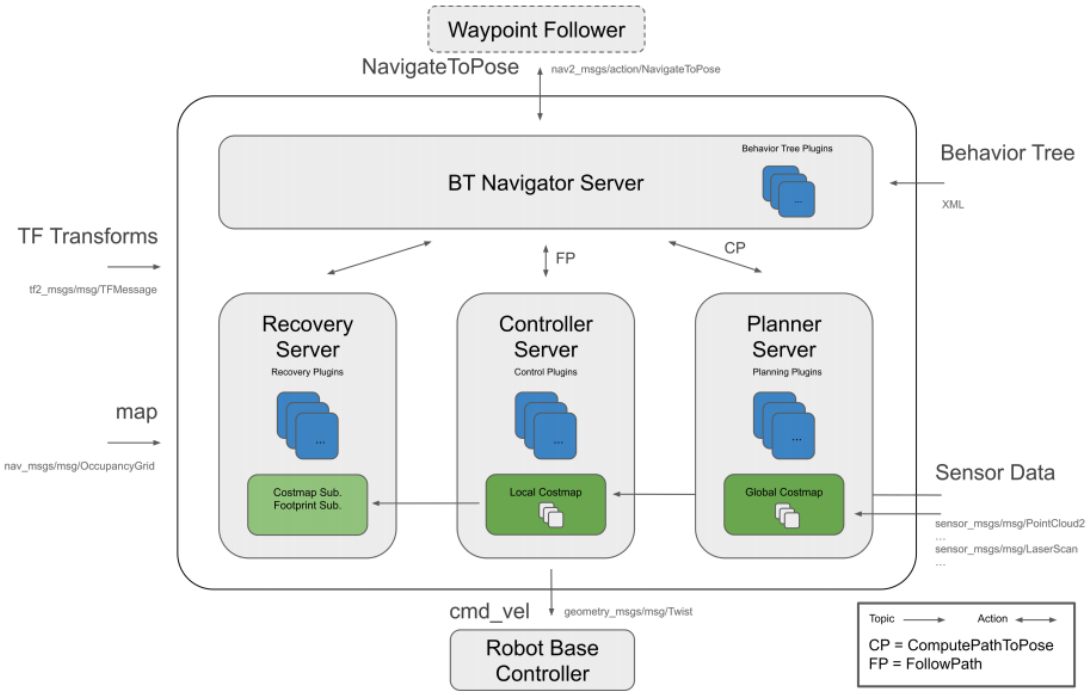
Navigation2导航

1、Navigation2简介

Navigation2整体架构图



Navigation2具有下列工具：

- 加载、提供和存储地图的工具（地图服务器Map Server）
- 在地图上定位机器人的工具 (AMCL)
- 避开障碍物从A点移动到B点的路径规划工具（Nav2 Planner）
- 跟随路径过程中控制机器人的工具（Nav2 Controller）
- 将传感器数据转换为机器人世界中的成本地图表达的工具（Nav2 Costmap 2D）
- 使用行为树构建复杂机器人行为的工具（Nav2 行为树和BT Navigator）
- 发生故障时计算恢复行为的工具（Nav2 Recoveries）
- 跟随顺序航点的工具（Nav2 Waypoint Follower）
- 管理服务器生命周期的工具和看门狗(Nav2 Lifecycle Manager)
- 启用用户自定义算法和行为的插件（Nav2 Core）

Navigation 2（Nav 2）是ROS 2中自带的导航框架，其目的是能够通过一种安全的方式使移动机器人从A点移动到B点。所以，Nav 2可以完成动态路径规划、计算电机速度、避开障碍物和恢复结构等行为。

Nav 2使用行为树（BT，Behavior Trees）调用模块化服务器来完成一个动作。动作可以是计算路径、控制工作（control efforts）、恢复或其他与导航相关的动作。这些动作都是通过动作服务器与行为树（BT）进行通信的独立节点。

资料参考网址：

Navigation2 文档：https://navigation.ros.org/index.html

Navigation2 github：https://github.com/ros-planning/navigation2

Navigation2 对应的论文：https://arxiv.org/pdf/2003.00368.pdf

Navigation2提供的插件：https://navigation.ros.org/plugins/index.html#plugins

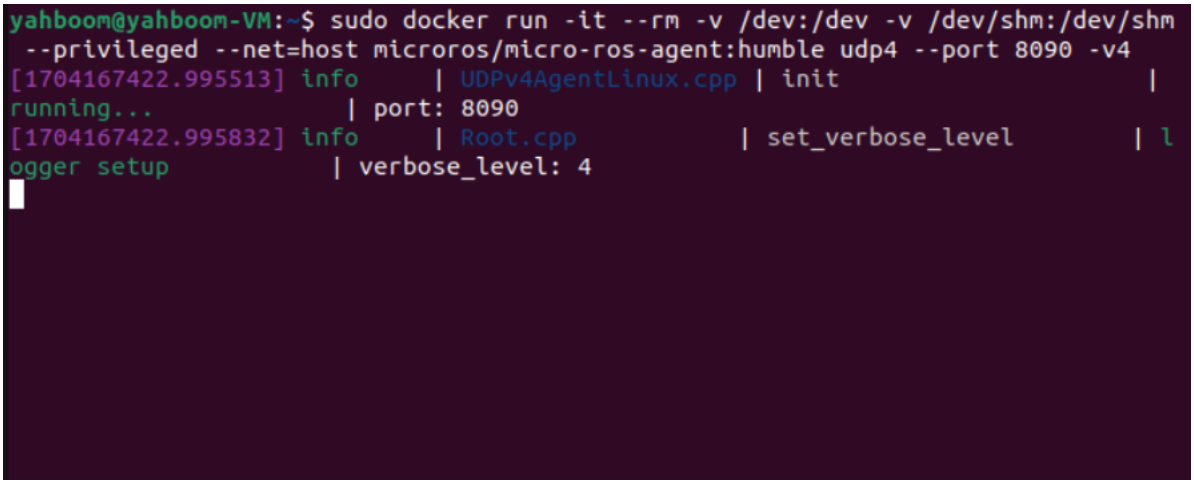
2、程序功能说明

小车连接上代理，运行程序，rviz中会加载地图。在rviz界面中，用【2D Pose Estimate】工具给定小车初始位姿，然后用【2D Goal Pose】工具给定小车一个目标点。小车结合自身环境，会规划出一条路径并且根据规划的路径移动到目的地，期间如果遇到障碍物，会自助避障，到达目的地后停车。

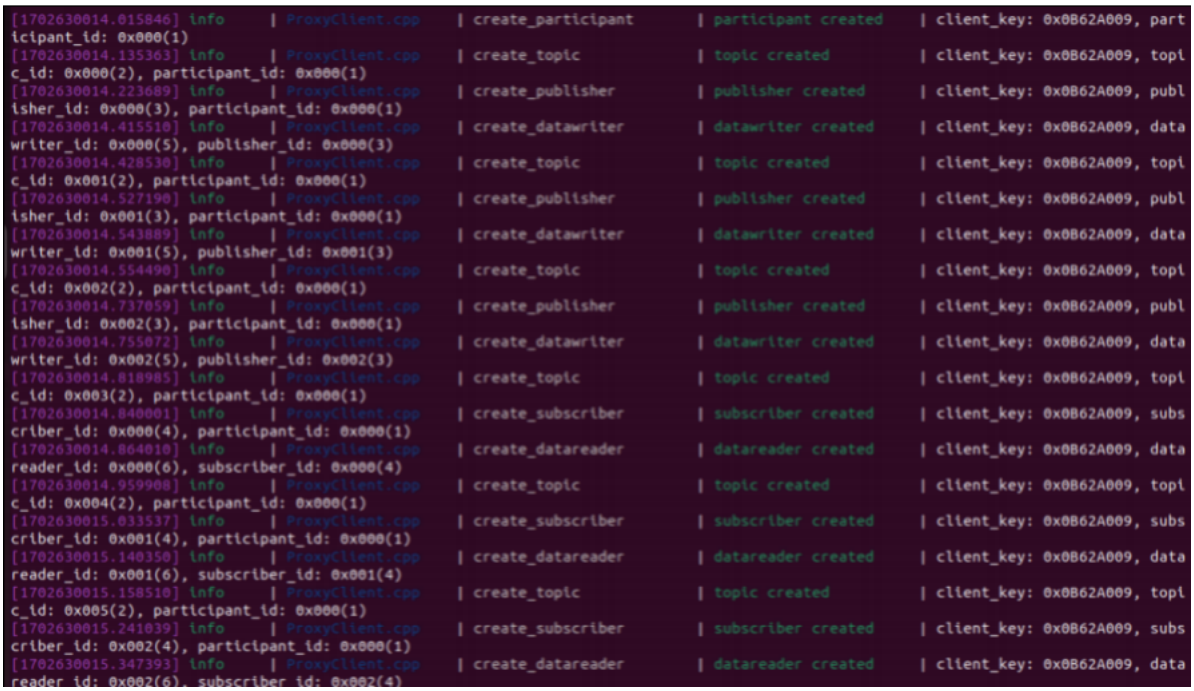
3、启动并连接代理

以配套虚拟机为例，输入以下指令启动代理，

```
sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --
net=host microros/micro-ros-agent:humble udp4 --port 8090 -v4
```



然后，打开小车开关，等待小车连接上代理，连接成功如下图所示，



4、启动程序

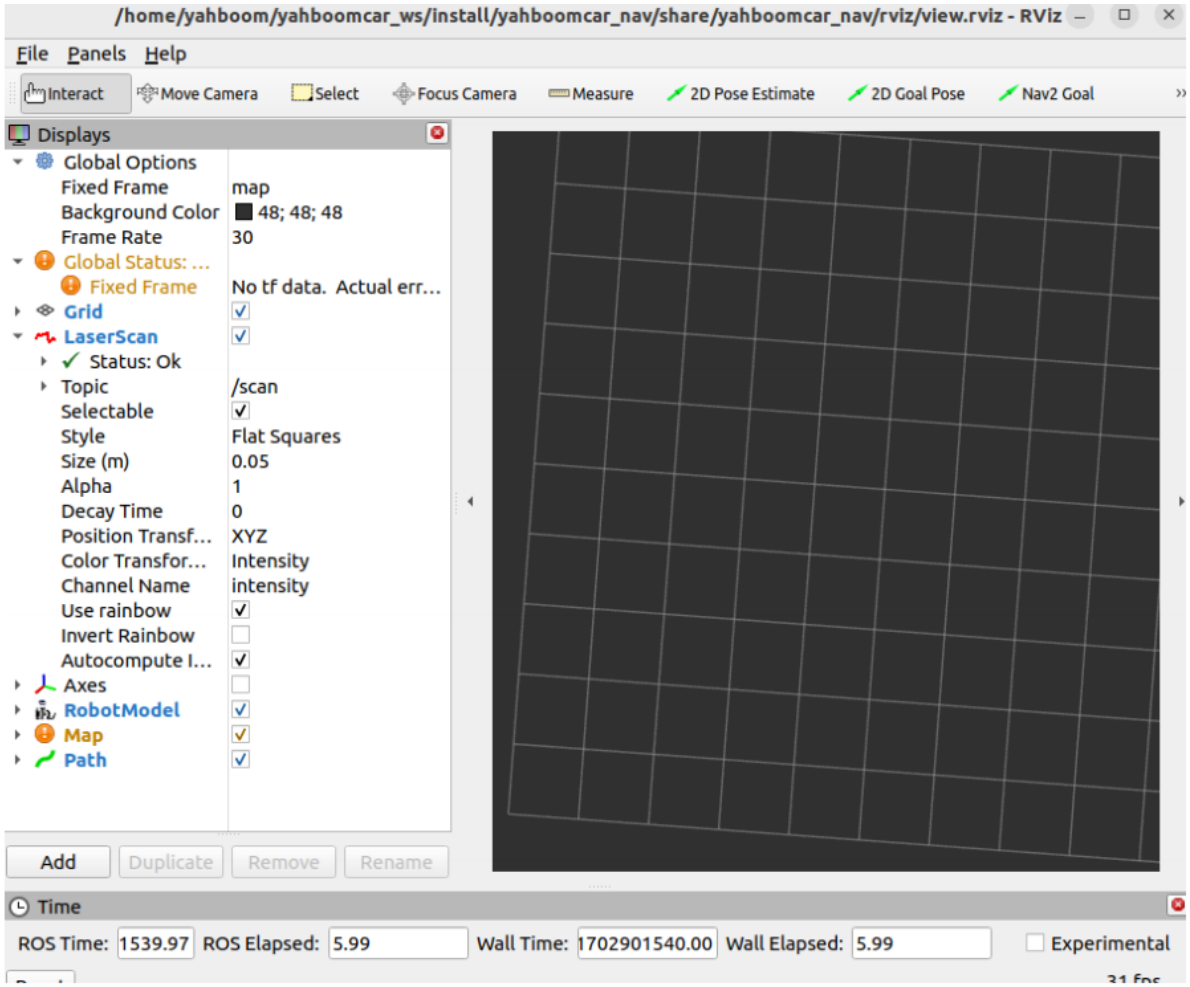
首先启动小车处理底层数据程序，终端输入，

```
ros2 launch yahboomcar_bringup yahboomcar_bringup_launch.py
```

```
[INFO] [imu_filter_madgwick_node-1]: process started with pid [6638]
[INFO] [ekf_node-2]: process started with pid [6640]
[INFO] [static_transform_publisher-3]: process started with pid [6642]
[INFO] [joint_state_publisher-4]: process started with pid [6644]
[INFO] [robot_state_publisher-5]: process started with pid [6646]
[INFO] [static_transform_publisher-6]: process started with pid [6658]
[static_transform_publisher-3] [WARN] [1702865272.944043208] []: Old-style arguments are deprecated; see --help for new-style argume
nts
[static_transform_publisher-6] [WARN] [1702865272.984740987] []: Old-style arguments are deprecated; see --help for new-style argume
nts
[static_transform_publisher-3] [INFO] [1702865272.991057276] [base_link_to_base_imu]: Spinning until stopped - publishing transform
[static_transform_publisher-3] translation: ('-0.002999', '-0.003000', '0.031701')
[static_transform_publisher-3] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-3] from 'base_link' to 'imu_frame'
[static_transform_publisher-6] [INFO] [1702865273.005707993] [static_transform_publisher_JH06Gexf4GRodmgs]: Spinning until stopped -
 publishing transform
[static_transform_publisher-6] translation: ('0.000000', '0.000000', '0.050000')
[static_transform_publisher-6] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-6] from 'base_footprint' to 'base_link'
[robot_state_publisher-5] [WARN] [1702865273.013202438] [kdl_parser]: The root link base_link has an inertia specified in the URDF,
but KDL does not support a root link with an inertia.  As a workaround, you can add an extra dummy link to your URDF.
[robot_state_publisher-5] [INFO] [1702865273.013312806] [robot_state_publisher]: got segment base_link
[robot_state_publisher-5] [INFO] [1702865273.013516195] [robot_state_publisher]: got segment imu_link
[robot_state_publisher-5] [INFO] [1702865273.013524175] [robot_state_publisher]: got segment jq1_Link
[robot_state_publisher-5] [INFO] [1702865273.013528144] [robot_state_publisher]: got segment jq2_Link
[robot_state_publisher-5] [INFO] [1702865273.013531665] [robot_state_publisher]: got segment radar_Link
[robot_state_publisher-5] [INFO] [1702865273.013535185] [robot_state_publisher]: got segment yh_Link
[robot_state_publisher-5] [INFO] [1702865273.013538763] [robot_state_publisher]: got segment yq_Link
[robot_state_publisher-5] [INFO] [1702865273.013542135] [robot_state_publisher]: got segment zh_Link
[robot_state_publisher-5] [INFO] [1702865273.013545612] [robot_state_publisher]: got segment zq_Link
[imu_filter_madgwick_node-1] [INFO] [1702865273.030399479] [imu_filter]: Starting ImuFilter
[imu_filter_madgwick_node-1] [INFO] [1702865273.031826501] [imu_filter]: Using dt computed from message headers
[imu_filter_madgwick_node-1] [INFO] [1702865273.031858361] [imu_filter]: The gravity vector is kept in the IMU message.
[imu_filter_madgwick_node-1] [INFO] [1702865273.032488302] [imu_filter]: Imu filter gain set to 0.100000
[imu_filter_madgwick_node-1] [INFO] [1702865273.032525566] [imu_filter]: Gyro drift bias set to 0.000000
[imu_filter_madgwick_node-1] [INFO] [1702865273.032531441] [imu_filter]: Magnetometer bias values: 0.000000 0.000000 0.000000
[imu_filter_madgwick_node-1] [INFO] [1702865273.053298796] [imu_filter]: First IMU message received.
[joint_state_publisher-4] [INFO] [1702865273.282975810] [joint_state_publisher]: Waiting for robot_description to be published on th
e robot_description topic...
```

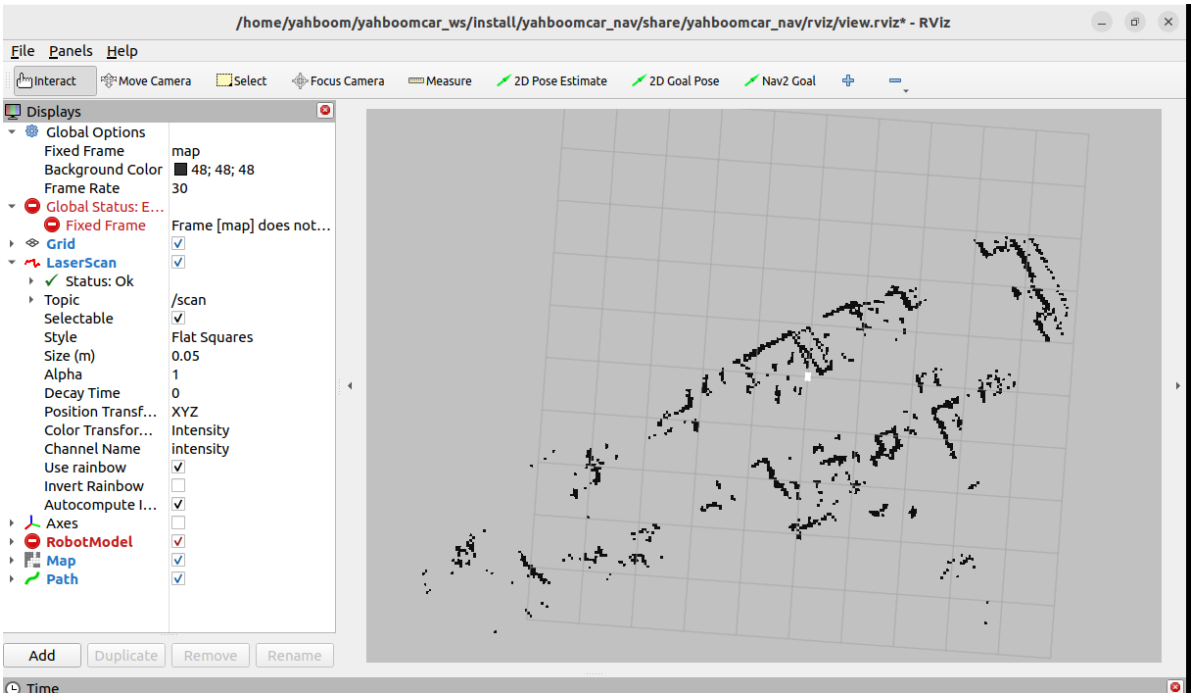
然后，启动rviz，可视化导航，终端输入

```
ros2 launch yahboomcar_nav display_launch.py
```

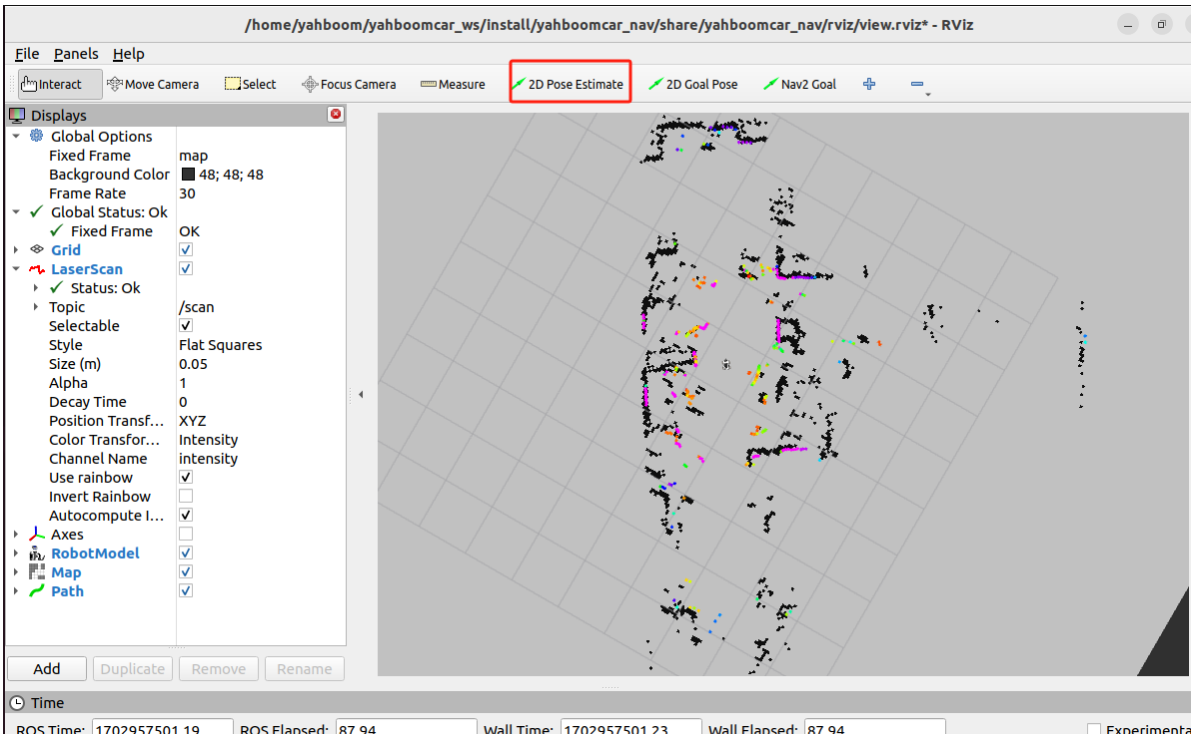


此时还没有显示地图加载，因为还没有启动导航的程序，所以没有地图加载。接下来运行导航节点，终端输入，

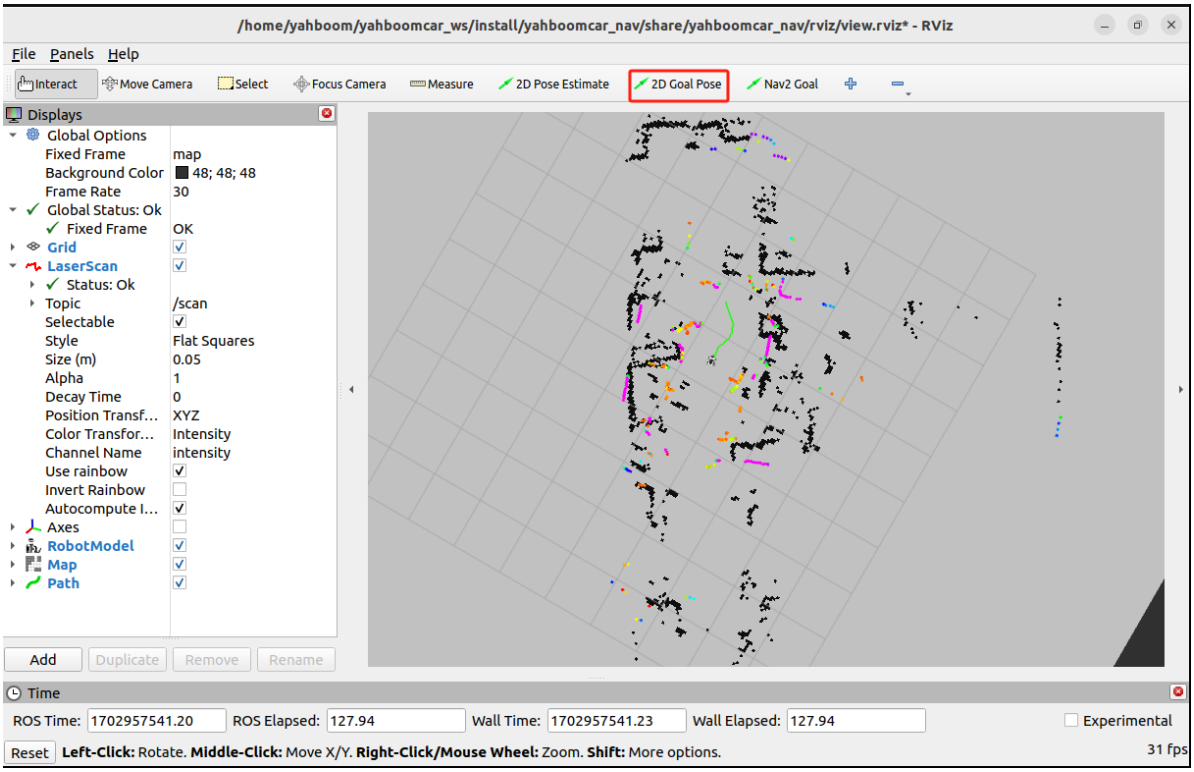
```
ros2 launch  yahboomcar_nav navigation_dwb_launch.py
```



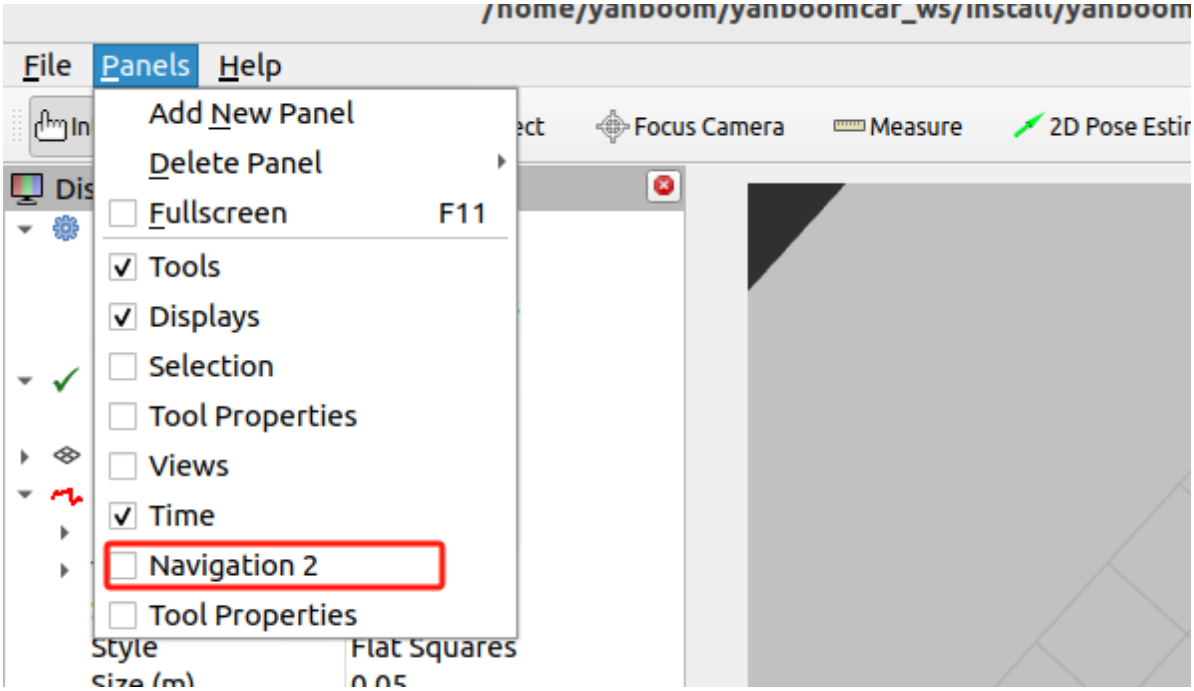
此时可以看到地图加载进去了，然后我们点击【2D Pose Estimate】，给小车设置初始位姿，根据小车在实际环境中的位置，在rviz中用鼠标点击拖动，小车模型移动我们设置的位置。如下图所示，雷达扫描的区域与实际障碍物大致重合则表示位姿准确。



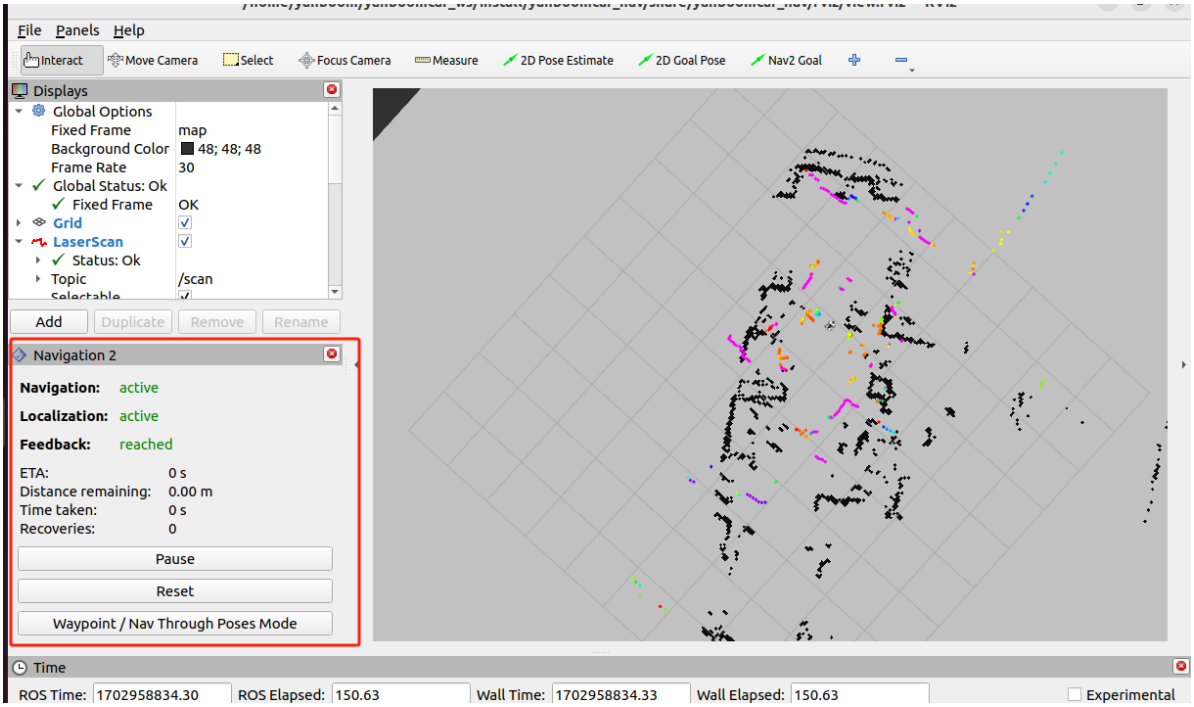
单点导航，点击【2D Goal Pose】工具，然后在rviz中选择一个目标点，小车结合周围的情况，规划出一条路径并且沿着路径移动到目标点。



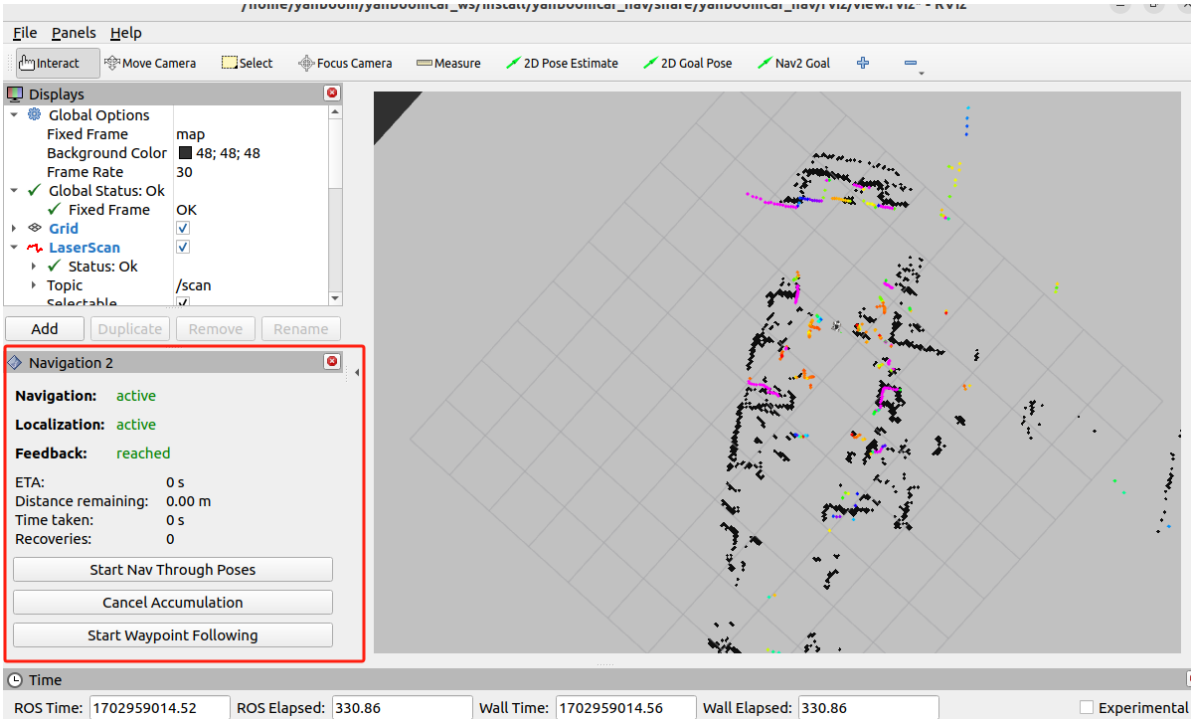
多点导航，需要把nav2的插件添加进来，



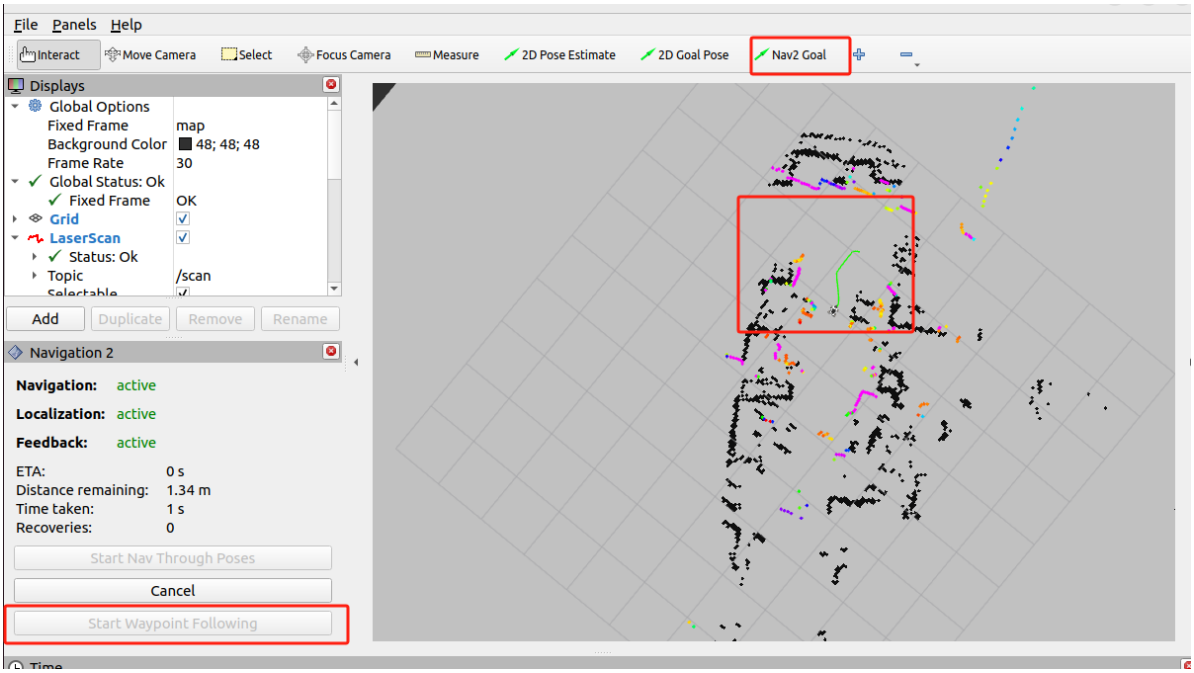
添加后，rviz显示如下，



然后点击【Waypoint/Nav Through Poses Mode】，



使用rivz工具栏中的【Nav2 Goal】给定任意的目标点，然后点击【Start Waypoint Following】开始规划路径导航。小车会根据选的点的先后顺序，到了目标点后会自动前往下一个点，无需进行操作。达到最后一个点后，小车停车等待下一个指令。



5、查看节点通讯图

终端输入，

```
ros2 run rqt_graph rqt_graph
```

如果一开始没有显示，选择【Nodes/Topics(all)】，然后点击左上角的刷新按钮。

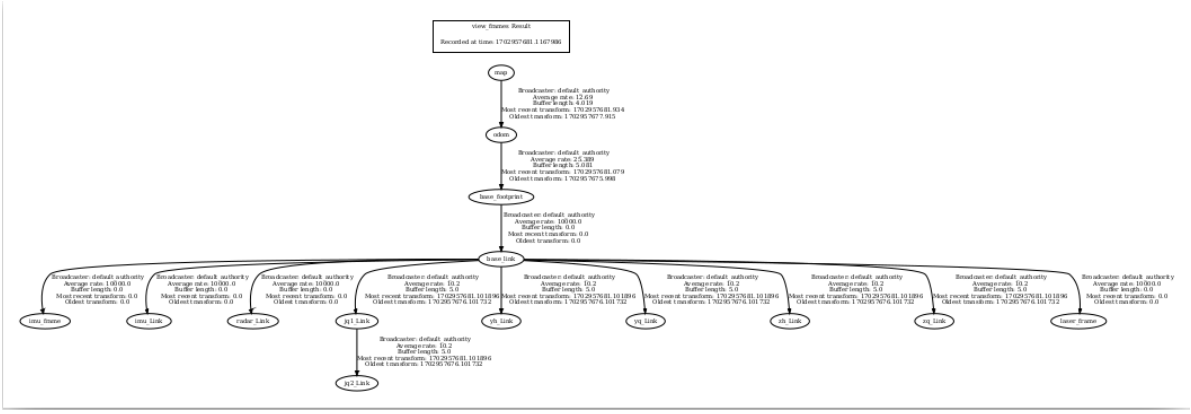
6、查看TF树

终端输入，

```
ros2 run tf2_tools view_frames
```

```
yahboom@yahboom-VM:~$ ros2 run tf2_tools view_frames
[INFO] [1702957676.100327021] [view_frames]: Listening to tf data for 5.0 seconds...
[INFO] [1702957681.102788842] [view_frames]: Generating graph in frames.pdf file...
[INFO] [1702957681.107154766] [view_frames]: Result:tf2_msgs.srv.FrameGraph_Response(frame_yaml="base_footprint: \n  parent: 'odom'\
n  broadcaster: 'default_authority'\n  rate: 25.389\n  most_recent_transform: 1702957681.079000\n  oldest_transform: 1702957675.9980
00\n  buffer_length: 5.081\nodom: \n  parent: 'map'\n  broadcaster: 'default_authority'\n  rate: 12.690\n  most_recent_transform: 17
02957681.934000\n  oldest_transform: 1702957677.915000\n  buffer_length: 4.019\nimu_frame: \n  parent: 'base_link'\n  broadcaster: '
default_authority'\n  rate: 10000.000\n  most_recent_transform: 0.000000\n  oldest_transform: 0.000000\n  buffer_length: 0.000\nbase
_link: \n  parent: 'base_footprint'\n  broadcaster: 'default_authority'\n  rate: 10000.000\n  most_recent_transform: 0.000000\n  old
est_transform: 0.000000\n  buffer_length: 0.000\nimu_Link: \n  parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate: 1000
0.000\n  most_recent_transform: 0.000000\n  oldest_transform: 0.000000\n  buffer_length: 0.000\nradar_Link: \n  parent: 'base_link'\
n  broadcaster: 'default_authority'\n  rate: 10000.000\n  most_recent_transform: 0.000000\n  oldest_transform: 0.000000\n  buffer_le
ngth: 0.000\njq1_Link: \n  parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate: 10.200\n  most_recent_transform: 1702957
681.101896\n  oldest_transform: 1702957676.101732\n  buffer_length: 5.000\njq2_Link: \n  parent: 'jq1_Link'\n  broadcaster: 'default
_authority'\n  rate: 10.200\n  most_recent_transform: 1702957681.101896\n  oldest_transform: 1702957676.101732\n  buffer_length: 5.0
00\nyh_Link: \n  parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate: 10.200\n  most_recent_transform: 1702957681.101896
\n  oldest_transform: 1702957676.101732\n  buffer_length: 5.000\nyq_Link: \n  parent: 'base_link'\n  broadcaster: 'default_authority
'\n  rate: 10.200\n  most_recent_transform: 1702957681.101896\n  oldest_transform: 1702957676.101732\n  buffer_length: 5.000\nzh_Lin
k: \n  parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate: 10.200\n  most_recent_transform: 1702957681.101896\n  oldest
_transform: 1702957676.101732\n  buffer_length: 5.000\nzq_Link: \n  parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate:
 10.200\n  most_recent_transform: 1702957681.101896\n  oldest_transform: 1702957676.101732\n  buffer_length: 5.000\nlaser_frame: \n
 parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate: 10000.000\n  most_recent_transform: 0.000000\n  oldest_transform:
0.000000\n  buffer_length: 0.000\n")
```

运行完毕后，会在终端的目录下生成两个文件分别是.gv和.pdf文件，其中的pdf文件就是TF树。



7、代码解析

这里只说明导航的navigation_dwb_launch.py，这个文件路径是，

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/launch
```

navigation_dwb_launch.py，

```
import os
from ament_index_python.packages import get_package_share_directory
from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument
from launch.actions import IncludeLaunchDescription
from launch.launch_description_sources import PythonLaunchDescriptionSource
from launch.substitutions import LaunchConfiguration
from launch_ros.actions import Node

def generate_launch_description():
    package_path = get_package_share_directory('yahboomcar_nav')
    nav2_bringup_dir = get_package_share_directory('nav2_bringup')

    use_sim_time = LaunchConfiguration('use_sim_time', default='false')
    map_yaml_path = LaunchConfiguration(
        'maps', default=os.path.join(package_path, 'maps', 'yahboom_map.yaml'))
    nav2_param_path = LaunchConfiguration('params_file', default=os.path.join(
        package_path, 'params', 'dwb_nav_params.yaml'))

    return LaunchDescription([
        DeclareLaunchArgument('use_sim_time', default_value=use_sim_time,
                              description='Use simulation (Gazebo) clock if
true'),
        DeclareLaunchArgument('maps', default_value=map_yaml_path,
                              description='Full path to map file to load'),
        DeclareLaunchArgument('params_file', default_value=nav2_param_path,
                              description='Full path to param file to load'),

        IncludeLaunchDescription(
            PythonLaunchDescriptionSource(
                [nav2_bringup_dir, '/launch', '/bringup_launch.py']),
            launch_arguments={
                'map': map_yaml_path,
                'use_sim_time': use_sim_time,
                'params_file': nav2_param_path}.items(),
```

```
        ),
        Node(
            package='tf2_ros',
            executable='static_transform_publisher',
            name='base_link_to_base_laser',
            arguments=['-0.0046412', '0' ,
'0.094079','0','0','0','base_link','laser_frame']
        ),
        Node(
            package='yahboomcar_nav',
            executable='stop_car'
        )
    ])
```

这里启动了以下几个节点：

- base_link_to_base_laser：发布静态的TF变换；
- stop_car：停车节点，ctrl c退出程序后，会发布停车速度给到小车；
- bringup_launch.py：启动导航的launch文件，文件位于，`/opt/ros/humble/share/nav2_bringup/launch`

另外还加载了一个导航参数配置文件dwb_nav_params.yaml和加载地图文件yahboom_map.yaml，导航参数表的位于，

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/params
```

地图文件位于，

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps
```

dwb_nav_params.yaml，

```
amcl:
  ros__parameters:
    use_sim_time: False
    alpha1: 0.2
    alpha2: 0.2
    alpha3: 0.2
    alpha4: 0.2
    alpha5: 0.2
    base_frame_id: "base_footprint"
    beam_skip_distance: 0.5
    beam_skip_error_threshold: 0.9
    beam_skip_threshold: 0.3
    do_beamskip: false
    global_frame_id: "map"
    lambda_short: 0.1
    laser_likelihood_max_dist: 2.0
    laser_max_range: 100.0
    laser_min_range: -1.0
    laser_model_type: "likelihood_field"
    max_beams: 60
    max_particles: 2000
    min_particles: 500
```

```
    odom_frame_id: "odom"
    pf_err: 0.05
    pf_z: 0.99
    recovery_alpha_fast: 0.0
    recovery_alpha_slow: 0.0
    resample_interval: 1
    robot_model_type: "nav2_amcl::DifferentialMotionModel"
    save_pose_rate: 0.5
    sigma_hit: 0.2
    tf_broadcast: true
    transform_tolerance: 1.0
    update_min_a: 0.2
    update_min_d: 0.25
    z_hit: 0.5
    z_max: 0.05
    z_rand: 0.5
    z_short: 0.05
    scan_topic: scan

bt_navigator:
  ros__parameters:
    use_sim_time: False
    global_frame: map
    robot_base_frame: base_link
    odom_topic: /odom
    bt_loop_duration: 10
    default_server_timeout: 20
    default_bt_xml_filename: "navigate_to_pose_w_replanning_and_recovery.xml"
    # 'default_nav_through_poses_bt_xml' and 'default_nav_to_pose_bt_xml' are use
defaults:
    # nav2_bt_navigator/navigate_to_pose_w_replanning_and_recovery.xml
    # nav2_bt_navigator/navigate_through_poses_w_replanning_and_recovery.xml
    # They can be set here or via a RewrittenYaml remap from a parent launch file
to Nav2.
    plugin_lib_names:
      - nav2_compute_path_to_pose_action_bt_node
      - nav2_compute_path_through_poses_action_bt_node
      - nav2_smooth_path_action_bt_node
      - nav2_follow_path_action_bt_node
      - nav2_spin_action_bt_node
      - nav2_wait_action_bt_node
      - nav2_assisted_teleop_action_bt_node
      - nav2_back_up_action_bt_node
      - nav2_drive_on_heading_bt_node
      - nav2_clear_costmap_service_bt_node
      - nav2_is_stuck_condition_bt_node
      - nav2_goal_reached_condition_bt_node
      - nav2_goal_updated_condition_bt_node
      - nav2_globally_updated_goal_condition_bt_node
      - nav2_is_path_valid_condition_bt_node
      - nav2_initial_pose_received_condition_bt_node
      - nav2_reinitialize_global_localization_service_bt_node
      - nav2_rate_controller_bt_node
      - nav2_distance_controller_bt_node
      - nav2_speed_controller_bt_node
```

```
      - nav2_truncate_path_action_bt_node
      - nav2_truncate_path_local_action_bt_node
      - nav2_goal_updater_node_bt_node
      - nav2_recovery_node_bt_node
      - nav2_pipeline_sequence_bt_node
      - nav2_round_robin_node_bt_node
      - nav2_transform_available_condition_bt_node
      - nav2_time_expired_condition_bt_node
      - nav2_path_expiring_timer_condition
      - nav2_distance_traveled_condition_bt_node
      - nav2_single_trigger_bt_node
      - nav2_goal_updated_controller_bt_node
      - nav2_is_battery_low_condition_bt_node
      - nav2_navigate_through_poses_action_bt_node
      - nav2_navigate_to_pose_action_bt_node
      - nav2_remove_passed_goals_action_bt_node
      - nav2_planner_selector_bt_node
      - nav2_controller_selector_bt_node
      - nav2_goal_checker_selector_bt_node
      - nav2_controller_cancel_bt_node
      - nav2_path_longer_on_approach_bt_node
      - nav2_wait_cancel_bt_node
      - nav2_spin_cancel_bt_node
      - nav2_back_up_cancel_bt_node
      - nav2_assisted_teleop_cancel_bt_node
      - nav2_drive_on_heading_cancel_bt_node
      - nav2_is_battery_charging_condition_bt_node

bt_navigator_navigate_through_poses_rclcpp_node:
  ros__parameters:
    use_sim_time: False

bt_navigator_navigate_to_pose_rclcpp_node:
  ros__parameters:
    use_sim_time: False

controller_server:
  ros__parameters:
    use_sim_time: False
    controller_frequency: 20.0
    min_x_velocity_threshold: 0.001
    min_y_velocity_threshold: 0.5
    min_theta_velocity_threshold: 0.001
    failure_tolerance: 0.3
    progress_checker_plugin: "progress_checker"
    goal_checker_plugins: ["general_goal_checker"] # "precise_goal_checker"
    controller_plugins: ["FollowPath"]

    # Progress checker parameters
    progress_checker:
      plugin: "nav2_controller::SimpleProgressChecker"
      required_movement_radius: 0.5
      movement_time_allowance: 10.0
    # Goal checker parameters
    #precise_goal_checker:
```

```
    #  plugin: "nav2_controller::SimpleGoalChecker"
    #  xy_goal_tolerance: 0.25
    #  yaw_goal_tolerance: 0.25
    #  stateful: True
    general_goal_checker:
      stateful: True
      plugin: "nav2_controller::SimpleGoalChecker"
      xy_goal_tolerance: 0.25
      yaw_goal_tolerance: 0.25
    # DWB parameters
    FollowPath:
      plugin: "dwb_core::DWBLocalPlanner"
      debug_trajectory_details: True
      min_vel_x: -0.20
      min_vel_y: 0.0
      max_vel_x: 0.30
      max_vel_y: 0.0
      max_vel_theta: 1.0
      min_speed_xy: -0.20
      max_speed_xy: 0.30
      min_speed_theta: -0.5
      # Add high threshold velocity for turtlebot 3 issue.
      # https://github.com/ROBOTIS-GIT/turtlebot3_simulations/issues/75
      acc_lim_x: 2.5
      acc_lim_y: 0.0
      acc_lim_theta: 3.2
      decel_lim_x: -2.5
      decel_lim_y: 0.0
      decel_lim_theta: -3.2
      vx_samples: 20
      vy_samples: 5
      vtheta_samples: 20
      sim_time: 1.7
      linear_granularity: 0.05
      angular_granularity: 0.025
      transform_tolerance: 0.2
      xy_goal_tolerance: 0.25
      trans_stopped_velocity: 0.25
      short_circuit_trajectory_evaluation: True
      stateful: True
      critics: ["RotateToGoal", "Oscillation", "BaseObstacle", "GoalAlign",
"PathAlign", "PathDist", "GoalDist"]
      BaseObstacle.scale: 0.02
      PathAlign.scale: 32.0
      PathAlign.forward_point_distance: 0.1
      GoalAlign.scale: 24.0
      GoalAlign.forward_point_distance: 0.1
      PathDist.scale: 32.0
      GoalDist.scale: 24.0
      RotateToGoal.scale: 32.0
      RotateToGoal.slowing_factor: 5.0
      RotateToGoal.lookahead_time: -1.0

local_costmap:
  local_costmap:
```

```yaml
    ros__parameters:
      update_frequency: 5.0
      publish_frequency: 2.0
      global_frame: odom
      robot_base_frame: base_link
      use_sim_time: False
      rolling_window: true
      width: 3
      height: 3
      resolution: 0.05
      robot_radius: 0.22
      plugins: ["voxel_layer", "inflation_layer"]
      inflation_layer:
        plugin: "nav2_costmap_2d::InflationLayer"
        cost_scaling_factor: 3.0
        inflation_radius: 0.55
      voxel_layer:
        plugin: "nav2_costmap_2d::VoxelLayer"
        enabled: True
        publish_voxel_map: True
        origin_z: 0.0
        z_resolution: 0.05
        z_voxels: 16
        max_obstacle_height: 2.0
        mark_threshold: 0
        observation_sources: scan
        scan:
          topic: /scan
          max_obstacle_height: 2.0
          clearing: True
          marking: True
          data_type: "LaserScan"
          raytrace_max_range: 3.0
          raytrace_min_range: 0.0
          obstacle_max_range: 2.5
          obstacle_min_range: 0.0
      static_layer:
        plugin: "nav2_costmap_2d::StaticLayer"
        map_subscribe_transient_local: True
      always_send_full_costmap: True

global_costmap:
  global_costmap:
    ros__parameters:
      update_frequency: 1.0
      publish_frequency: 1.0
      global_frame: map
      robot_base_frame: base_link
      use_sim_time: False
      robot_radius: 0.22
      resolution: 0.05
      track_unknown_space: true
      plugins: ["static_layer", "obstacle_layer", "inflation_layer"]
      obstacle_layer:
        plugin: "nav2_costmap_2d::ObstacleLayer"
```

```
        enabled: True
        observation_sources: scan
        scan:
          topic: /scan
          max_obstacle_height: 2.0
          clearing: True
          marking: True
          data_type: "LaserScan"
          raytrace_max_range: 3.0
          raytrace_min_range: 0.0
          obstacle_max_range: 2.5
          obstacle_min_range: 0.0
      static_layer:
        plugin: "nav2_costmap_2d::StaticLayer"
        map_subscribe_transient_local: True
      inflation_layer:
        plugin: "nav2_costmap_2d::InflationLayer"
        cost_scaling_factor: 3.0
        inflation_radius: 0.55
      always_send_full_costmap: True

map_server:
  ros__parameters:
    use_sim_time: False
    # Overridden in launch by the "map" launch configuration or provided default
value.
    # To use in yaml, remove the default "map" value in the
tb3_simulation_launch.py file & provide full path to map below.
    yaml_filename: ""

map_saver:
  ros__parameters:
    use_sim_time: False
    save_map_timeout: 5.0
    free_thresh_default: 0.25
    occupied_thresh_default: 0.65
    map_subscribe_transient_local: True

planner_server:
  ros__parameters:
    expected_planner_frequency: 20.0
    use_sim_time: False
    planner_plugins: ["GridBased"]
    GridBased:
      plugin: "nav2_navfn_planner/NavfnPlanner"
      tolerance: 0.5
      use_astar: false
      allow_unknown: true

smoother_server:
  ros__parameters:
    use_sim_time: False
    smoother_plugins: ["simple_smoother"]
    simple_smoother:
      plugin: "nav2_smoother::SimpleSmoother"
```

```
      tolerance: 1.0e-10
      max_its: 1000
      do_refinement: False

behavior_server:
  ros__parameters:
    costmap_topic: local_costmap/costmap_raw
    footprint_topic: local_costmap/published_footprint
    cycle_frequency: 10.0
    behavior_plugins: ["spin", "backup", "drive_on_heading", "assisted_teleop",
"wait"]
    spin:
      plugin: "nav2_behaviors/Spin"
    backup:
      plugin: "nav2_behaviors/BackUp"
    drive_on_heading:
      plugin: "nav2_behaviors/DriveOnHeading"
    wait:
      plugin: "nav2_behaviors/Wait"
    assisted_teleop:
      plugin: "nav2_behaviors/AssistedTeleop"
    global_frame: odom
    robot_base_frame: base_link
    transform_tolerance: 0.1
    use_sim_time: False
    simulate_ahead_time: 2.0
    max_rotational_vel: 1.0
    min_rotational_vel: 0.4
    rotational_acc_lim: 3.2

robot_state_publisher:
  ros__parameters:
    use_sim_time: False

waypoint_follower:
  ros__parameters:
    use_sim_time: False
    loop_rate: 20
    stop_on_failure: false
    waypoint_task_executor_plugin: "wait_at_waypoint"
    wait_at_waypoint:
      plugin: "nav2_waypoint_follower::WaitAtWaypoint"
      enabled: True
      waypoint_pause_duration: 200

velocity_smoother:
  ros__parameters:
    use_sim_time: False
    smoothing_frequency: 20.0
    scale_velocities: False
    feedback: "OPEN_LOOP"
    max_velocity: [0.26, 0.0, 1.0]
    min_velocity: [-0.26, 0.0, -1.0]
    max_accel: [2.5, 0.0, 3.2]
    max_decel: [-2.5, 0.0, -3.2]
```

```
    odom_topic: "odom"
    odom_duration: 0.1
    deadband_velocity: [0.0, 0.0, 0.0]
    velocity_timeout: 1.0
```

该参数表配置了导航launch文件中，启动的每个节点需要的参数。