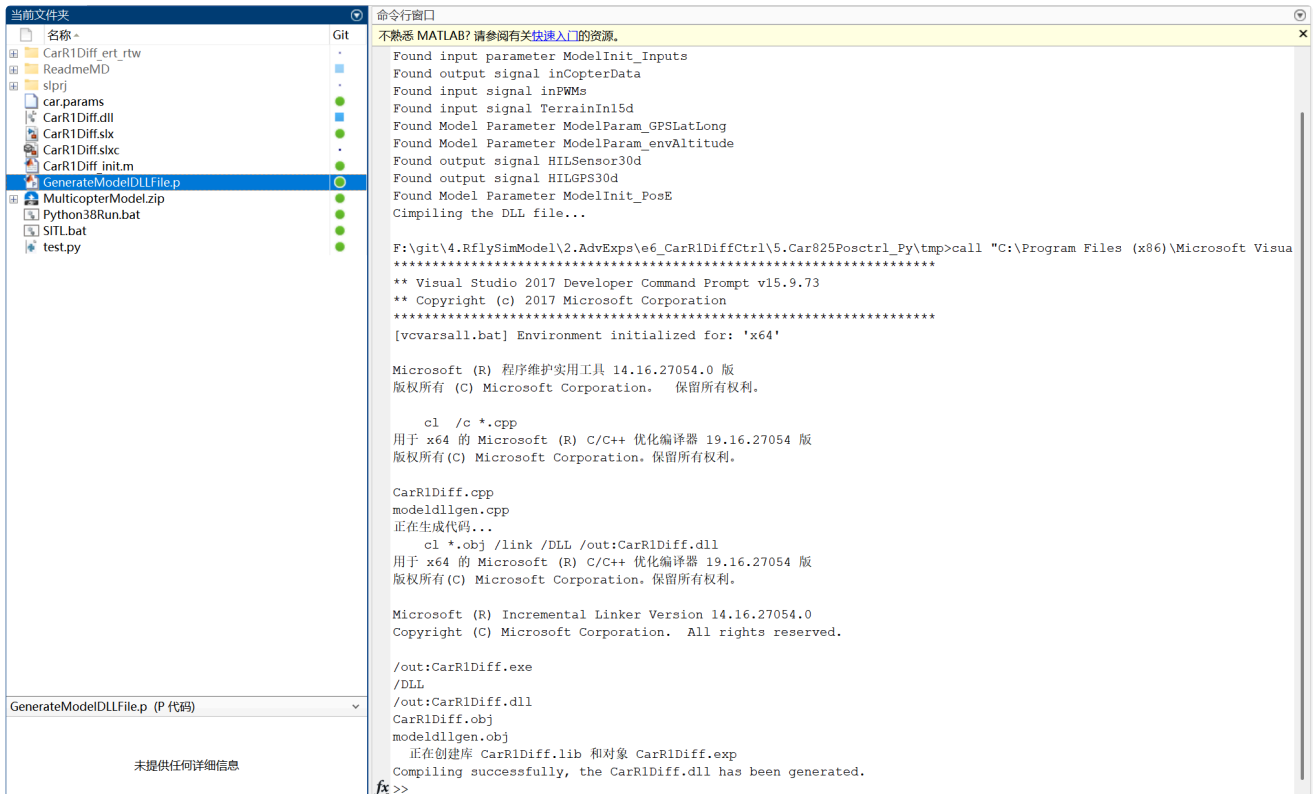
### 编译摘要

编译的顶层模型目标:

模型	操作	重新编译原因
CarR1Diff	代码已生成。	代码生成信息文件不存在。

编译了 1 个模型，共 1 个模型(0 个模型已经是最新的)  
编译持续时间: 0h 0m 16.037s

在编译成功之后，在MATLAB运行 [GenerateModelDLLFile.p](#) 生成DLL动力学模型。



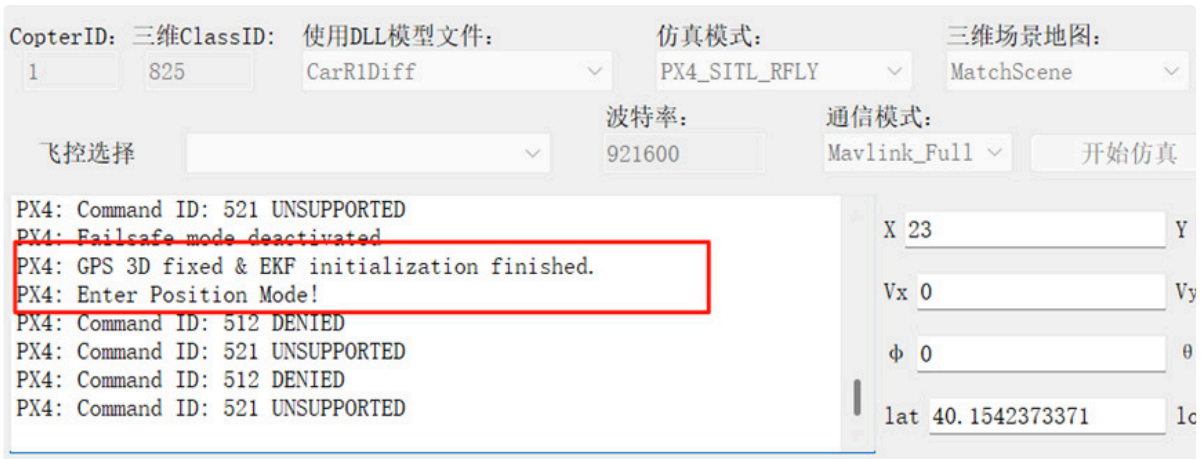
## step2调用DLL文件

将例程文件夹生成的 `CarR1Diff.dll` 文件，复制到 `*:\PX4PSP\CopterSim\external\model` 文件夹下。

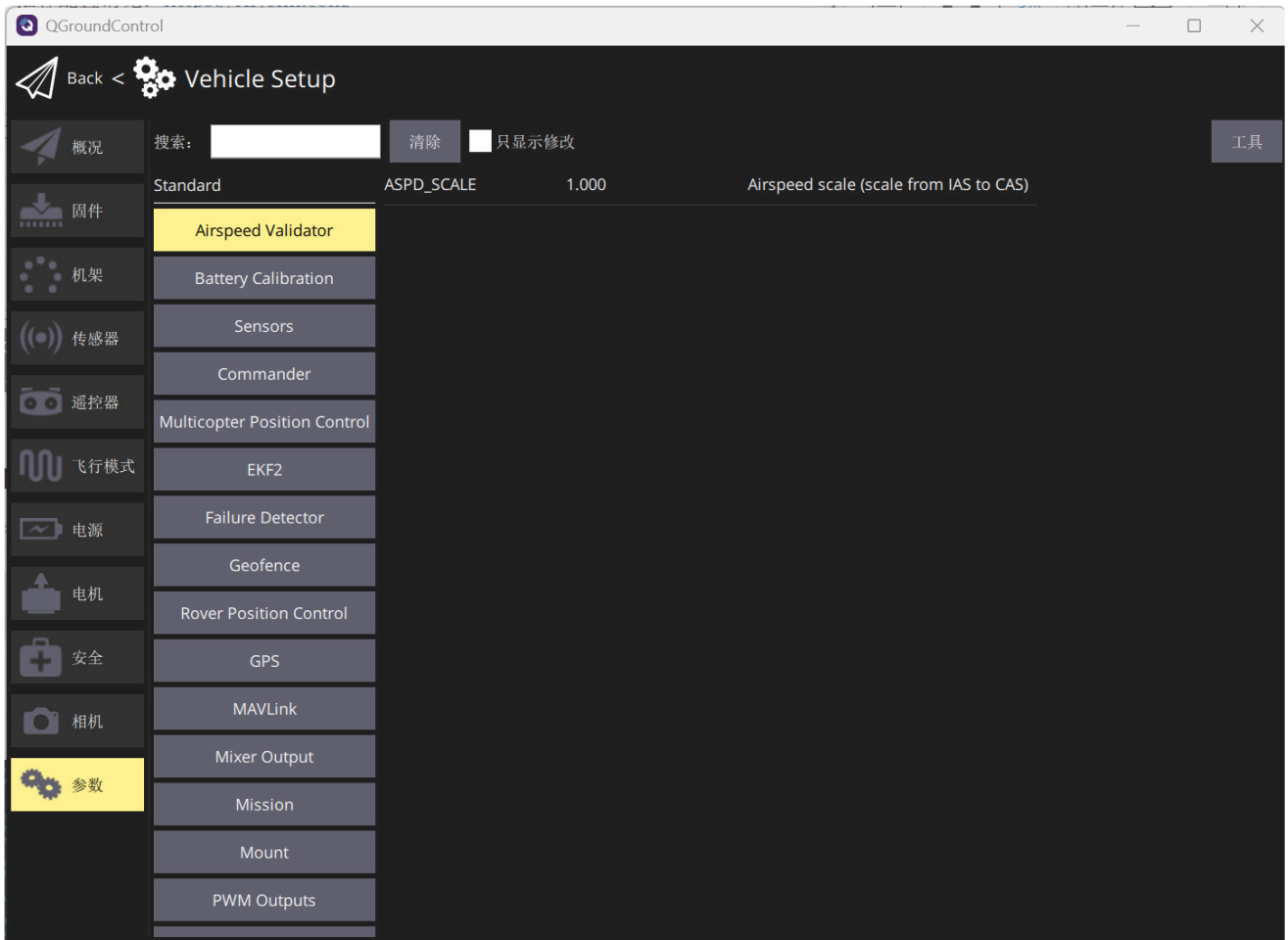
名称	修改日期	类型	大小
AircraftMathworks.dll	2023/9/26 1:25	应用程序扩展	255 KB
CarNoCtrl.dll	2025/6/19 10:12	应用程序扩展	226 KB
CarR1Diff.dll	2025/6/20 11:00	应用程序扩展	211 KB
CopterSILVelCtrl.dll	2023/9/14 18:10	应用程序扩展	156 KB
DIISilNoPX4TempDemo.dll	2023/9/26 0:45	应用程序扩展	149 KB
Exp1_MinModelTemp.dll	2023/9/26 0:18	应用程序扩展	226 KB
Exp2_MaxModelTemp.dll	2023/9/19 18:03	应用程序扩展	240 KB
FixWingModel.dll	2023/10/30 23:17	应用程序扩展	255 KB
MulticopterNoCtrl.dll	2020/6/18 17:46	应用程序扩展	277 KB
MulticopterNOPx4.dll	2024/9/13 14:01	应用程序扩展	170 KB
MultSILSwarm.dll	2019/7/30 16:37	应用程序扩展	284 KB
SmallFixedWingUAVnoctrlHIL.dll	2023/9/26 1:50	应用程序扩展	255 KB
Trailer.dll	2023/9/20 11:47	应用程序扩展	355 KB

## step3加载参数

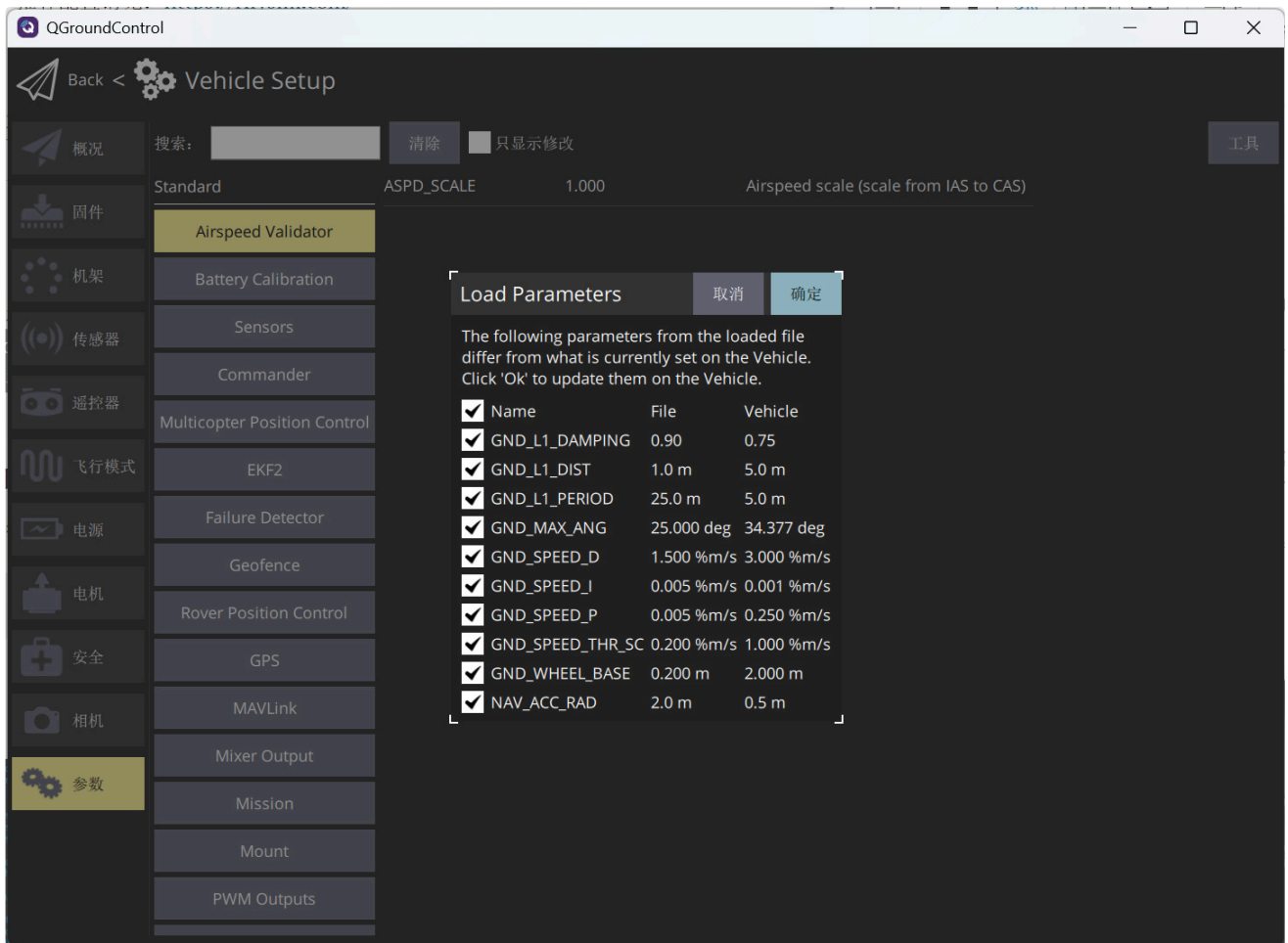
以管理员身份运行 `SITL.bat` 一键启动脚本，在弹出窗口选择是，会打开一个CopterSim，一个RflySim，一个QGroundControl，等待CopterSim信息栏出现PX4: GPS 3D fixed & EKF initialization finished.字段。



找到QGroundControl点击左上角图标-Vehicle Setup-参数。

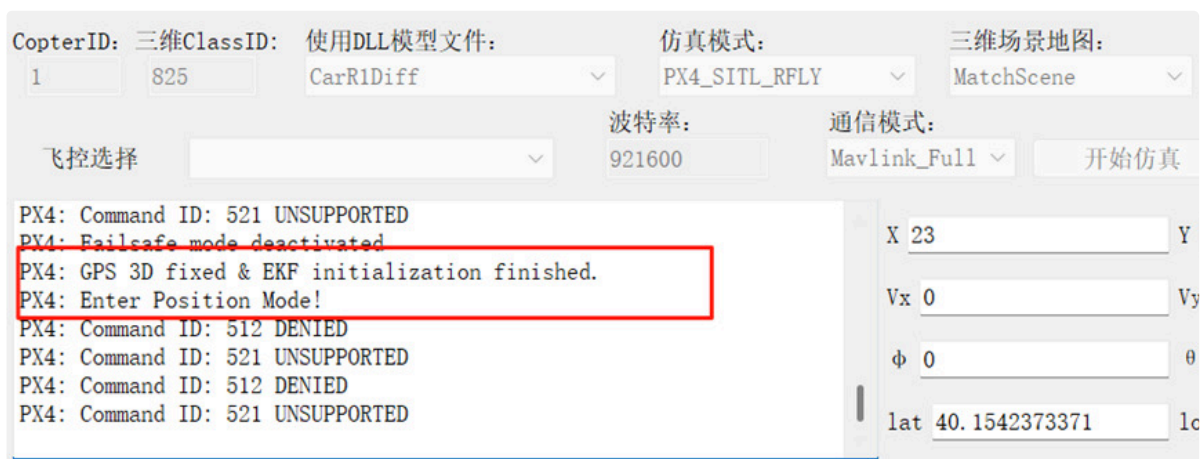


在参数页面点击工具-加载参数选择文件夹中car.params参数文件点击确定进行导入。



## step4软件在环仿真

重新运行 `SITL.bat` 一键启动脚本，在弹出窗口选择是，会打开一个CopterSim，一个RflySim，一个QGroundControl，等待CopterSim信息栏出现PX4: GPS 3D fixed & EKF initialization finished.字段。



打开 `Python38Run.bat` 运行 `python test.py` 并执行。

```
F:\git\4.RflySimModel\2.AdvExps\e6_CarR1DiffCtrl\5.Car825Posctrl_Py>python test.py
HostIP is 192.168.1.3
Start listening CopterSim heartbeat Msg ...
Got time msg from CopterSim # 1 , running on this PC
End listening CopterSim heartbeat.
Got 1 CopterSim on the LAN.
Request CopterSim Send data.
开始进入Offboard模式
开始进入航路寻迹模式
4.49335065664419
4.334313950458667
4.240955612627169
4.174678707636225
4.108617219493457
4.010847350052959
3.9695965351787987
2.9258184026242662
```

在RflySim3D查看实验效果。



## step5:还原参数

在桌面RflyTools文件夹中打开WinWSL运行make distclean等待执行完毕，重新运行软件在环脚本，等待编译成功，参数也会还原。