

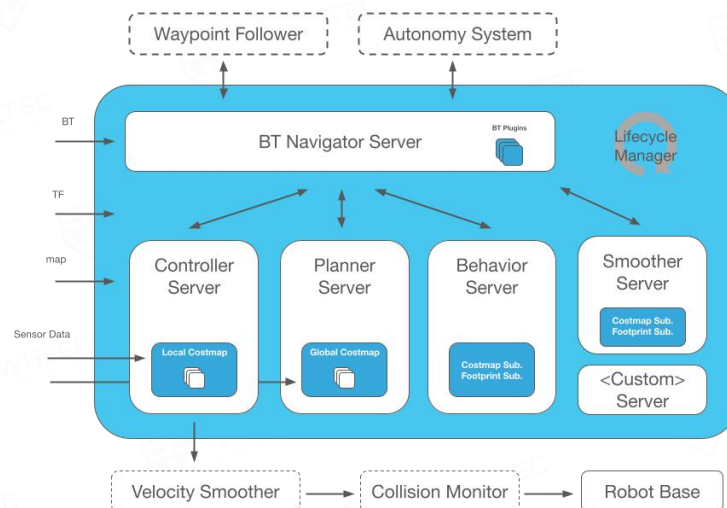
ROS2 2D 导航功能使用与讲解

1. 功能简介

① Nav2 概述

Nav2 项目是 ROS Navigation Stack 的继承者。该项目旨在找到一种让移动机器人从 A 点移动到 B 点的安全方法。它也可以应用于涉及机器人导航的其他应用，例如跟随动态点。这将完成动态路径规划、计算电机速度、避开障碍物和结构恢复行为。

Nav2 的预期输入是符合 REP-105 的 TF 转换，如果使用静态成本地图层，则为地图源，BT XML 文件和任何相关的传感器数据源。然后它将为完整或非完整机器人的电机提供有效的速度命令以跟随。



Nav2 具有以下工具：

- 加载、服务和存储地图（地图服务器）
- 在地图上定位机器人（AMCL）
- 围绕障碍物规划从 A 到 B 的路径（Nav2 规划器）
- 按照路径控制机器人（Nav2 控制器）
- 平滑路径计划更加连续和可行（Nav2 Smoother）
- 将传感器数据转换为世界的成本地图表示（Nav2 Costmap 2D）
- 使用行为树构建复杂的机器人行为（Nav2 行为树和 BT Navigator）
- 计算发生故障时的恢复行为（Nav2 Recoveries）

- 跟随顺序航点 (Nav2 Waypoint Follower)
- 管理服务器的生命周期和看门狗 (Nav2 Lifecycle Manager)
- 启用用户的自定义算法和行为的插件 (Nav2 Core)

NavFn 使用 A* 或 Dijkstra 算法计算从姿势到目标姿势的合适路径。DWB 将使用 DWA 算法来计算遵循路径的控制结果，并使用其自己的几个插件用于轨迹修正。恢复行为包括：等待、旋转、清除成本图和备份。有一套 BT 插件可以调用这些服务器和计算条件。最后，还有一组 Rviz 插件，用于与堆栈交互并控制生命周期。

② ROS1 To ROS2

从 ROS1 到 ROS2，Navigation 的变化是巨大的。而其中最为显著的变化则是使用行为树来组合各个功能模块，以实现搭乐高积木的开发效果。

ROS1 中的导航主要由 move_base 来组合各个模块的功能。在 move_base 中实现了一个状态机，全局路径规划器和局部路径规划器都是以插件的形式在相应的状态中被调用的。P2P 导航的流程是：先在一个状态里调用全局路径规划器，然后进入到另外一个状态传路径给局部路径规划器，并使其执行该路径。

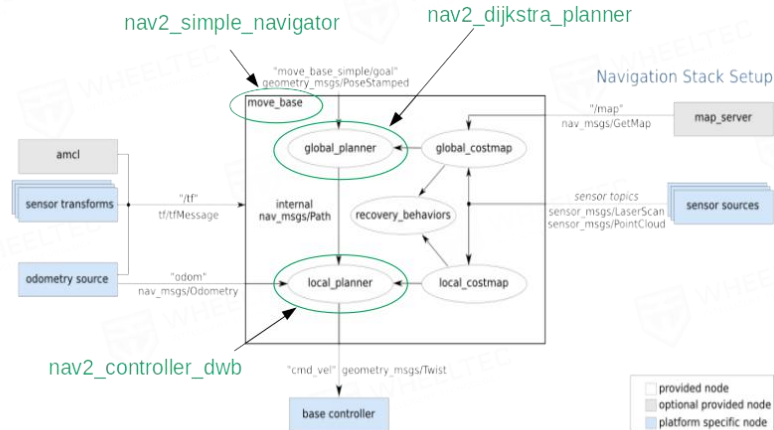
Nav2 不再是一个单一的整体状态机，而是利用动作服务器和 ROS 2 的低延迟、可靠的通信来分离想法。各个功能的组织则有所不同。全局路径规划功能由各个全局路径规划的插件实现，这些插件被加载到 Planner Server 中。可以通过相应的 action 向 Planner Server 请求全局规划的路径。此时 P2P 导航流程为：行为树上的相应节点会向 Planner Server 请求一条规划到目标点的路径。然后另一个树上的节点与 Controller Server 通信，将规划好的路径发给 Controller Server 去执行。

nav2_bt_navigator 在 move_base 顶层替换为 Action 接口，以使用基于树的操作模型完成导航任务。它使用行为树来实现更复杂的状态机并添加行为作为额外的动作服务器。这些行为树是可配置的 XML。

计划、行为和控制器服务器也是 BT 导航器可以调用来计算的操作服务器。所有 3 个服务器都可以托管许多算法的许多插件，每个插件都可以从导航行为树中单独调用以实现特定行为。提供的默认插件是从 ROS 1 移植而来的，即：DWB、NavFn 以及类似的行为，例如旋转和清除成本图。还添加了等待固定持续时间的新行为。这些服务器通过它们的操作服务器从 BT 导航器调用以计算

结果或完成任务。状态由 BT 导航器行为树维护。

所有这些更改使得在启动/运行时用实现相同接口的任何其他算法替换任何这些节点成为可能。



移植功能包：

amcl: 移植到 nav2_amcl

map_server: 移植到 nav2_map_server

nav2_planner: 替换 global_planner, 托管 N 规划器插件

nav2_controller: 替换 local_planner, 托管 N 控制器插件

Navfn: 移植到 nav2_navfn_planner

DWB: 替换 DWA 并移植到 nav2_dwb_controller 元包下的 ROS 2

nav_core: 移植为 nav2_core 并更新接口

costmap_2d: 移植为 nav2_costmap_2d

新功能包：

nav2_bt_navigator: 替换 move_base 状态机

nav2_lifecycle_manager: 处理服务器程序生命周期

nav2_waypoint_follower: 可以通过多个航点来执行复杂的任务

nav2_system_tests: 一组 CI 集成测试和模拟基础教程

nav2_rviz_plugins: 一个 rviz 插件, 用于控制 Navigation2 服务器、命令、

取消和导航

nav2_experimental: 深度强化学习控制器的实验 (和不完整) 工作

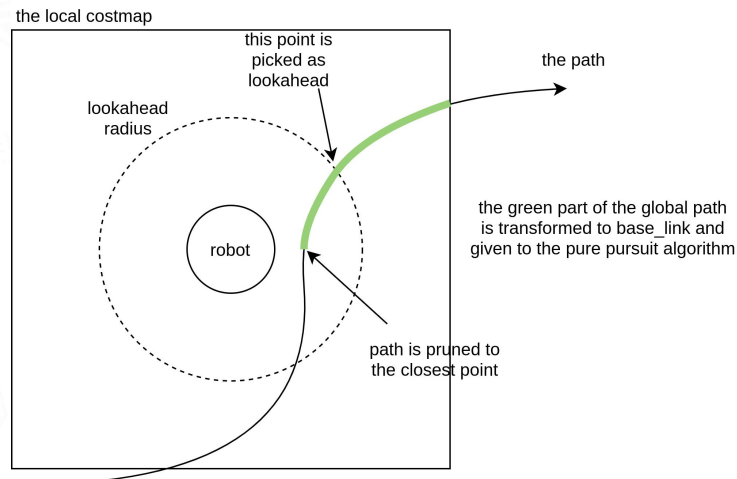
navigation2_behavior_trees: 行为树库的包装器, 用于调用 ROS 动作服务器

③ WHEELTEC NAV2 的路径规划器

WHEELTEC -Nav2 使用 Nav2 Planner 的服务器插件 SmacPlanner 作为规划师，使用 SmacPlannerHybrid 分支，使用 Reeds-shepp 运动模型的 SE2 Hybrid-A* 实现，具有更平滑和多分辨率的查询。支持差速、全向以及阿克曼运动模型。关于 SmacPlanner 的更多信息请查看：

https://github.com/ros-planning/navigation2/tree/main/nav2_smac_planner

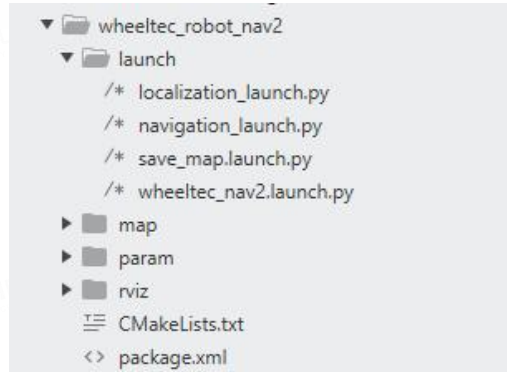
局部轨迹规划器使用 RegulatedPurePursuitController 算法计算导航路径，它实现了纯追踪算法的变体来跟踪路径。这种变体我们称之为受监管的纯追踪算法，因为它对碰撞和线速度有额外的监管条款。它还实现了 Adaptive Pure Pursuit 算法背后的基础知识，以根据当前速度改变前瞻距离。这个插件实现了 nav2_core::Controller 接口，允许它在导航堆栈中用作控制器服务器的动作服务器（controller_server）中的本地轨迹规划器。



该控制器适用于所有类型的机器人，包括差速器、腿式和阿克曼转向车辆。它也可以用于全向平台，但不能充分利用底座的横向运动。如果需要用到横向运动，可以使用 DWB 算法，目前镜像默认使用 RegulatedPurePursuitController 做局部轨迹规划器。

④ WHEELTEC NAV2

wheeltec_nav2 功能包的主要作用是在机器人端实现 2D 导航与其它拓展性功能。



localization_launch.py 是启动机器人在地图中的定位，使用 amcl 蒙特卡洛定位，以及打开 map_server 节点加载地图；

navigation_launch.py 启动导航的一些插件，包括规划器、恢复器和行为树；

在 wheeltec_nav2.launch.py 文件中，除了调用 navigation_launch.py 和 localization_launch.py 文件，还打开了初始化底层控制节点和雷达启动节点，将导航的地图传入 localization_launch.py 中，Nav2 的插件参数和行为树的相关参数传入 navigation_launch.py 文件。在 wheeltec_nav2.launch.py 文件中，如果设置 autostart: = False，则需要单击 RViz 中的开始按钮以初始化节点。同时需要确保 use_sim_time 设置为 False，因为我们要使用系统时间而不是 Gazebo 中的模拟时间。

map 文件夹存放导航所需要的地图，所以导航之前需要先构建地图。

param 文件夹存放 Nav2 的插件参数和行为树的 yaml 文件，子文件夹 wheeltec_param 存放 WHEELTEC 不同车型的插件参数。

2. 使用方法

WHEELTEC NAV2 提供了两种导航方式，分别是点到点导航与路标跟随导航。首先将小车放置在建图起点，再运行导航功能。

① 点到点导航

在进行 SSH 远程登录后的终端运行雷达跟随功能对应的启动文件

（SSH 远程登录操作请查看 ROS2 小车上手操作文档或功能演示视频）

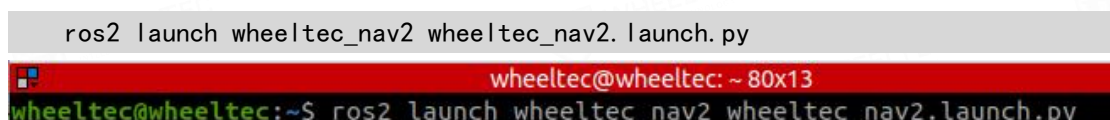


图 1 运行 2D 导航功能

在虚拟机终端开启配置好的 rviz 文件查看建图效果

```
ros2 launch wheeltec_rviz2 wheeltec_rviz.launch.py
```

wheeltec@humble: ~ 92x4

```
wheeltec@humble:~$ ros2 launch wheeltec_rviz2 wheeltec_rviz.launch.py
```

图2 运行rviz可视化工具

成功运行后rviz可视化界面如下图，点击【Nav2 Goal】或【Setgoal】发布导航点

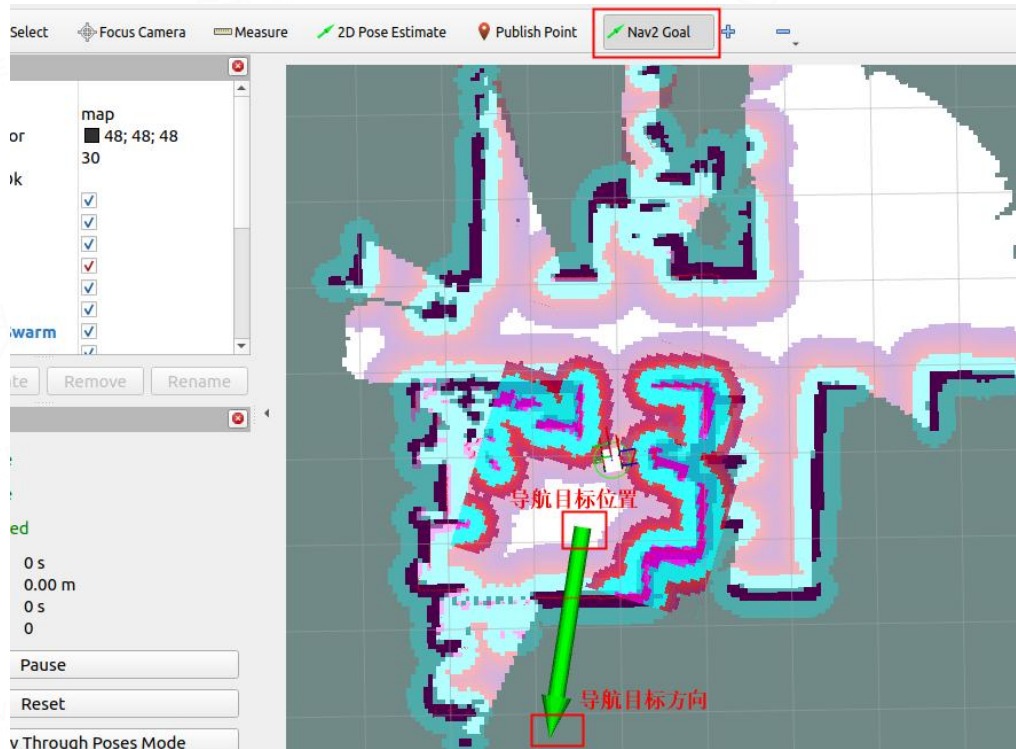


图3 发布导航点

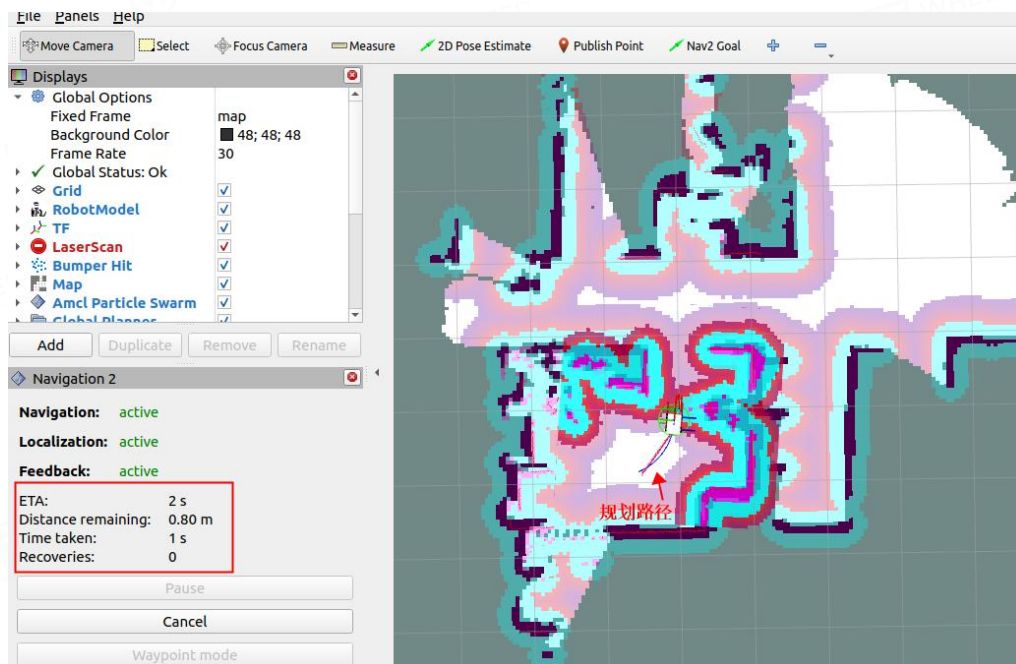
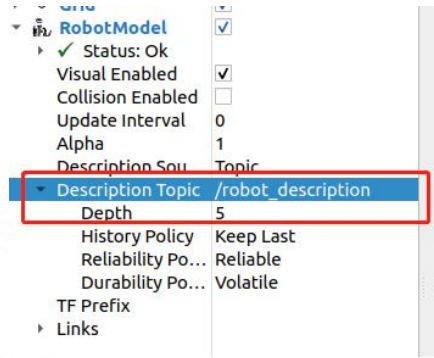


图4 rviz显示效果

在 Nav2 RViz 的左下角方向，用户可以直接通过 RVIZ2 面板找到预计到达时间、距离目标的剩余距离、导航开始后经过的时间以及导航操作期间执行的恢复次数等信息

如果 rviz 没有显示模型，需要检查 RobotModel 中的 Description 有没有选中话题，话题为/robot_description。



② 路标跟随导航

路标跟踪是导航系统的基本功能。它告诉我们的系统如何使用导航到达多个目的地。在 2D 点到点导航的步骤下，运行启动 2D 导航的命令行并打开 rviz2 后，点击左下角的“waypoint mode”切换成路标跟随模式

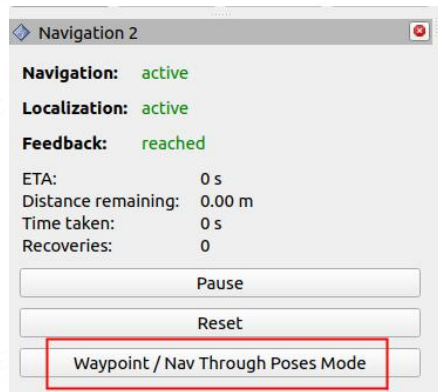


图 5 切换导航模式

到这一步，再点击 rviz2 上方的“Nav2 Goal”开始设定路标位置

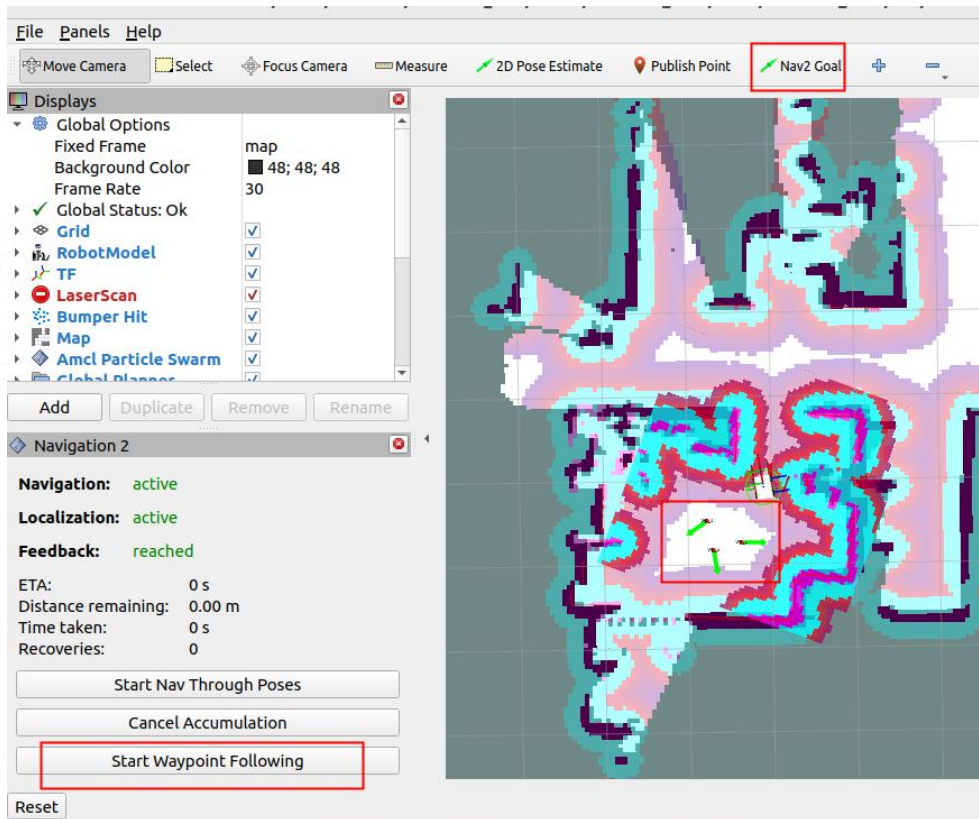


图 6 rviz 可视化工具显示效果

标完位置后，点击右下角的”Start Waypoint Following ”开始实现路标跟随。
途中可以点击 Cancel 取消前往目标点

3. 注意事项

运行 2D 导航时需注意障碍物膨胀后，小车是否能够正常规划路径进行导航，遇到狭窄位置或障碍物阻拦必经路径情况下，小车可能会出现规划失败无法导航的情况。

4. 功能讲解

2D 导航功能的启动，包括设置地图路径、启动底盘与雷达等硬件配件与启动 nav2 导航相关节点，其中启动导航节点主要涉及两个文件：一个是 localization_launch.py，它用于启动机器人在地图中的定位，使用 amcl 蒙特卡洛定位，以及 map_server 节点加载地图；另一个是 navigation_launch.py，它包括启动导航的一些插件，包括规划器、恢复器和行为树。关于 nav2 的解析，感兴趣的朋友可以到网上查找更多详细信息

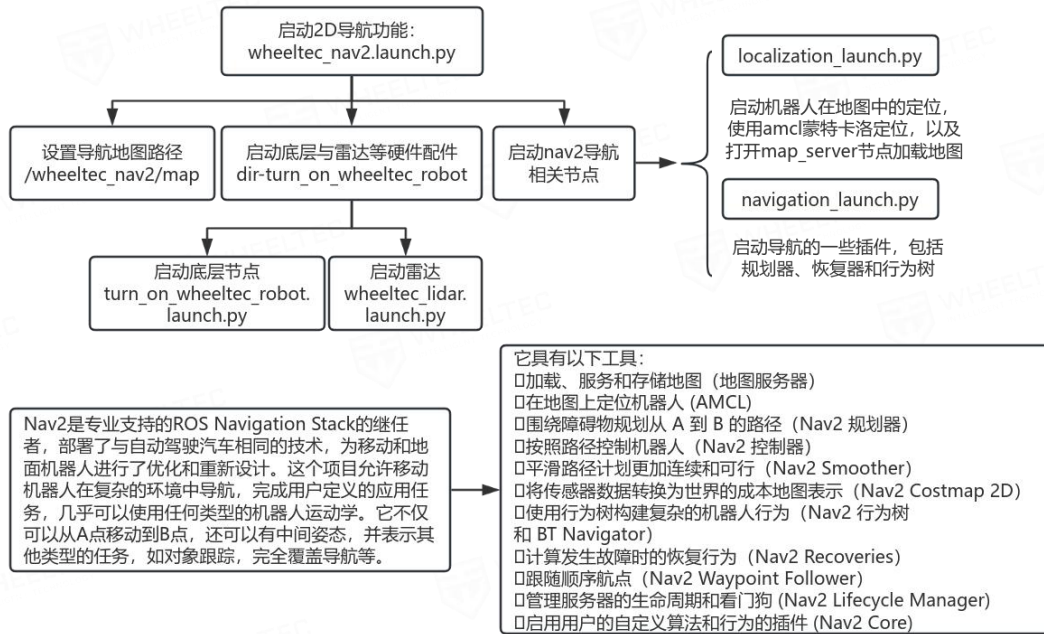


图 7 2D 导航详解

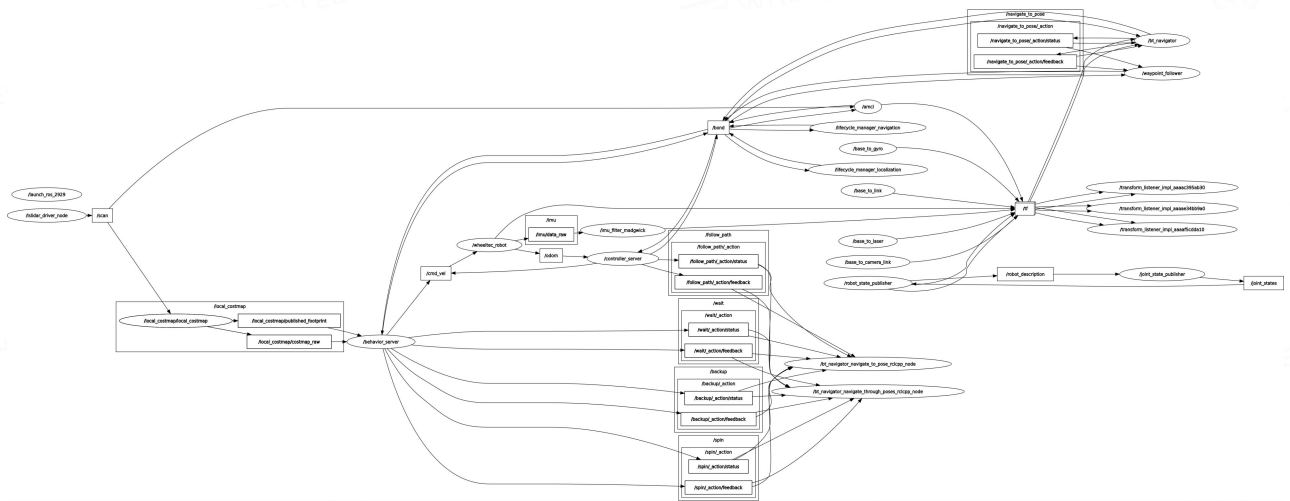


图 8 2D 导航功能对应的 rqt_graph

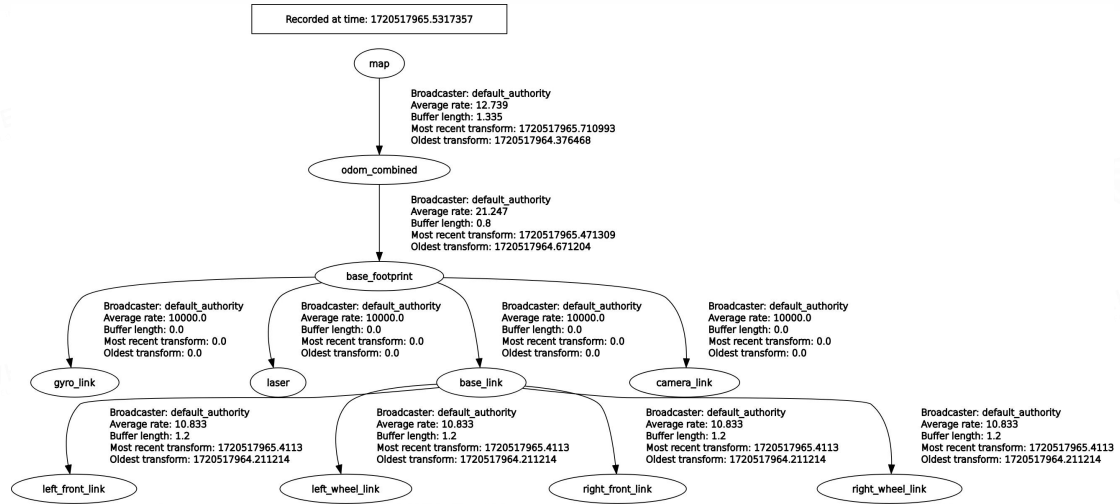


图 9 2D 导航功能对应的 tf 图