JetRacer 自动驾驶沙盘使用与讲解

1.功能简介

JetRacer 是一款基于 Jetson Nano 的自动驾驶小车，它预装了一个特定的操作系统镜像，旨在简化设置过程并提供一个开箱即用的开发环境。

JetRacer 预安装了深度学习框架 TensorFlow 和 PyTorch，支持构建和训练模型，以及 TensorRT 用于高性能推理的深度学习优化器和运行时，让自动驾驶更快更安全。它还预安装了 Jupyter Notebook，支持网络交互式编程，使用户开发更方便。

本功能基于 JetRacer 开发一款基于 jetson nano 主控自动驾驶深度沙盘（在 orin 系列主控也有适配该功能），JetRacer 功能用于沙盘中的道路保持，同时结合了 ROS1 的 darknet_ros 深度学习中的交通标志识别功能，利用交通识别功能识别交通标志，完成指定的动作。

2.使用方法

① Jetson nano 主控修改.bashrc（orin 系列不用）

修改.bashrc 文件，将其中一行取消注释，就能正常使用交通沙盘功能（ROS 功能中的 darknet_ros 深度学习也可正常使用）。若想使用小车的基础 ROS 功能，则添加#注释即可。（沙盘功能包中的 cv_bridge 环境与 ROS 功能的纯视觉建图、3D 建图、AR 标签识别等功能的 cv_bridge 环境冲突了，运行相关功能需要切换 cv_bridge 环境），修改完重新打开命令窗口

图 1　gedit ~/.bashrc

② 修改 wheeltec_jetracer.launch 中的车型

根据当前的小车型号，修改 wheeltec_jetracer.launch 中关于 ros_param_file 参数的注释，当前的小车型号 ros_param_file 参数取消注释，其余的小车型号 ros_param_file 参数注释掉。

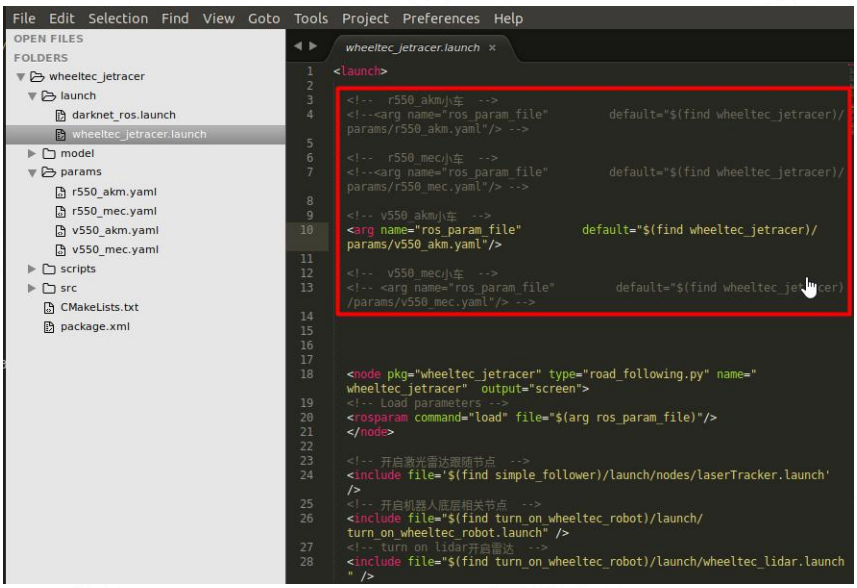

图 2　wheeltec_jetracer.launch 参数

③ 修改 params 中的参数

根据当前的小车型号，选择修改 wheeltec_jetracer/params 下的 yaml 参数文件。

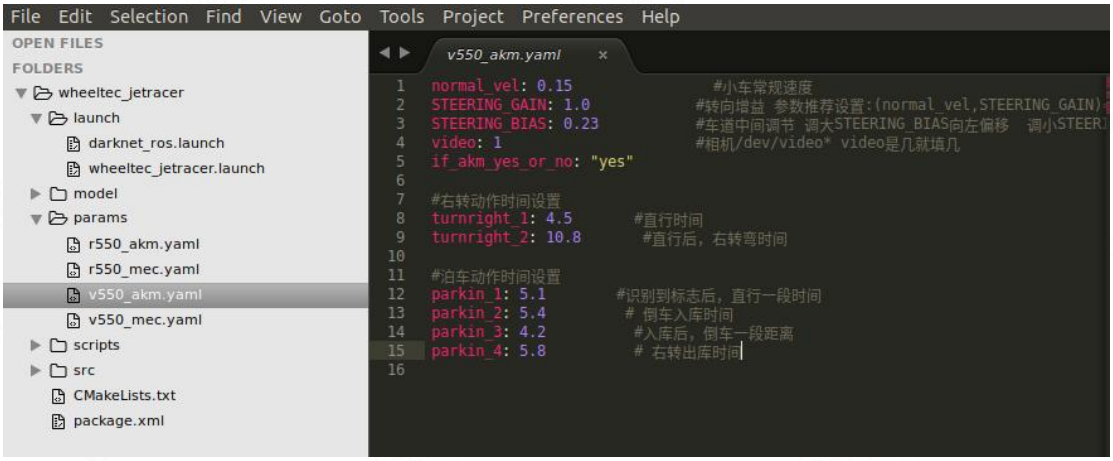

图 3　params 参数

其中，video 参数根据可调节视角的 astrapro/C70 相机的/dev/video 的序号修改（运行 ll /dev 查看，一般 R550 小车为 0；V550 小车时，主控为 jetson nano 时为 1，orin 主控时为 2），normal_vel 为小车常规速度，常规速度不能太大，一般设置为 0.15m/s 即可。STEERING_GAIN 参数则为小车的转向增益，如果小

车左右摇摆，降低转向增益，如果小车没有转弯或者转弯时小车冲出外道，则增加转向增益。STEERING_BIAS 参数则为车道中间调节参数，如果要使汽车靠右，使转向偏置设置更大，如果要使汽车靠左，使转向偏置设置更小。

如果遇到转弯标志转弯或者泊车动作有误，可以通过修改 yaml 中的动作时间来调整。

④ 调节相机视角

运行相机功能，打开可调节视角的 astrapro/C70 相机

```
roslaunch turn_on_wheeltec_robot wheeltec_camera.launch
```

运行打开 rqt_image_view 便可看到相机视角

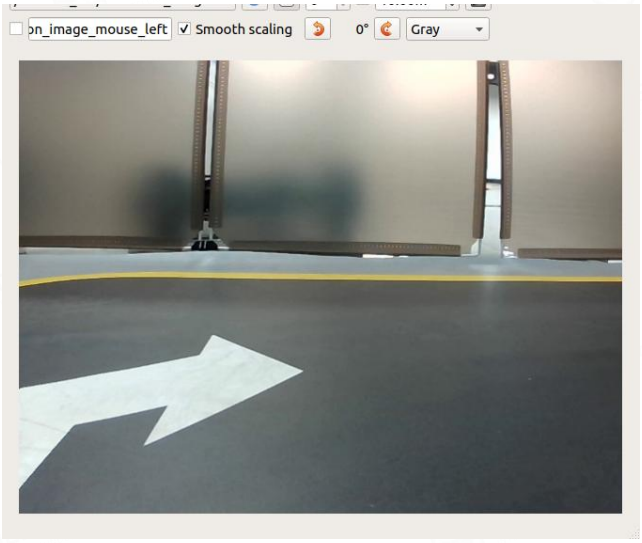

图 4 rqt_image_view

我们需要先把小车放在指定位置（大概小车右前轮在直线与弯道连接处），调节相机视角，使前方的黄线在图像中间位置即可（如图 3）。

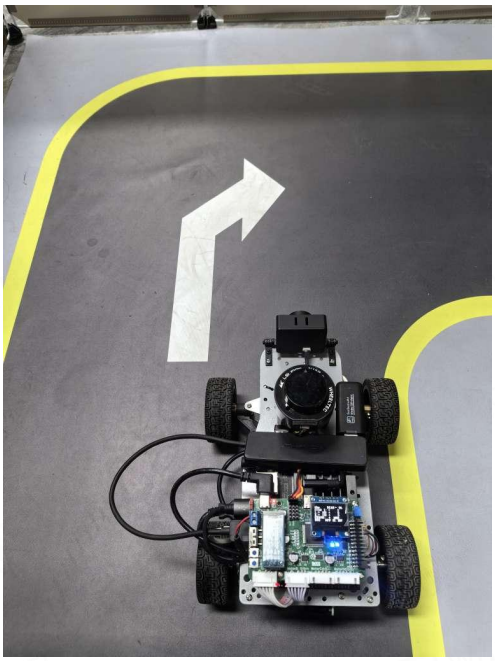

图 5 小车放置位置

⑤ 运行 JetRacer 自动驾驶沙盘功能

把小车放在外围一圈的外道直道中，然后运行 JetRacer 自动驾驶沙盘功能：

先运行适配好的 darknet_ros 深度交通标志识别：

```
roslaunch wheeltec_jetracer darknet_ros.launch
```

再运行 JetRacer 自动驾驶功能（需要等大概 3 分钟运行加载完成）：

```
roslaunch wheeltec_jetracer wheeltec_jetracer.launch
```

运行效果如图所示：

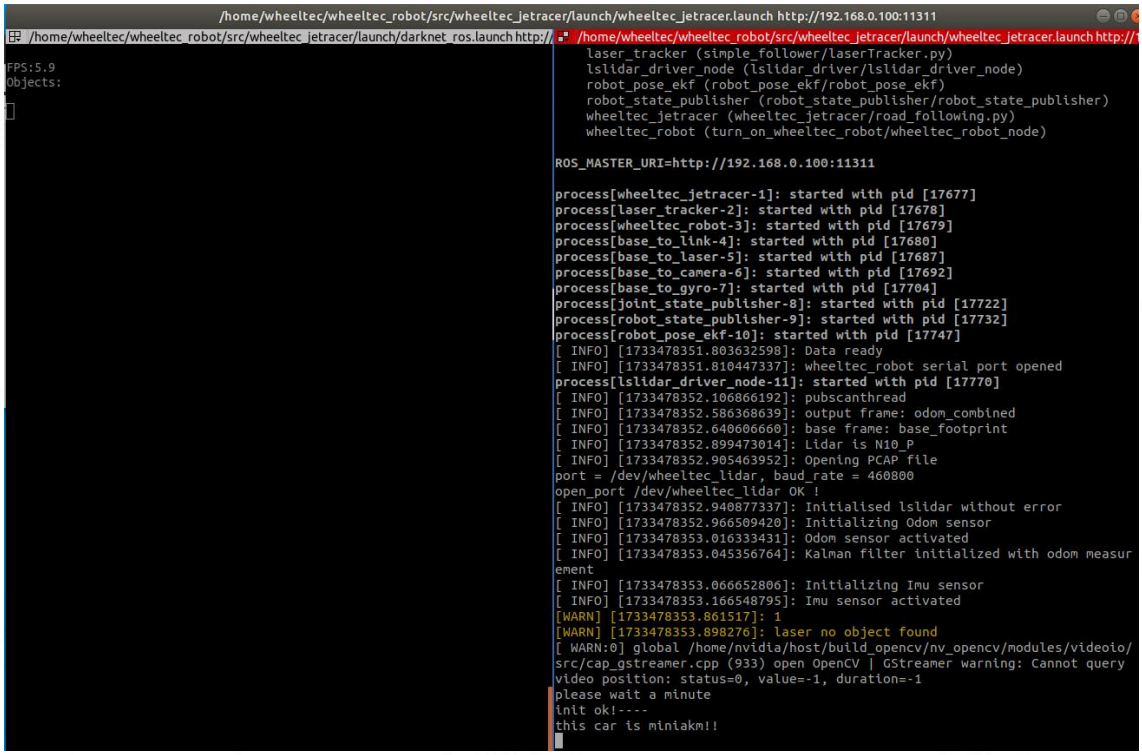

图 6　终端运行 jetracer

JetRacer 自动驾驶功能主要识别的交通标志如图所示：

图 7　交通标志识别

每种交通标志分别执行下面动作：

直行：加速直行一段时间后恢复正常状态

转弯：右转弯动作

学校、慢行、公交车、斑马线标志牌：减速一段时间后恢复正常状态

P：倒车入库后，行驶出去

Stop：小车停 3 秒后，恢复正常运动状态

路障：小车停止运动（标志牌存在的话会一直识别到标志牌导致小车会一直停）

每一种交通标志执行动作在 wheeltec_jetracer/scripts/road_following.py 文件中可以看到。

图 8　交通标志执行动作编程

我们目前的沙盘地图如下：

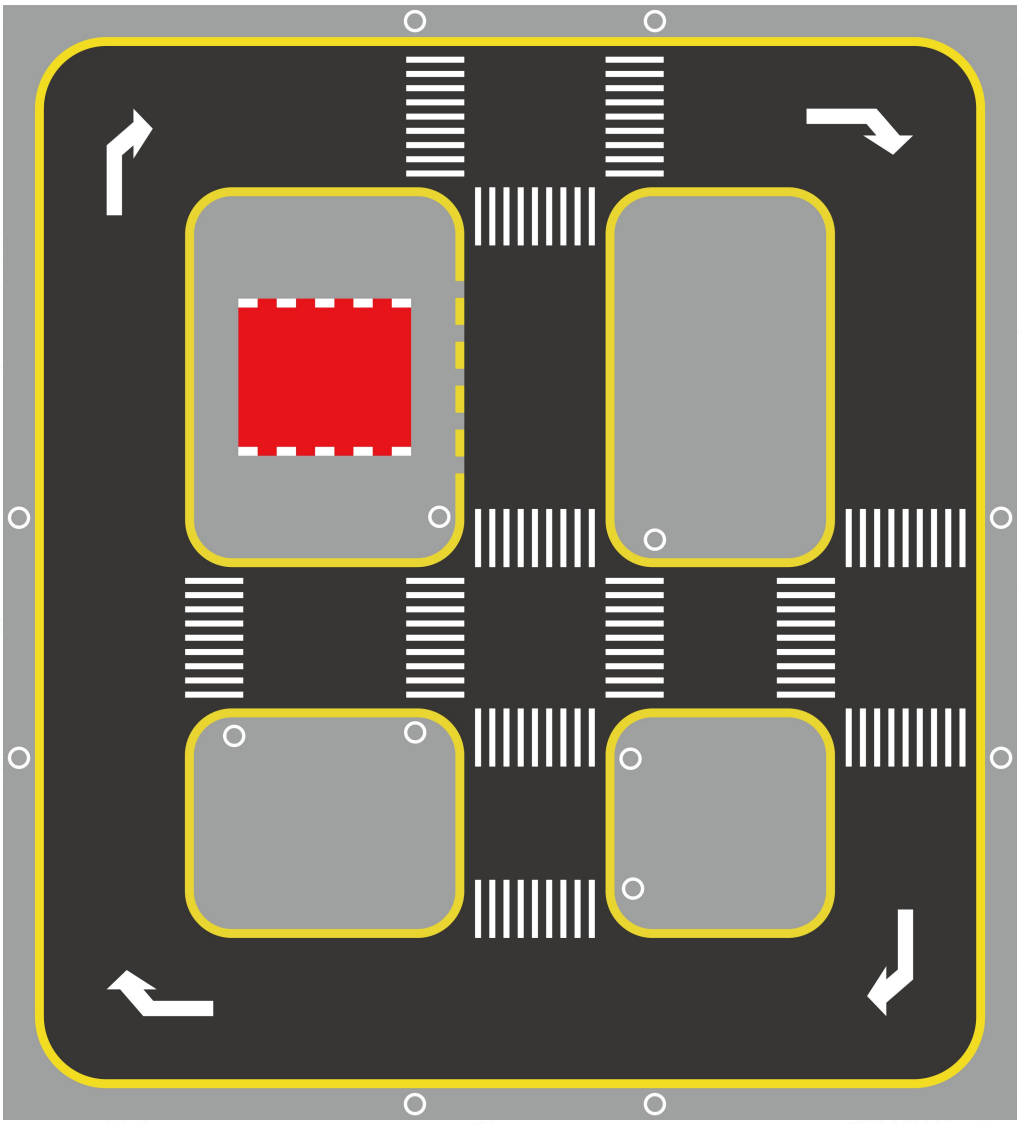

图 9　交通沙盘地图

沙盘的行驶路线有三种：

第一：不放转弯标志牌的情况下，小车围绕最外一圈一直行驶。此时，可以在最外一圈摆放交通标志牌（路障标志牌可以放在道路中间，小车识别到标志牌会停止运动，拿走标志牌则恢复正常运动状态。其余除了转弯标志牌外，可以放在最外一圈的小圆中，小车识别到标志牌会执行相应动作）

第二：在 1、2 处摆放右转交通标志牌，小车会在 1 处识别到转弯标志，并驶入十字路口位置，在 2 处识别到右转交通标志，会驶出十字路口位置。在此中，可以在 5 处摆放 P 交通标志，或者在 6、7 处摆放其余的交通标志，来执行相应动作。

第三：在 3、4 处摆放右转交通标志牌，小车会在 3 处识别到转弯标志，并

驶入十字路口位置，在 4 处识别到右转交通标志，会驶出十字路口位置。在此中，可以在 1、8 处摆放其余的交通标志，来执行相应动作。

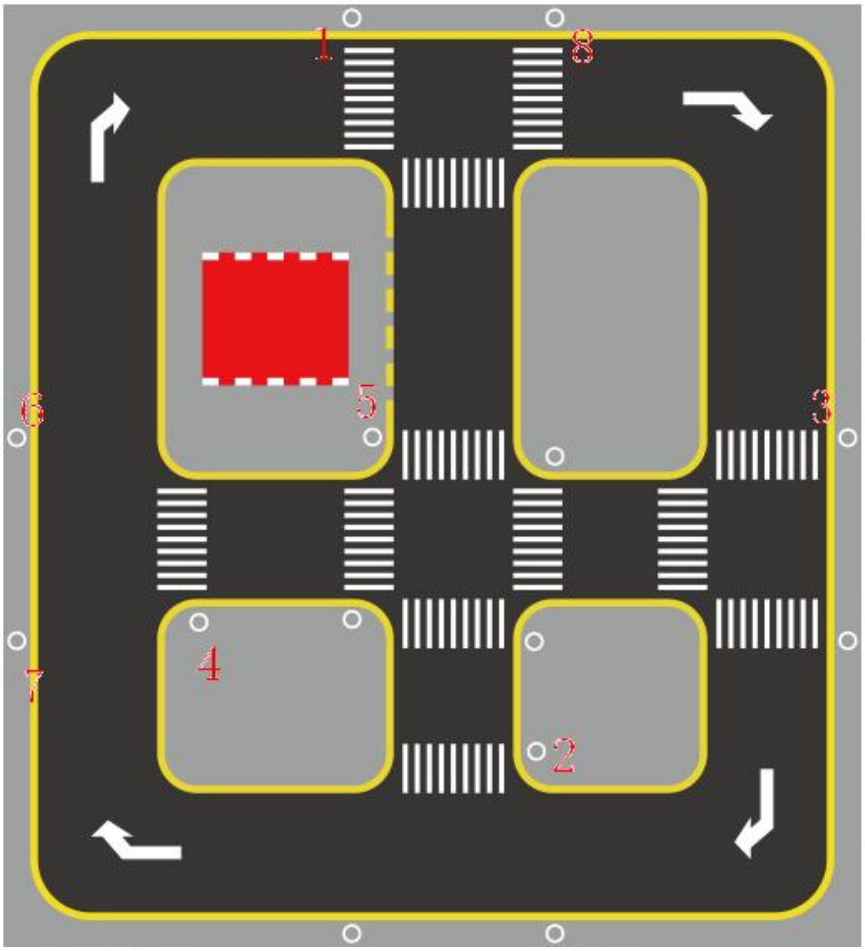

图 10　交通标志摆放

3.注意事项

①　相机视角要按要求调整合适位置：过高则容易导致小车转弯过晚，小车冲出跑道，过低则导致小车不能正常识别交通标志牌

②　必须要放转弯标志驶出十字路口位置：如果不放，小车有几率可能不按正常路线走，小车失控。

③　小车尽量在道路中间或者稍微偏左一点行驶，避免影响交通标志的识别。

4.功能讲解

雷达跟随功能涉及源码主要存放于工作空间下 wheeltec_jetracer 功能包中。主要是运行功能包中的 script/road_following.py 文件来实现沙盘功能，包括了车

道保持训练模型的加载、小车车道保持、相机打开、对于 darknet_ros 深度交通标志识别结果处理、交通标志成功识别后固定动作的执行等功能。

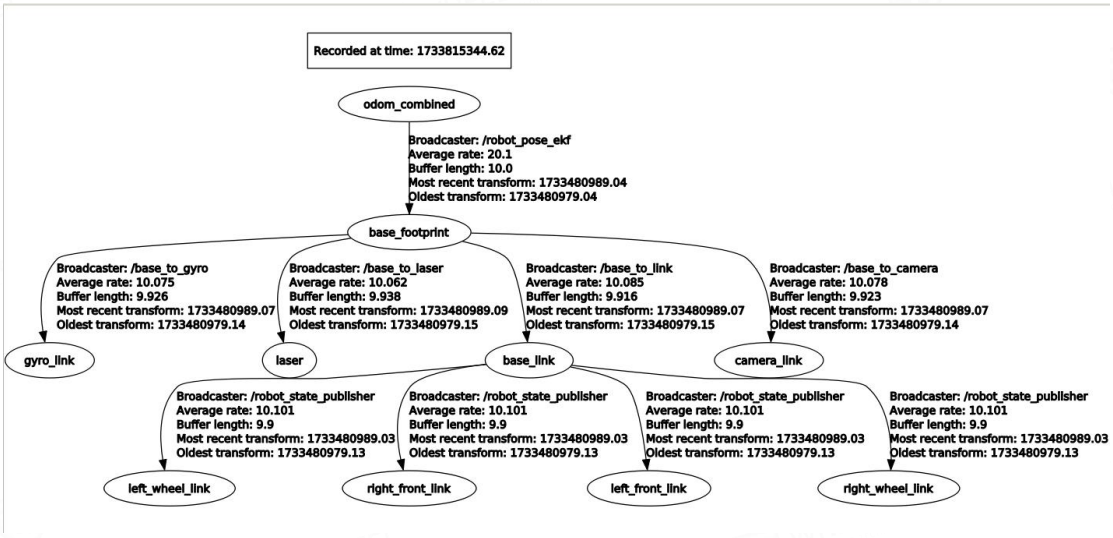

图 11 功能节点示意图

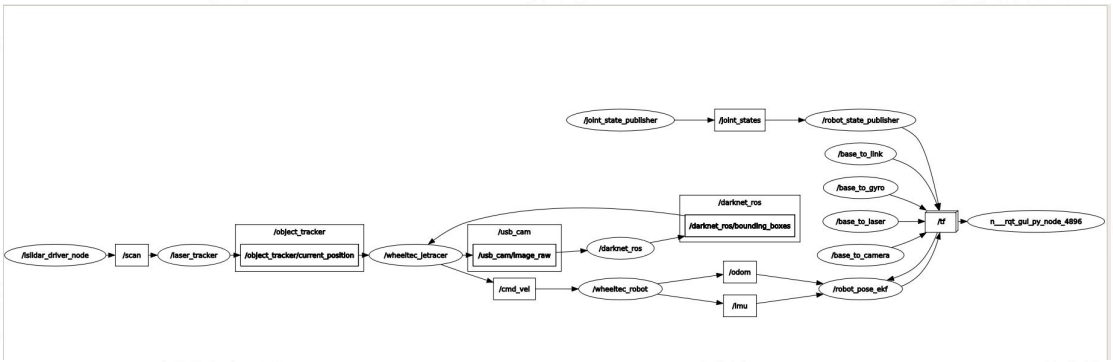

图 12 rqt_graph