

## Ros 导航小车仿真环境搭建

### 0、ros 安装

(建议 csdn 参考 [ros ubuntu20.04 安装教程](#))

期间，如果在 `rosdep update` 卡住，请参考 b 站教程：[rosdep update-哔哩哔哩\\_Bilibili](#)

### 1、TurtleBot3 配置

完成 [TurtleBot3 \(robotis.com\)](#) 中的第六节的三个配置教程，注意在每个小节中都要选择 noetic 版本，如下图：



此外，建议比较详细的阅读配置的文字部分，否则可能会漏掉某些配置过程和要求。

- 6.1 Gazebo Simulation (需要安装 `turtlebot3` 和 `turtlebot3_msgs` 的依赖包)
- 6.2 SLAM Simulation
- 6.3 Navigation Simulation

这三个小节分别对应 gazebo 模型的导入，slam 建图，与导航仿真的实现

在 6.3 节的导航模拟过程中，会缺少 `move_base` 包而发生报错，报错信息中会提到缺少 `dwa_local_planner` 等之类的信息，在 `rviz` 上出现的现象为小车静止在原地，无法运动。这时则需要安装相应的 `move_base` 包，见下一节：

如果 6.3.4 小节中的 Set Navigation Goal 也就是在 `rviz` 上的导航可以成功运行的话，则说明配置成功。

### 2、有两种方式可以选择

- `Move_base` 源码安装地址：<https://github.com/ros-planning/navigation> 源码下载后需要编译和安装。git 下来的包并不可以直接编译，而是需要调整一下目录结构。
- 也可以直接通过命令行安装: `sudo apt install ros-noetic-navigation`

### 3、teb 包安装

这一步的目的是下载 `teb`，并将 `move_base` 的默认局部路径规划算法由 `dwa` 切换为 `teb`。首先 google 搜索 `ros teb` 并 git clone 源代码，（这里的 `teb` 包需要 clone 在 `ros` 工作空间的 `src` 目录下），并完成如下命令：

- `catkin_make` 完成编译
- `catkin_make install` (不要执行，这里的 `install` 如果不指定安装目录，会自动安装到当前文件夹下，导致无法被 `move_base` 找到，因此要指定安装目录，看下面)
- `sudo su` 切换到 root，否则下一步无法安装到 `/opt/ros/noetic`
- (可能会需要 `source` 一下)
- `catkin_make -DCMAKE_INSTALL_PREFIX=/opt/ros/noetic install` 指定目录安装

`catkin_make install` 指定目录安装的参考网站为：

[http://wiki.ros.org/catkin/commands/catkin\\_make](http://wiki.ros.org/catkin/commands/catkin_make))

### 4、修改 TurtleBot 的 launch 文件，将 DWA 修改为 TEB

- 先运行 `sudo find / -name turtlebot3_navigation.launch`
- 找到 `turtlebot3_navigation.launch` 在系统中的位置后，`cd` 到目录下，用 `vscode` 打开 `move_base.launch` 文件：

修改如下：（修改其中的一行，注释其中的两行，他们都与 dwa 有关）

```
move_base.launch x
move_base.launch
1 <launch>
2 <!-- Arguments -->
3 <arg name="model" default="$(env TURTLEBOT3_MODEL)" doc="model type [burger, waffle, waffle_pi]"/>
4 <arg name="cmd_vel_topic" default="/cmd_vel" />
5 <arg name="odom_topic" default="odom" />
6 <arg name="move_forward_only" default="false"/>
7
8 <!-- move_base -->
9 <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
10   <param name="base_local_planner" value="teb_local_planner/TebLocalPlannerROS" />
11   <roscppparam file="$(find turtlebot3_navigation)/param/costmap_common_params_${arg model}.yaml" command="load" ns="global_costmap" />
12   <roscppparam file="$(find turtlebot3_navigation)/param/costmap_local_params_${arg model}.yaml" command="load" ns="local_costmap" />
13   <roscppparam file="$(find turtlebot3_navigation)/param/global_costmap_params.yaml" command="load" />
14   <roscppparam file="$(find turtlebot3_navigation)/param/move_base_params.yaml" command="load" />
15   <!-- <roscppparam file="$(find turtlebot3_navigation)/param/dwa_local_planner_params_${arg model}.yaml" command="load" /> -->
16   <remap from="cmd_vel" to="$(arg cmd_vel_topic)"/>
17   <remap from="odom" to="$(arg odom_topic)"/>
18   <!-- <param name="DWAPlannerROS/min_vel_x" value="0.0" if="$(arg move_forward_only)" /> -->
19 </node>
20 </launch>
21
22
```

5、重新运行 launch，小车可以正常导航，则配置成功。