

2D 导航功能使用与讲解

1. 功能简介

2D 导航功能是通过激光雷达结合 SLAM 算法实现的，同时导航软件包的功能是基于 ROS Navigation 实现的。在 2D 导航过程中，雷达识别所呈现的效果为 2D 平面效果。在导航功能中，需要使用 2D 建图功能中所创建的地图。

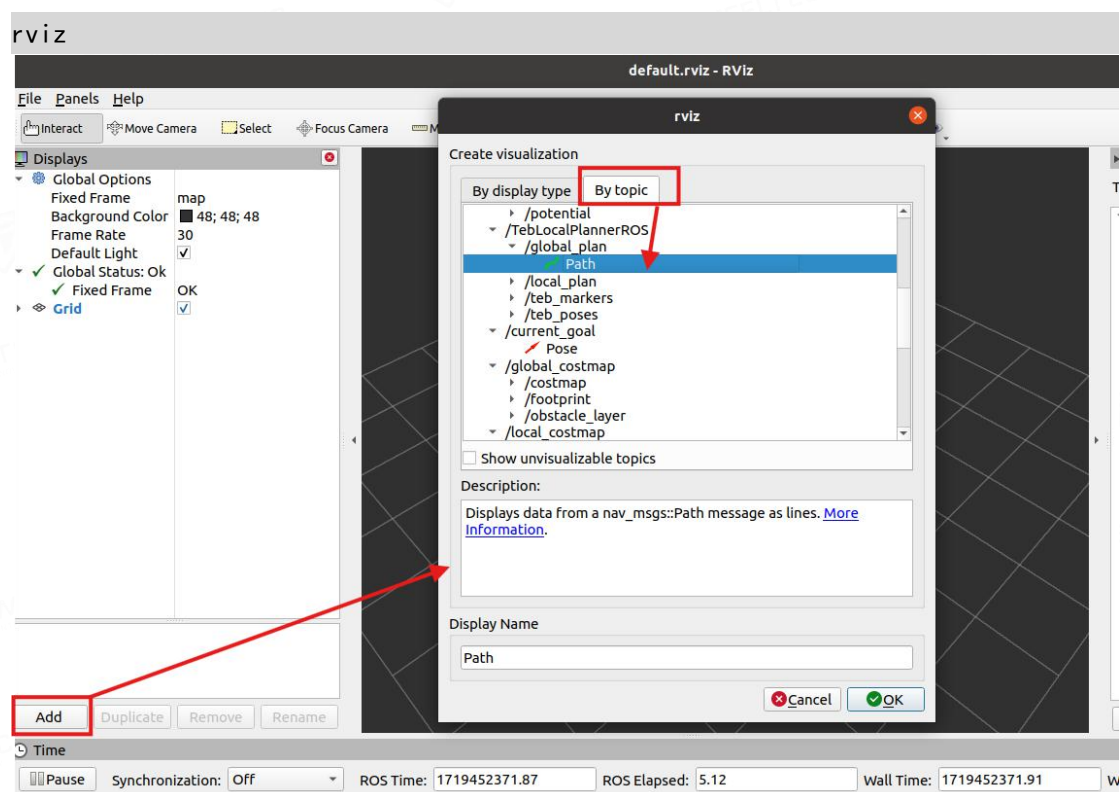
2. 使用方法

使用前需要确保上位机连接小车 WiFi，并已经 SSH 远程登录到小车端。

- 1) 打开终端输入启动 2D 导航功能的指令

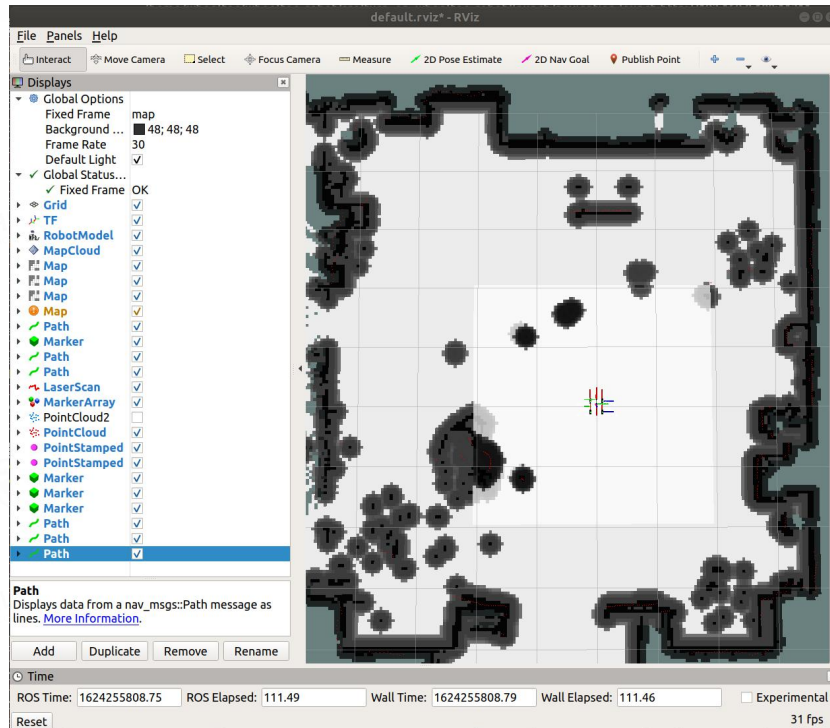
```
roslaunch turn_on_wheeltec_robot navigation.launch
```

- 2) 在上位机端打开 rviz 可视化工具，如下图在 rviz 中的 Add 按键中的 By topic 选项通过话题的形式添加可视化选项。



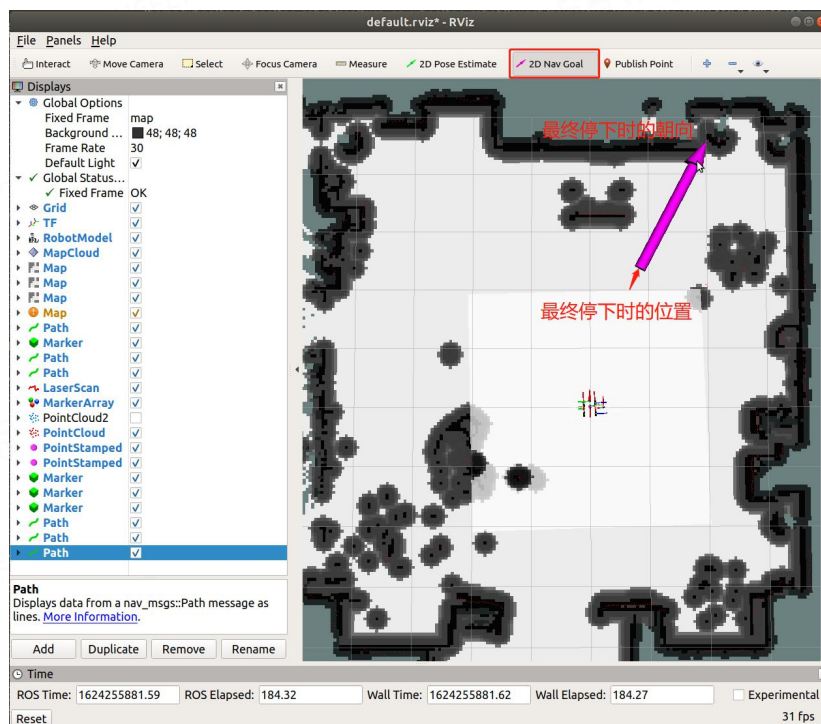
开启 rviz 后添加可视化选项

添加完成后 rviz 显示画面如下图所示：



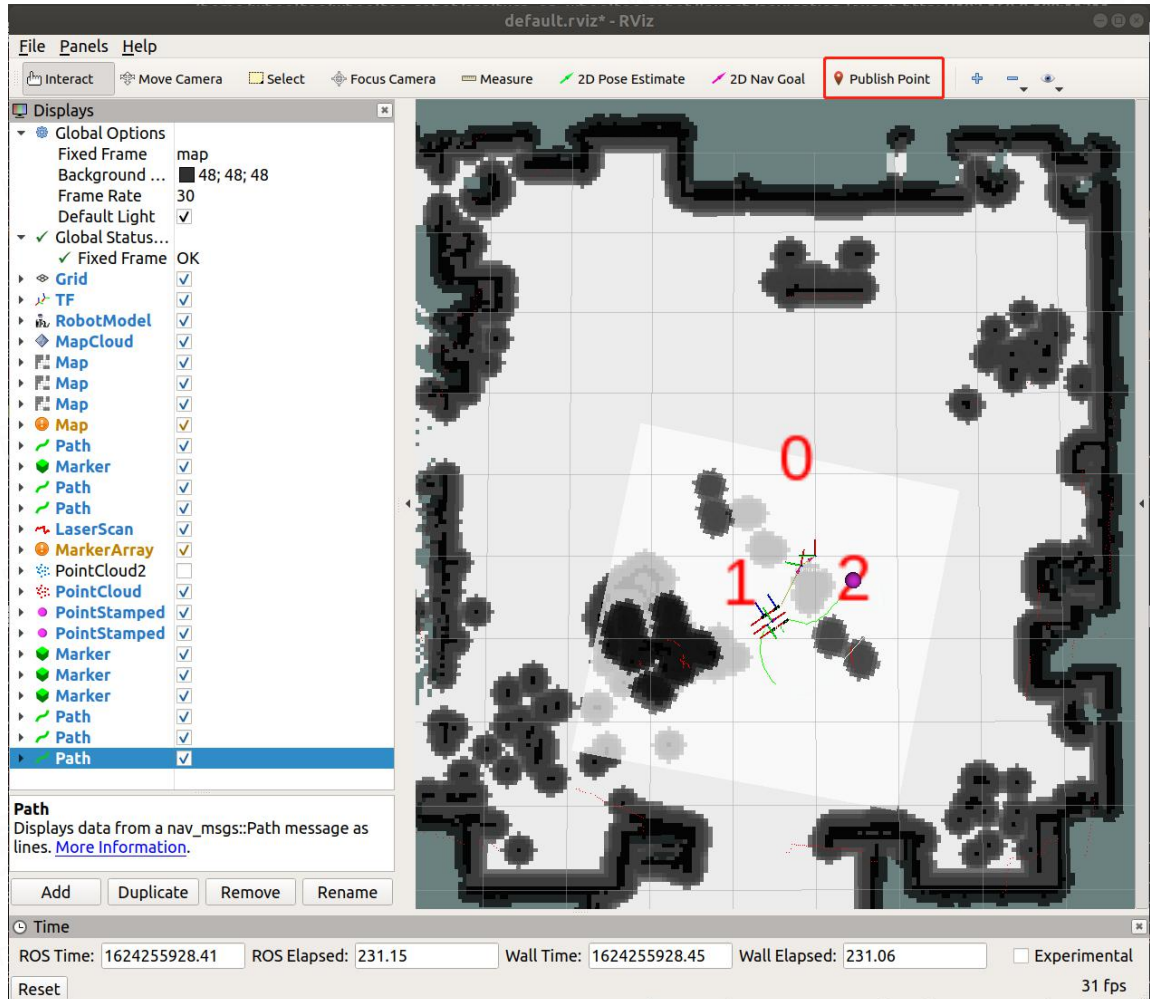
导航功能开启后 rviz 的显示界面

导航模式有两种，一种是单点导航，点击 rviz 上的 2D Nav Goal 按键，然后在地图中选定导航点与方向，单击鼠标确定导航目标点位置，并且不要松开，继续拖动选择导航目标方向，确定后松手，小车就会运动到所指定的目标点，在运行 rviz 的终端也会显示目标点的坐标。



2D 导航功能——单点导航

另一种是多点导航，点击 rviz 界面上方的 Publish Point 就可以设置用于多点导航的点了，当设置多个点时，小车会在这几个点之间往复运动，使用 Publish Point 工具进行多点导航时，所设置的导航点默认方向都是小车的初始车头方向。



2D 导航功能——多点导航

3. 注意事项

① 开始导航前加载地图

2D 导航功能的实现需要提前进行建图，且导航起点要与建图起点一致，建议建图起点选好参照物，这样在导航前放置小车时，导航起点与建图起点就不会有太大的偏差，将小车放在建图起点后开启 2D 导航功能。示例中设定了导航所用地图为 2D 建图功能中所建的 WHEELTEC.yaml 文件，打开 rviz，可以看到所设置的导航地图，注意，要等 2D 导航节点完全开启后再运行 rviz，这样可以避免因为节点还未完全开启而导致的地图加载失败。

② 小车不运动问题

关于单点导航，注意如果给了目标点但小车并不运动，较大可能是多机通信没有设置好，可参考【多机通信】功能进行配置，还有一个原因可能是电量不足，当 ROS 教育机器人低于 10V、ROS 机器人低于 20V 时，会因电量不足而无法运动，电量可在 STM32 主控板上的 OLED 屏幕右下角进行查看，也有可能是小车之前按下了急停开关或使能开关拨在 OFF 处，可以查看小车 32 端显示屏下方为 ON 或 OFF 字样进行判断。

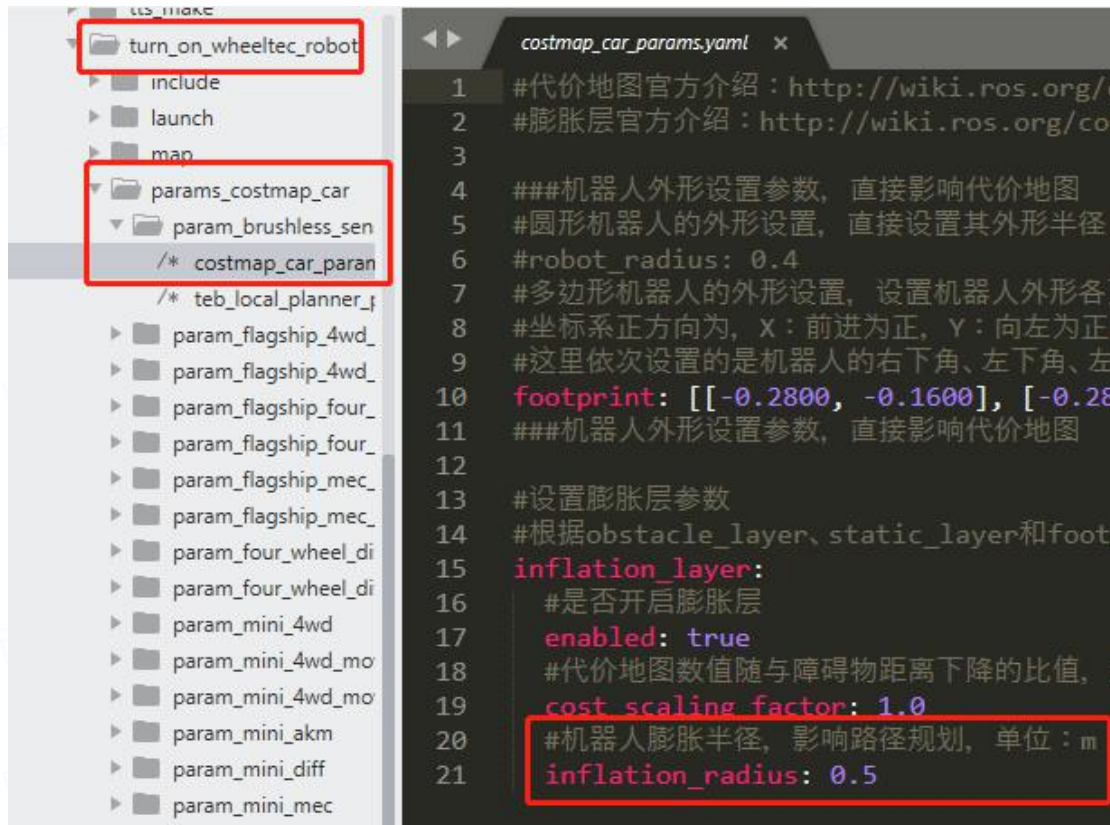
③ 重新设置多点导航的导航点

关于多点导航时可以设置导航点方向的解决方法可参照我们的论坛中的帖子 <http://bbs.wheeltec.net/forum.php?mod=viewthread&tid=2593>。如果想要重新设置导航点，可在终端输入 c 进行清除，清除后小车会到达清除前所设置的下一个导航目标点并停下，此时可以直接开始重新设置导航目标点，无需移动小车位置。

④ 导航过程中障碍物膨胀层问题

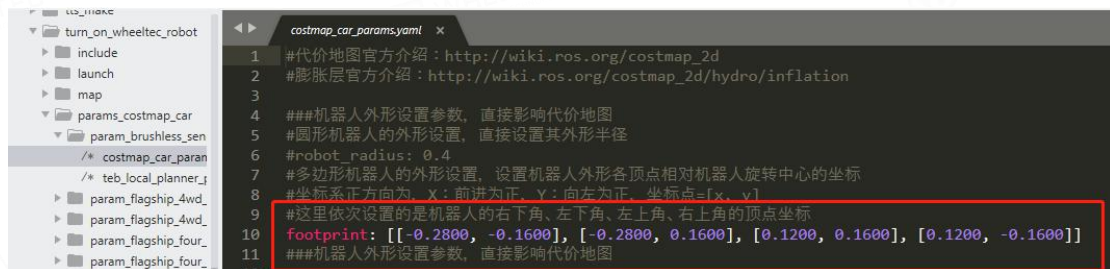
导航过程中，当所规划路径遇到较窄处时，小车可能会因判断无法通过而停下，但实际小车是充分可以通过障碍物间的间隙的，这是因为我们给障碍物设置了膨胀半径，膨胀半径范围内依旧认为是障碍物不能通过，可以通过修改膨胀半径的方式解决，膨胀半径定义在

/turn_on_wheeltec_robot/params_costmap_car/param_【carmode】下的 costmap_car_params.yaml 文件中，修改 inflation radius 参数的值即可修改膨胀半径，【carmode】为车型，我们发货前均已选定正确车型，用户可以在 turn_on_wheeltec_robot 功能包路径下的 turn_on_wheeltec_robot.launch 文件中进行查看。当然，如果车身与障碍间隙太小，即使膨胀半径修改为 0 依旧是无法穿过的。



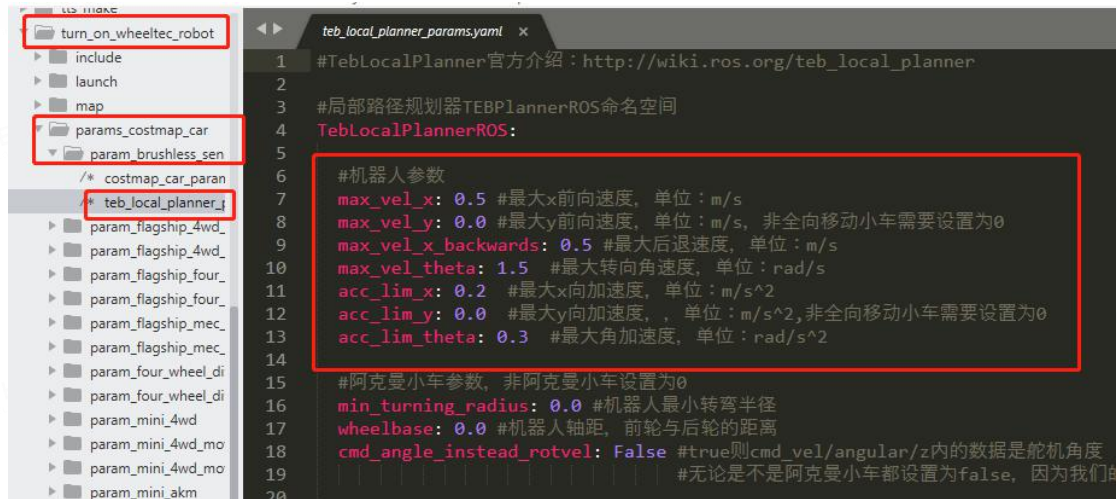
膨胀半径修改

膨胀半径是以小车中心为原点进行设定的，因此，当膨胀半径小于车身时，膨胀半径将不起作用，此时起作用的是同一个文件中对于小车外形的定义，即 footprint，如果要通过的间隙较小，可以考虑修改 footprint，但不建议这么做，较窄间隙小车撞车几率较高。



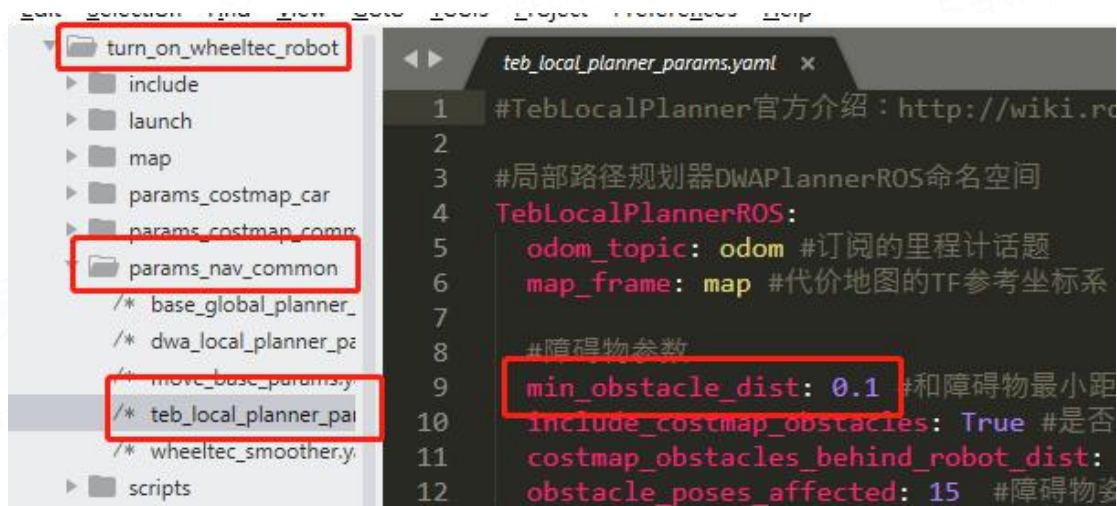
小车车身 footprint

在我们的源码中，默认使用的局部路径规划算法为 teb 导航算法，因此导航速度可在 turn_on_wheeltec_robot/params_costmap_car/param_【carmode】/teb_local_planner_params.yaml 中进行修改。



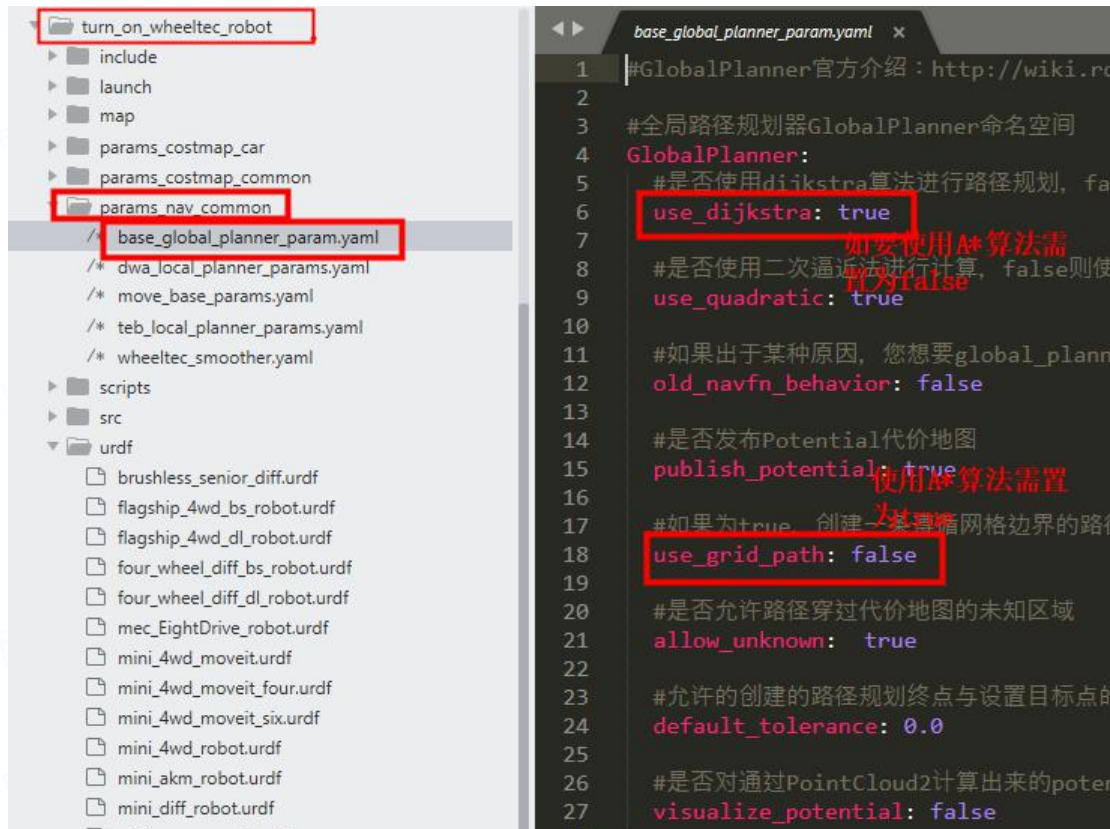
导航速度修改

该文件中车型相关参数 `vertices` 与前文中提到的 `costmap_car_params.yaml` 文件中的 `footprint` 是一致的，同时还有一个影响导航实时避障的参数 `min_obstacle_dist`，这个参数是以车身外侧为基准的。



车身外形参数与障碍物最小距离

局部路径规划算法的选择在下文程序解析中将会介绍，这里先讲一下如何切换全局路径规划算法： `/turn_on_wheeltec_robot/param_nav_common` `/base_global_planner_params.yaml` 文件是关于全局路径规划的参数设置，在我们的源码中默认使用 Dijkstra 算法，如果要更改为 A*算法，可将 `use_dijkstra` 参数的值改为 `false`，同时 `use_grid_path` 参数的值也取反改为 `true`。

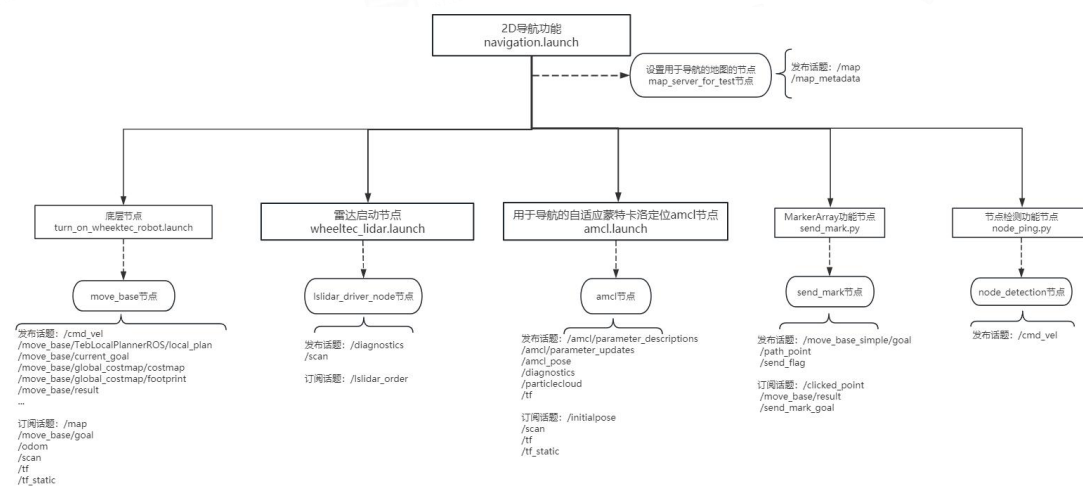


全局路径规划算法切换

4. 功能讲解

2D 导航节点的开启是通过运行 turn_on_wheeltec_robot 功能包路径下的 navigation.launch 文件开启的

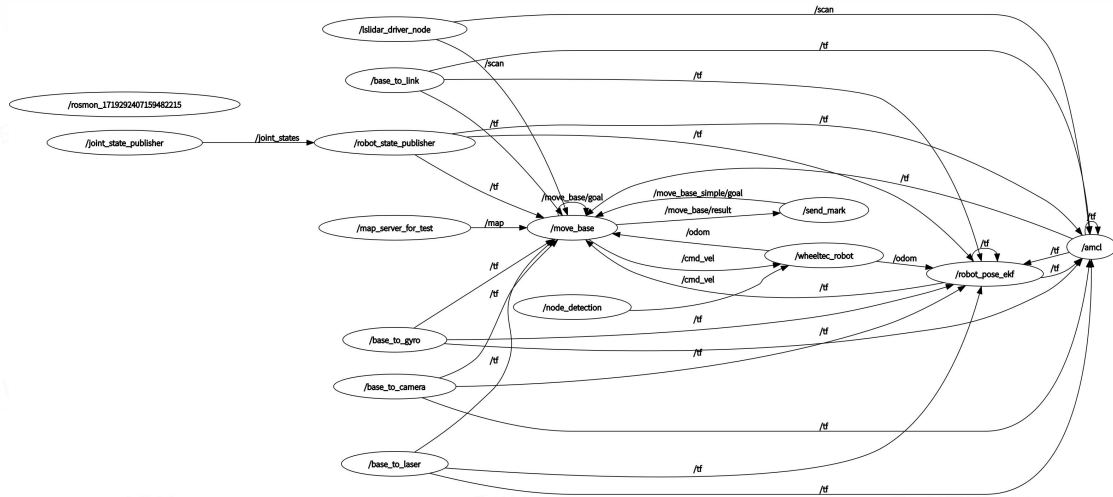
① 启动 2D 导航功能



2D 导航功能启动框架

② 2D 导航功能节点关系

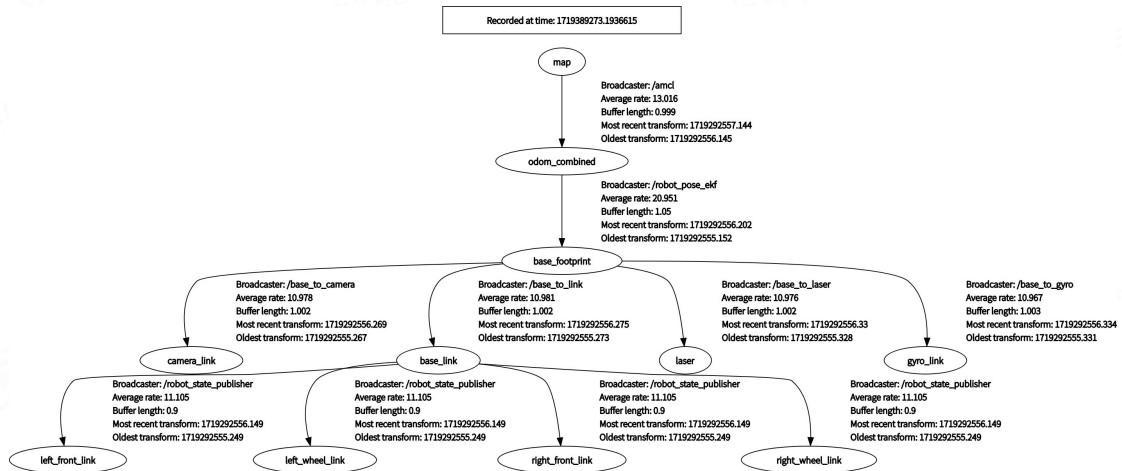
rqt_graph



2D 导航功能节点关系图

③ 2D 导航功能 TF 树

```
roslaunch rqt_tf_tree rqt_tf_tree
```



2D 导航功能 TF 树

④ 2D 导航功能启动文件介绍

```
<launch>
<!-- 开启机器人底层相关节点 同时开启导航功能 -->
<include file="$(find
turn_on_wheeltec_robot)/launch/turn_on_wheeltec_robot.launch">
  <arg name="navigation" default="true"/>
</include>

<!-- turn on lidar 开启雷达 -->
<include file="$(find turn_on_wheeltec_robot)/launch/wheeltec_lidar.launch" />
<!-- 设置需要用于导航的地图 -->
<arg name="map_file" default="$(find
turn_on_wheeltec_robot)/map/WHEELTEC.yaml"/>
<node name="map_server_for_test" pkg="map_server" type="map_server" args="$(arg
```

```
map_file)">
</node>

<!-- 开启用于导航的自适应蒙特卡洛定位 amcl>
<include file="$(find turn_on_wheeltec_robot)/launch/include/amcl.launch" />
<!-- MarkerArray 功能节点 -->
<node name='send_mark' pkg="turn_on_wheeltec_robot" type="send_mark.py">
</node>

</launch>
```

首先与 2D 建图一样，都需要开启底层节点与雷达节点，开启底层节点时传入参数 `navigation`，将参数值置为 `true`，底层节点中根据该参数值变为 `true` 来开启导航算法，导航算法需在 `turn_on_wheeltec_robot.launch` 文件中进行设定。接下来是导航地图的设置，这里如果想设置其他路径或名称的地图可在 `map_file` 参数中对 `default` 值进行修改。

再接下来是开启了用于导航的自适应蒙特卡洛定位，主要作用是提供定位功能以 `tf` 形式输出。

最后是开启多点导航功能节点，与多点导航目标点发布相关的文件为 `turn_on_wheeltec_robot` 路径下的 `send_mark.py` 文件。