Navigation2 single-point navigation avoid

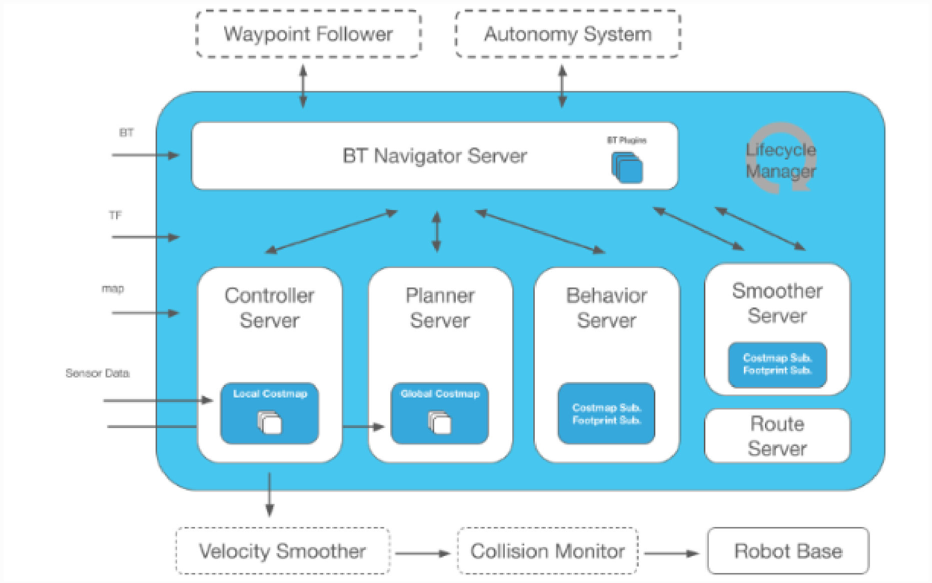
1. Program Functionality

After running the program, a map will load in rviz. In the rviz interface, use the [2D Pose Estimate] tool to set the car's initial pose, and then use the [2D Goal Pose] tool to set a target point. The car will then calculate a path based on its surroundings and move to its destination. If it encounters obstacles, it will automatically avoid them and stop at its destination.

2. Navigation2 Introduction

2.1 Introduction

Navigation2 Overall Architecture



Navigation2 provides the following tools:

- Tools for loading, serving, and storing maps (Map Server)
- Tools for localizing the robot on a map (AMCL)
- Tools for planning paths from point A to point B while avoiding obstacles (Nav2 Planner)
- Tools for controlling the robot while following a path (Nav2 Controller)
- Tools for converting sensor data into a costmap representation of the robot's world (Nav2 Costmap 2D)
- Tools for building complex robot behaviors using behavior trees (Nav2 Behavior Trees and BT Navigator)
- Tools for calculating recovery behaviors in the event of a failure (Nav2 Recoveries)
- Tools for following sequential waypoints (Nav2 Waypoint Follower)
- Tools and watchdogs for managing server lifecycles (Nav2 Lifecycle Manager)
- Plugins for enabling user-defined algorithms and behaviors (Nav2 Core)

Navigation 2 (Nav2) is the native navigation framework in ROS 2. Its purpose is to safely move mobile robots from point A to point B. Nav2 implements behaviors such as dynamic path planning, motor speed calculation, obstacle avoidance, and structure recovery.

Nav2 uses behavior trees (BTs) to call modular servers to perform an action. Actions can include path calculation, control efforts, recovery, or other navigation-related actions. These actions are independent nodes that communicate with the behavior tree (BT) through the action server.

2.2 Related Materials

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

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

Navigation2 Paper: https://arxiv.org/pdf/2003.00368.pdf

3. Running Examples

3.1 Pre-use Notes

This lesson uses the Raspberry Pi as an example.

For Raspberry Pi and Jetson-Nano boards, you need to open a terminal on the host computer and enter the command to enter the Docker container. Once inside the Docker container, enter the commands mentioned in this lesson in the terminal. For instructions on entering the Docker container from the host computer, refer to **[01. Robot Configuration and Operation Guide] -- [4.Enter Docker (For JETSON Nano and RPi 5)]**.

For RDKX5 and Orin boards, simply open a terminal and enter the commands mentioned in this lesson.

3.2 Single-Point Navigation

Note:

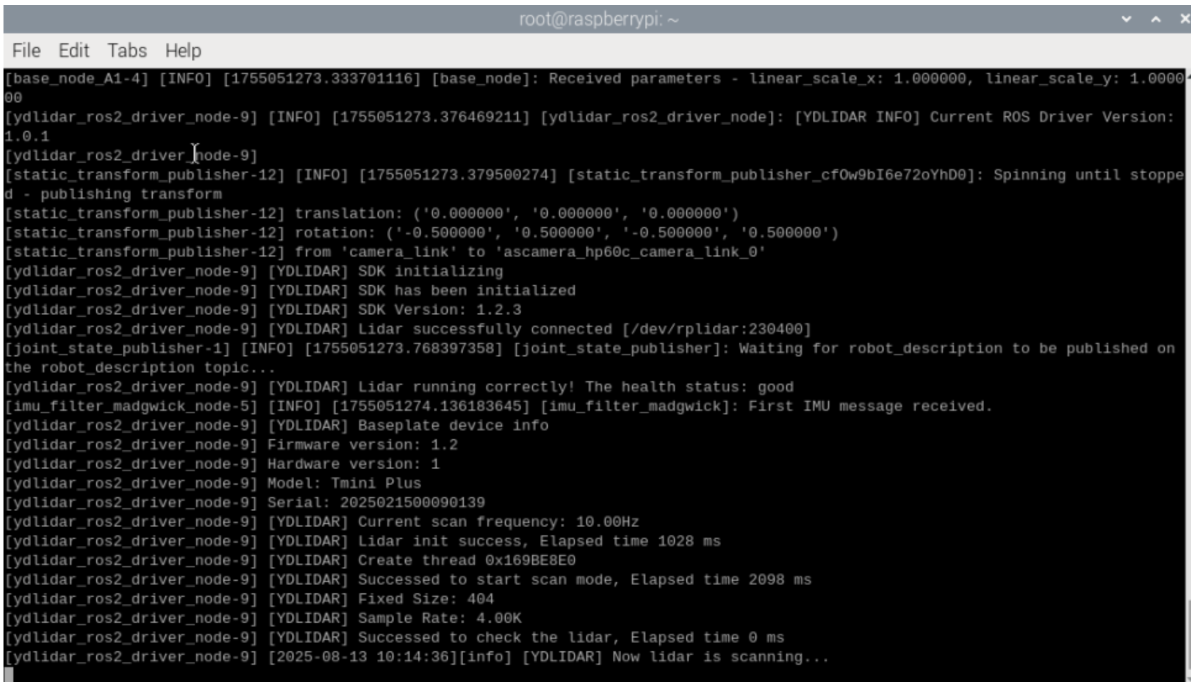
- **For Raspberry Pi and Jetson-Nano boards, you need to enter the Docker container first. For RDKX5 and Orin controllers, this is not necessary.**

  Enter the Docker container (see [20. Docker course] --- [4. Docker Startup Script] for steps).

  All the following commands must be executed within the same Docker container **(see [20. Docker course] --- [3. Docker Submission and Multi-Terminal Access] for steps).**
- This section requires at least one existing map. Refer to **[5.Gmapping-SLAM mapping], [6.Cartographer-SLAM mapping], [7.slam_toolbox mapping]** or any of the SLAM Mapping courses.

Robot car terminal launches the underlying sensor command:

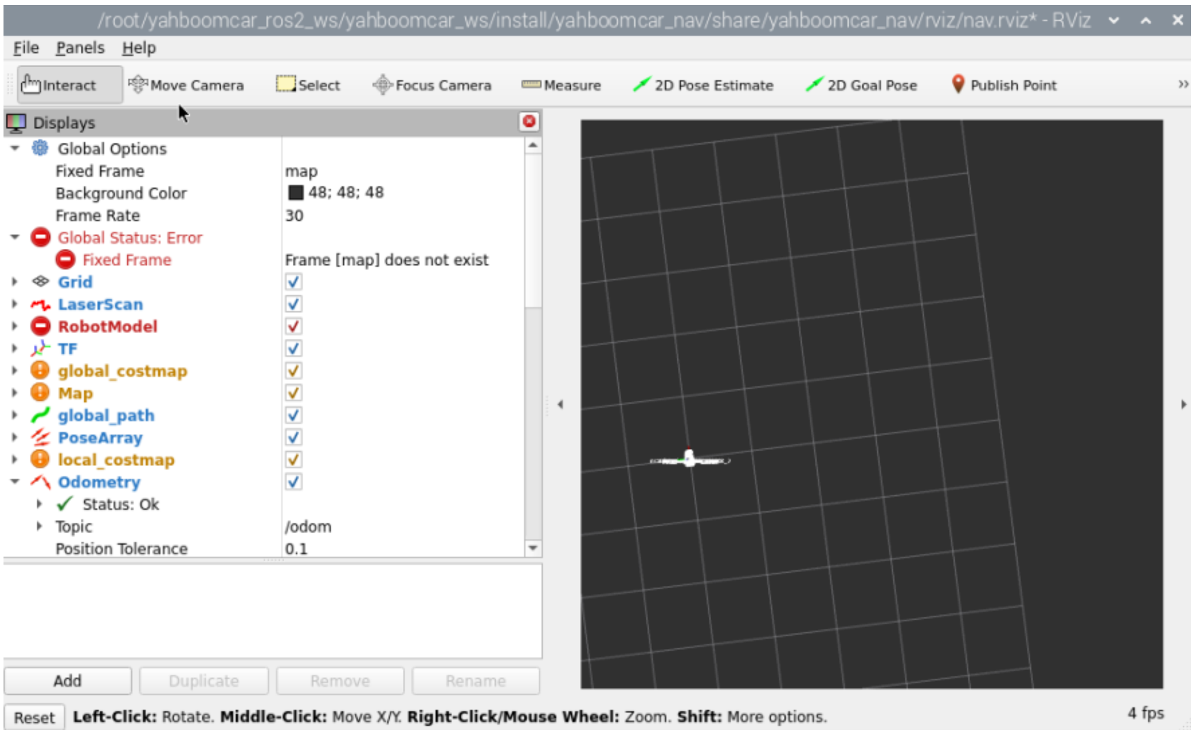
```
ros2 launch yahboomcar_nav laser_bringup_launch.py
```



```
[base_node_A1-4] [INFO] [1755051273.333701116] [base_node]: Received parameters - linear_scale_x: 1.000000, linear_scale_y: 1.0000
00
[ydlidar_ros2_driver_node-9] [INFO] [1755051273.376469211] [ydlidar_ros2_driver_node]: [YDLIDAR INFO] Current ROS Driver Version:
1.0.1
[ydlidar_ros2_driver_node-9]
[static_transform_publisher-12] [INFO] [1755051273.379500274] [static_transform_publisher_cfOw9bI6e72oYhD0]: Spinning until stoppe
d - publishing transform
[static_transform_publisher-12] translation: ('0.000000', '0.000000', '0.000000')
[static_transform_publisher-12] rotation: ('-0.500000', '0.500000', '-0.500000', '0.500000')
[static_transform_publisher-12] from 'camera_link' to 'ascamera_hp60c_camera_link_0'
[ydlidar_ros2_driver_node-9] [YDLIDAR] SDK initializing
[ydlidar_ros2_driver_node-9] [YDLIDAR] SDK has been initialized
[ydlidar_ros2_driver_node-9] [YDLIDAR] SDK Version: 1.2.3
[ydlidar_ros2_driver_node-9] [YDLIDAR] Lidar successfully connected [/dev/rplidar:230400]
[joint_state_publisher-1] [INFO] [1755051273.768397358] [joint_state_publisher]: Waiting for robot_description to be published on
the robot_description topic...
[ydlidar_ros2_driver_node-9] [YDLIDAR] Lidar running correctly! The health status: good
[imu_filter_madgwick_node-5] [INFO] [1755051274.136183645] [imu_filter_madgwick]: First IMU message received.
[ydlidar_ros2_driver_node-9] [YDLIDAR] Baseplate device info
[ydlidar_ros2_driver_node-9] Firmware version: 1.2
[ydlidar_ros2_driver_node-9] Hardware version: 1
[ydlidar_ros2_driver_node-9] Model: Tmini Plus
[ydlidar_ros2_driver_node-9] Serial: 2025021500090139
[ydlidar_ros2_driver_node-9] [YDLIDAR] Current scan frequency: 10.00Hz
[ydlidar_ros2_driver_node-9] [YDLIDAR] Lidar init success, Elapsed time 1028 ms
[ydlidar_ros2_driver_node-9] [YDLIDAR] Create thread 0x169BE8E0
[ydlidar_ros2_driver_node-9] [YDLIDAR] Successed to start scan mode, Elapsed time 2098 ms
[ydlidar_ros2_driver_node-9] [YDLIDAR] Fixed Size: 404
[ydlidar_ros2_driver_node-9] [YDLIDAR] Sample Rate: 4.00K
[ydlidar_ros2_driver_node-9] [YDLIDAR] Successed to check the lidar, Elapsed time 0 ms
[ydlidar_ros2_driver_node-9] [2025-08-13 10:14:36][info] [YDLIDAR] Now lidar is scanning...
```

Enter the command to start RViz visualization mapping (RViz visualization can be started on either the vehicle's infotainment system or a virtual machine; **choose one**, but do not start it on both the virtual machine and the vehicle's infotainment system simultaneously :). The image below shows the input on the vehicle's infotainment system.

```
ros2 launch yahboomcar_nav display_nav_launch.py
```

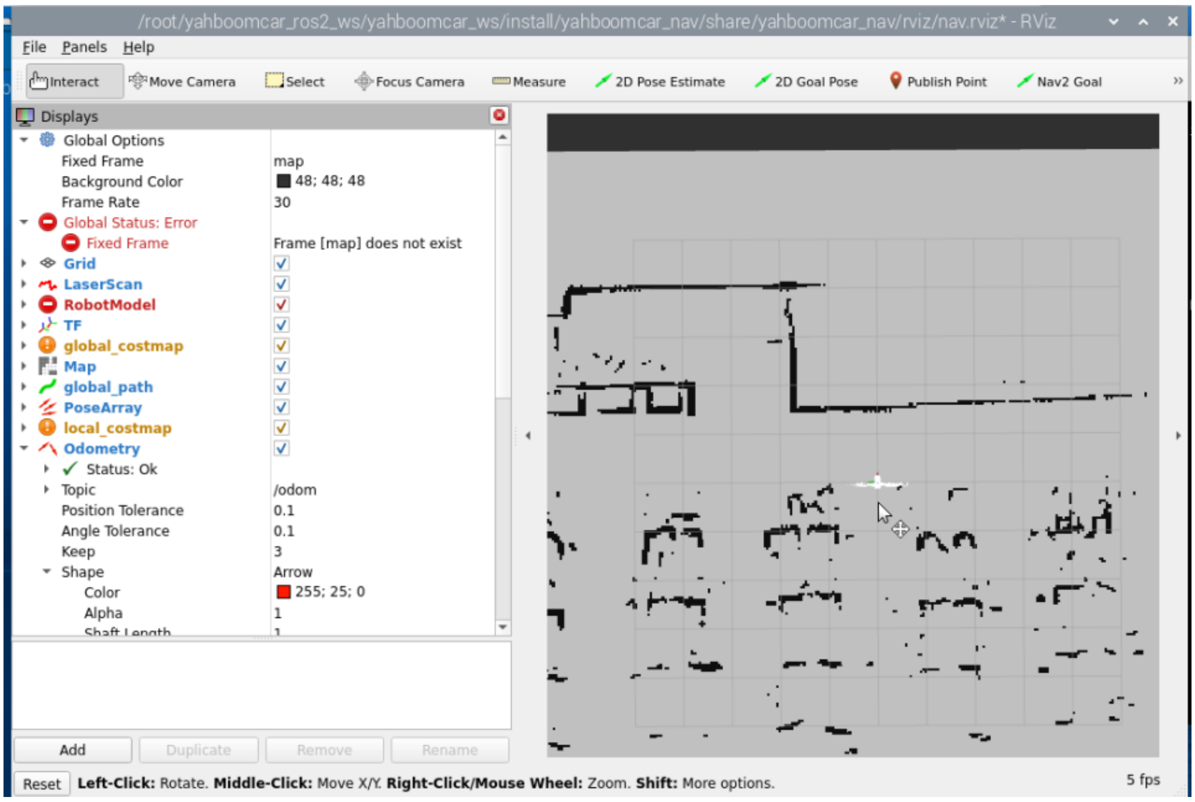


```
Global Options
  Fixed Frame           map
  Background Color      48; 48; 48
  Frame Rate            30
Global Status: Error
  Fixed Frame           Frame [map] does not exist
Grid                    ✓
LaserScan               ✓
RobotModel              ✓
TF                      ✓
global_costmap          ✓
Map                     ✓
global_path             ✓
PoseArray               ✓
local_costmap           ✓
Odometry                ✓
  Status: Ok
  Topic                 /odom
  Position Tolerance    0.1
```

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.

At this point, the map is not yet loaded because the navigation program has not started. Next, run the navigation node by entering **(you need to be in the same Docker terminal as above)** in the terminal.

```
# Choose one of two navigation algorithms
# teb
ros2 launch yahboomcar_nav navigation_teb_launch.py
# dwa
ros2 launch yahboomcar_nav navigation_dwa_launch.py
```

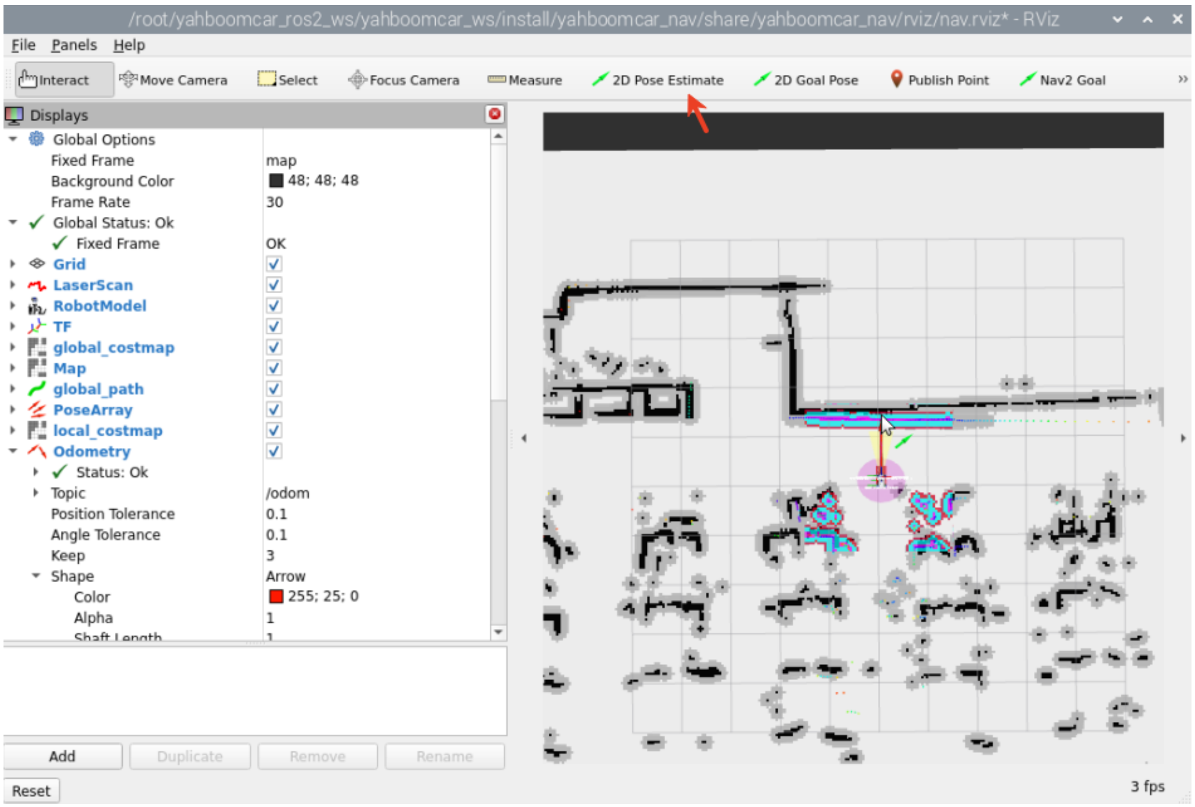


/root/yahboomcar_ros2_ws/yahboomcar_ws/install/yahboomcar_nav/share/yahboomcar_nav/rviz/nav.rviz* - RViz

Global Options	
Fixed Frame	map
Background Color	■ 48; 48; 48
Frame Rate	30
Global Status: Error	
Fixed Frame	Frame [map] does not exist
Grid	✓
LaserScan	✓
RobotModel	✓
TF	✓
global_costmap	✓
Map	✓
global_path	✓
PoseArray	✓
local_costmap	✓
Odometry	✓
Status: Ok	
Topic	/odom
Position Tolerance	0.1
Angle Tolerance	0.1
Keep	3
Shape	Arrow
Color	🟥 255; 25; 0
Alpha	1
Shaft Length	1

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:** Zoom. **Shift:** More options.

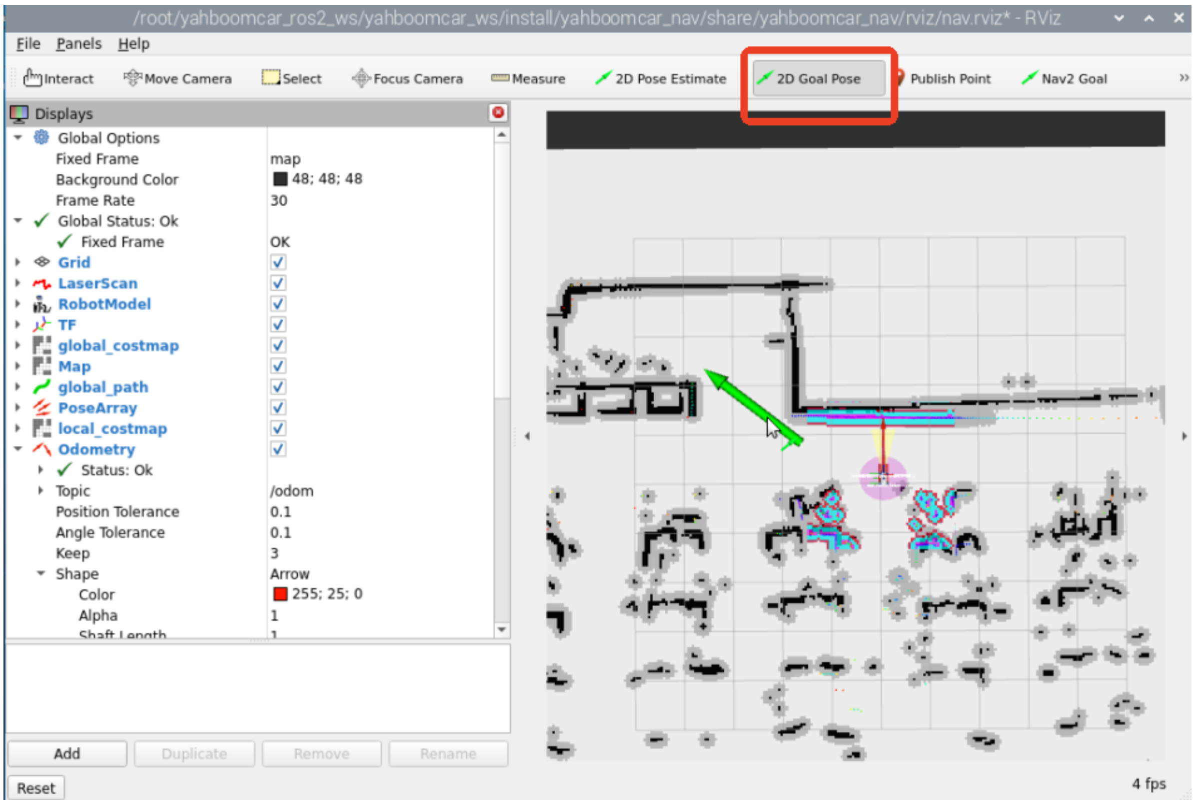
At this point, you can see the map has loaded. Then, click on 【2D Pose Estimate】 to set the initial pose for the robot. Based on the robot's position in the actual environment, use the mouse to click and drag in RViz to move the robot model to the position you set. As shown in the image below, if the radar-scanned area roughly overlaps with the actual obstacles, the pose is accurate. After pose initialization, the robot model and the inflated region will appear in the RViz interface.



/root/yahboomcar_ros2_ws/yahboomcar_ws/install/yahboomcar_nav/share/yahboomcar_nav/rviz/nav.rviz* - RViz

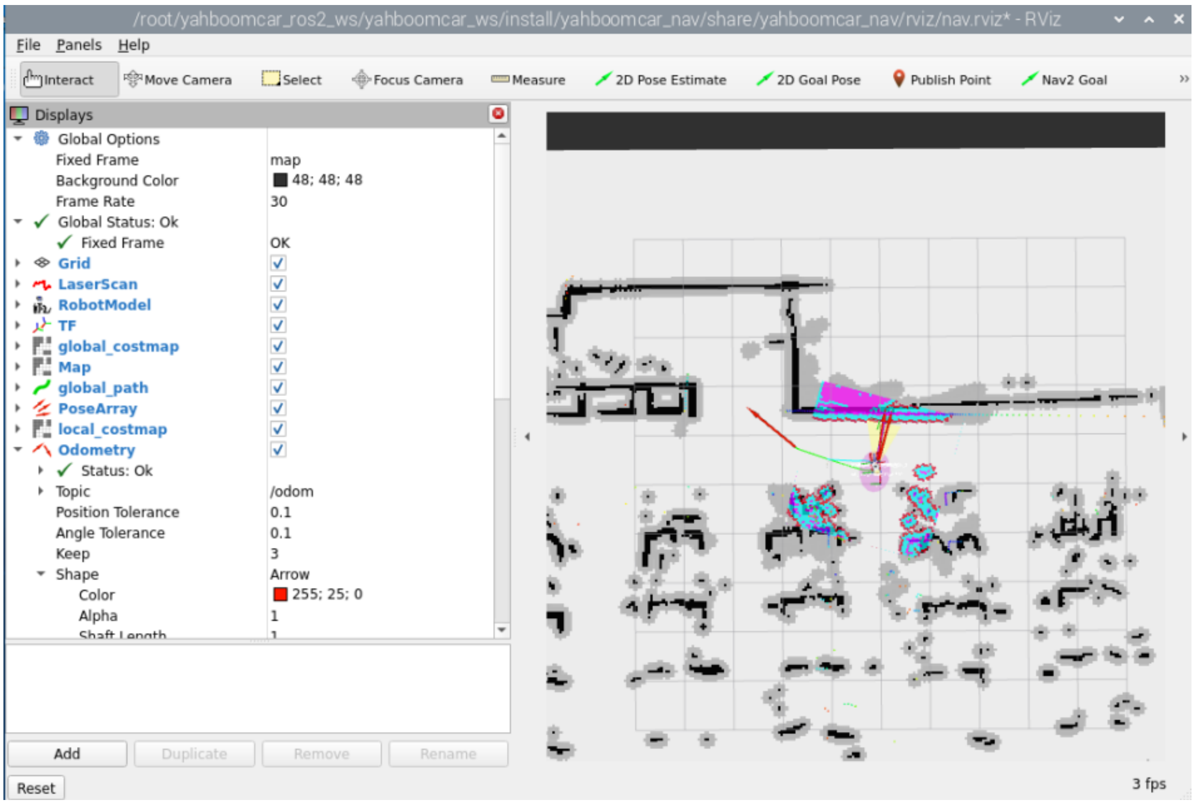
Global Options	
Fixed Frame	map
Background Color	■ 48; 48; 48
Frame Rate	30
Global Status: Ok	
Fixed Frame	OK
Grid	✓
LaserScan	✓
RobotModel	✓
TF	✓
global_costmap	✓
Map	✓
global_path	✓
PoseArray	✓
local_costmap	✓
Odometry	✓
Status: Ok	
Topic	/odom
Position Tolerance	0.1
Angle Tolerance	0.1
Keep	3
Shape	Arrow
Color	🟥 255; 25; 0
Alpha	1
Shaft Length	1

For single-point navigation, click the 【2D Goal Pose】 tool, then use the mouse to select a target point and orientation in RVIZ, then release.



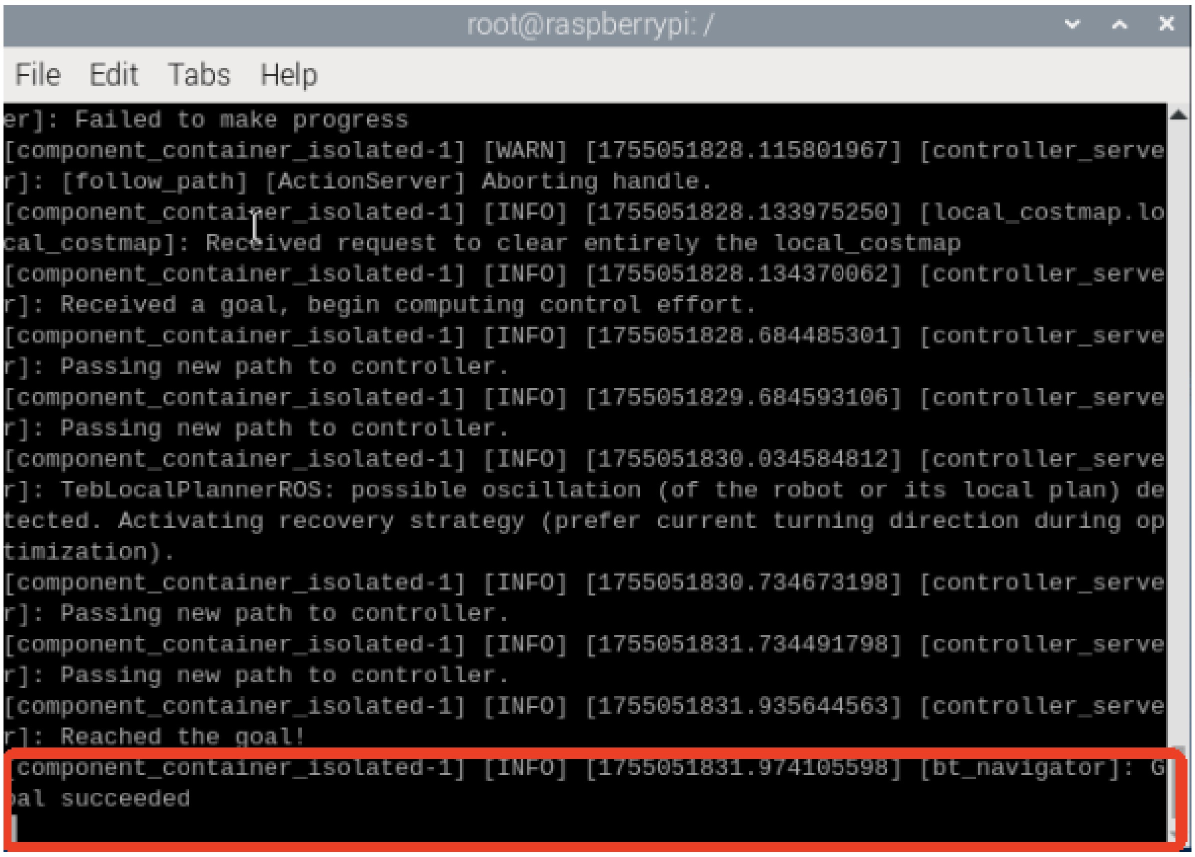
```
Global Options
    Fixed Frame            map
    Background Color       48; 48; 48
    Frame Rate             30
Global Status: Ok
    Fixed Frame            OK
Grid                       ✓
LaserScan                  ✓
RobotModel                 ✓
TF                         ✓
global_costmap             ✓
Map                        ✓
global_path                ✓
PoseArray                  ✓
local_costmap              ✓
Odometry                   ✓
    Status: Ok
    Topic                  /odom
    Position Tolerance     0.1
    Angle Tolerance        0.1
    Keep                   3
    Shape                  Arrow
        Color              255; 25; 0
        Alpha              1
        Shaft Length       1
```

The robot plans a path based on its surroundings and moves along it to the target point.



```
Global Options
    Fixed Frame            map
    Background Color       48; 48; 48
    Frame Rate             30
Global Status: Ok
    Fixed Frame            OK
Grid                       ✓
LaserScan                  ✓
RobotModel                 ✓
TF                         ✓
global_costmap             ✓
Map                        ✓
global_path                ✓
PoseArray                  ✓
local_costmap              ✓
Odometry                   ✓
    Status: Ok
    Topic                  /odom
    Position Tolerance     0.1
    Angle Tolerance        0.1
    Keep                   3
    Shape                  Arrow
        Color              255; 25; 0
        Alpha              1
        Shaft Length       1
```

After the robot successfully reaches the target point, the vehicle terminal will display "Goal succeeded," indicating successful navigation.



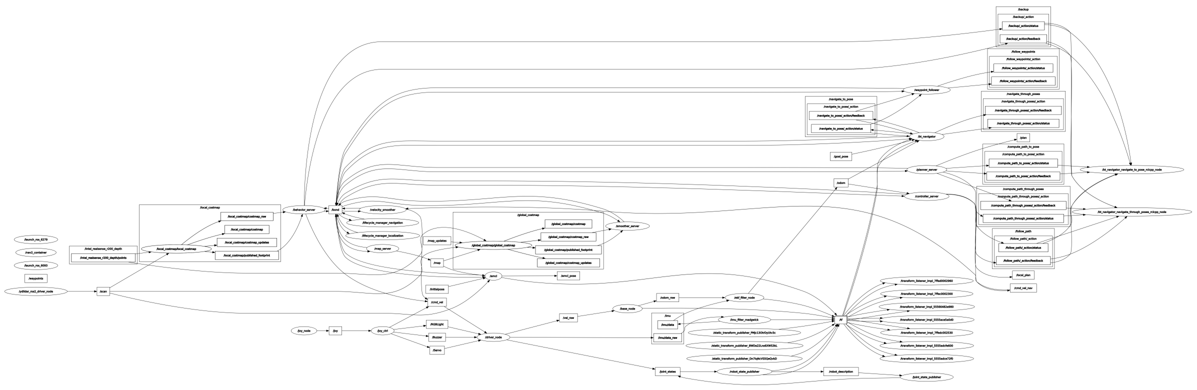
```
er]: Failed to make progress
[component_container_isolated-1] [WARN] [1755051828.115801967] [controller_serve
r]: [follow_path] [ActionServer] Aborting handle.
[component_container_isolated-1] [INFO] [1755051828.133975250] [local_costmap.lo
cal_costmap]: Received request to clear entirely the local_costmap
[component_container_isolated-1] [INFO] [1755051828.134370062] [controller_serve
r]: Received a goal, begin computing control effort.
[component_container_isolated-1] [INFO] [1755051828.684485301] [controller_serve
r]: Passing new path to controller.
[component_container_isolated-1] [INFO] [1755051829.684593106] [controller_serve
r]: Passing new path to controller.
[component_container_isolated-1] [INFO] [1755051830.034584812] [controller_serve
r]: TebLocalPlannerROS: possible oscillation (of the robot or its local plan) de
tected. Activating recovery strategy (prefer current turning direction during op
timization).
[component_container_isolated-1] [INFO] [1755051830.734673198] [controller_serve
r]: Passing new path to controller.
[component_container_isolated-1] [INFO] [1755051831.734491798] [controller_serve
r]: Passing new path to controller.
[component_container_isolated-1] [INFO] [1755051831.935644563] [controller_serve
r]: Reached the goal!
[component_container_isolated-1] [INFO] [1755051831.974105598] [bt_navigator]: G
oal succeeded
```

4. View the Node Communication Graph

Enter the terminal:

```
ros2 run rqt_graph rqt_graph
```

If the graph does not display initially, select [Nodes/Topics (all)] and click the refresh button in the upper left corner. The original graph is too large; you can view it in the current lesson folder.

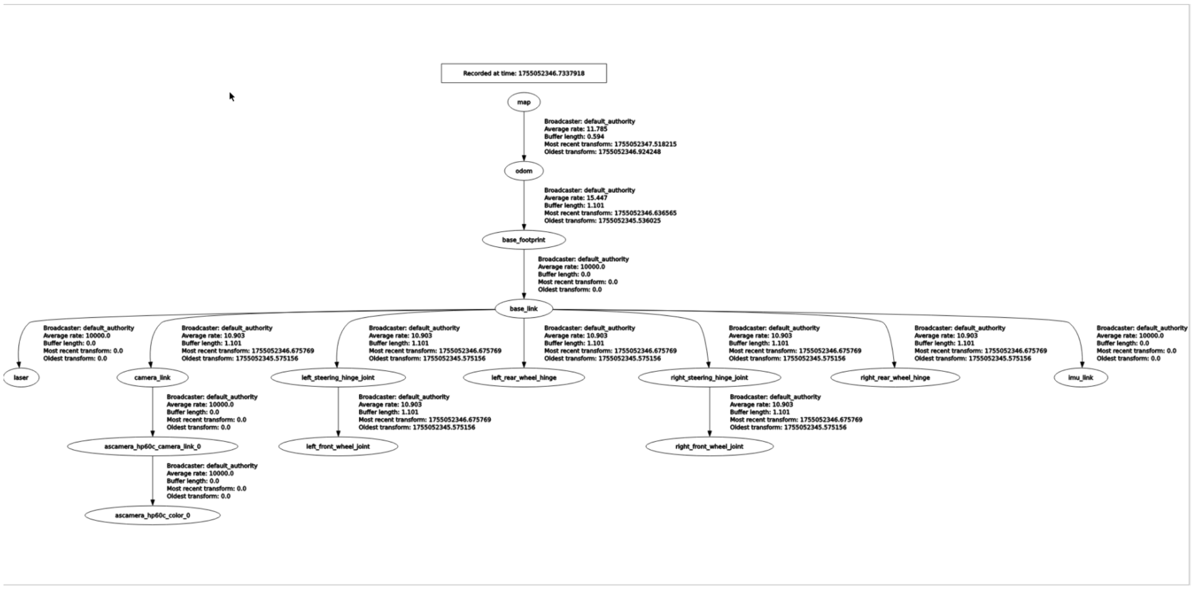


5. View the TF Tree

Enter the terminal:

```
ros2 run rqt_tf_tree rqt_tf_tree
```

If the page doesn't display initially, click the refresh icon in the upper left corner to refresh the page. The original image is too large; you can view it in this lesson's folder.


Recorded at time: 1755052346.7337918
map
Broadcaster: default_authority
Average rate: 11.785
Buffer length: 0.594
Most recent transform: 1755052347.518215
Oldest transform: 1755052346.924248
odom
Broadcaster: default_authority
Average rate: 15.447
Buffer length: 1.101
Most recent transform: 1755052346.636565
Oldest transform: 1755052345.536025
base_footprint
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
base_link
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
laser
Broadcaster: default_authority
Average rate: 10.903
Buffer length: 1.101
Most recent transform: 1755052346.675769
Oldest transform: 1755052345.575156
camera_link
Broadcaster: default_authority
Average rate: 10.903
Buffer length: 1.101
Most recent transform: 1755052346.675769
Oldest transform: 1755052345.575156
left_steering_hinge_joint
Broadcaster: default_authority
Average rate: 10.903
Buffer length: 1.101
Most recent transform: 1755052346.675769
Oldest transform: 1755052345.575156
left_rear_wheel_hinge
Broadcaster: default_authority
Average rate: 10.903
Buffer length: 1.101
Most recent transform: 1755052346.675769
Oldest transform: 1755052345.575156
right_steering_hinge_joint
Broadcaster: default_authority
Average rate: 10.903
Buffer length: 1.101
Most recent transform: 1755052346.675769
Oldest transform: 1755052345.575156
right_rear_wheel_hinge
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
imu_link
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
ascamera_hp60c_camera_link_0
Broadcaster: default_authority
Average rate: 10.903
Buffer length: 1.101
Most recent transform: 1755052346.675769
Oldest transform: 1755052345.575156
left_front_wheel_joint
Broadcaster: default_authority
Average rate: 10.903
Buffer length: 1.101
Most recent transform: 1755052346.675769
Oldest transform: 1755052345.575156
right_front_wheel_joint
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
ascamera_hp60c_color_0

6. Navigation Explanation

6.1 Core Steps of the Navigation Process

6.1.1 Phase 1: System Initialization and Map Loading

- **Map Acquisition**
- Load a pre-built raster map from `map_server`
- The map contains static obstacle information and serves as the basic environment model for navigation.

Cost Map Initialization

- **Global Costmap**: Based on the static map, used for global path planning.
- **Local Costmap**: Integrates the real-time `scan` topic of the lidar for dynamic obstacle avoidance.

6.1.2 Phase 2: Sensor Data Processing and Environmental Perception

Sensor Data Access

- LiDAR (/scan), IMU (/imu/data), and odometry (/odom) data are input via ROS topics.

Cost Map Update

- The local cost map updates dynamic obstacle information in real time. Each grid cell calculates an occupancy probability and cost based on sensor data (e.g., the closer to the obstacle, the higher the cost).
- Inflation: Expands the edges of obstacles to prevent the robot from getting too close.

6.1.3 Phase 3: Path Planning (Global and Local Planning)

Global Path Planning

- Input: Start point (current robot pose), end point (goal pose), and global cost map.
- The Dijkstra planner algorithm generates an optimal path (a series of discrete coordinate points) from the start point to the end point.

Local Path Planning and Tracking

The local planner, DWBLocalPlanner, generates short-term control commands that the robot can execute based on the global path and the local costmap.

6.1.4 Phase 4: Control Execution and Behavior Decision-Making

Controller Output

- The controller converts the local path into the robot's linear velocity (`linear.x`) and angular velocity (`angular.z`), which are published via the `/cmd_vel` topic.
- Velocity Smoothing: This prevents abrupt changes in robot motion and improves stability.

Behavior Tree Decision Process

The behavior tree defines the priority and state transition logic for navigation tasks. A typical process is as follows:

1. **Checking if the target is reachable**: If global planning fails, trigger a "recovery behavior" (e.g., replanning).
2. **Performing Local Obstacle Avoidance**: When an obstacle is detected in the local costmap, temporarily deviate from the global path.
3. **Reaching the Target**: When the error between the robot's pose and the target pose is less than a threshold, the task is completed.

Recovery Behavior Mechanism

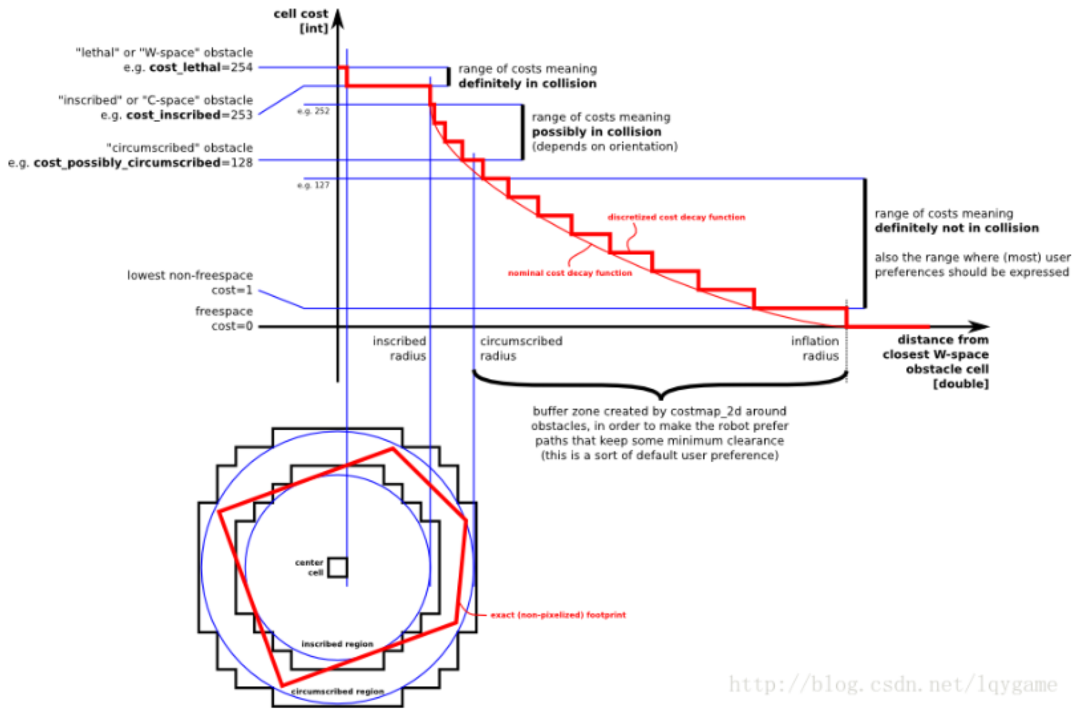
When navigation encounters an anomaly (such as a completely blocked path), a pre-set recovery strategy (such as rotating to search for a new path or backing off to replan) is triggered.

6.2 Key Technical Principles

6.2.1 Cost Map

- **Probabilistic Grid Representation**: Each grid stores an occupancy probability (0-1), which is updated based on sensor data (such as ray detection from a LiDAR).
- **Multi-layer Cost Overlay**: Static cost (for obstacles in the map) + dynamic cost (for obstacles detected by real-time sensors) + inflation cost (for safety distance).

A cost map is a two-dimensional or three-dimensional map created and updated by the robot using sensor information. The figure below provides a brief overview.



cell cost [int]
"lethal" or "W-space" obstacle e.g. cost_lethal=254
"inscribed" or "C-space" obstacle e.g. cost_inscribed=253
e.g. 252
"circumscribed" obstacle e.g. cost_possibly_circumscribed=128
e.g. 127
lowest non-freespace cost=1
freespace cost=0
range of costs meaning definitely in collision
range of costs meaning possibly in collision (depends on orientation)
range of costs meaning definitely not in collision
also the range where (most) user preferences should be expressed
discretized cost decay function
nominal cost decay function
inscribed radius
circumscribed radius
inflation radius
distance from closest W-space obstacle cell [double]
buffer zone created by costmap_2d around obstacles, in order to make the robot prefer paths that keep some minimum clearance (this is a sort of default user preference)
center cell
exact (non-pixelized) footprint
inscribed region
circumscribed region
http://blog.csdn.net/lqygame

In the image above, the red area represents obstacles in the costmap, the blue area represents obstacles expanded by the radius of the robot's inscribed circle, and the red polygon represents the footprint (the vertical projection of the robot's outline). To avoid collisions, the footprint should not intersect the red area, and the robot's center should not intersect the blue area.

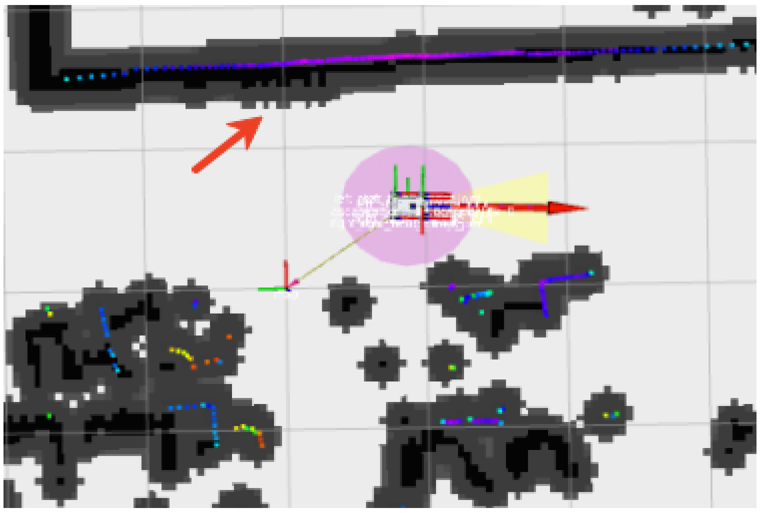
The ROS costmap uses a grid format, with each cell value (cell cost) ranging from 0 to 255. It is divided into three states: occupied (obstacles present), free area (no obstacles), and unknown area.

The specific states and values are shown in the following figure:

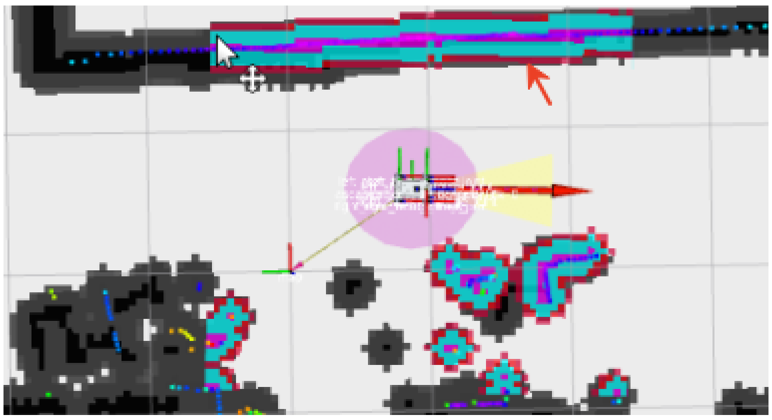
The image above can be divided into five sections, with the red polygon representing the robot's outline:

- Lethal: If the robot's center coincides with the center of this grid, the robot will inevitably collide with the obstacle.
- Inscribed: If the grid's circumcircle is inscribed within the robot's outline, the robot will inevitably collide with the obstacle. - Possibly circumscribed: The mesh's circumscribed circle circumscribes the robot's outline. In this case, the robot is essentially near an obstacle, so there's no collision.
- Freespace: Space without obstacles.
- Unknown: Space without known obstacles.

Global costmap:



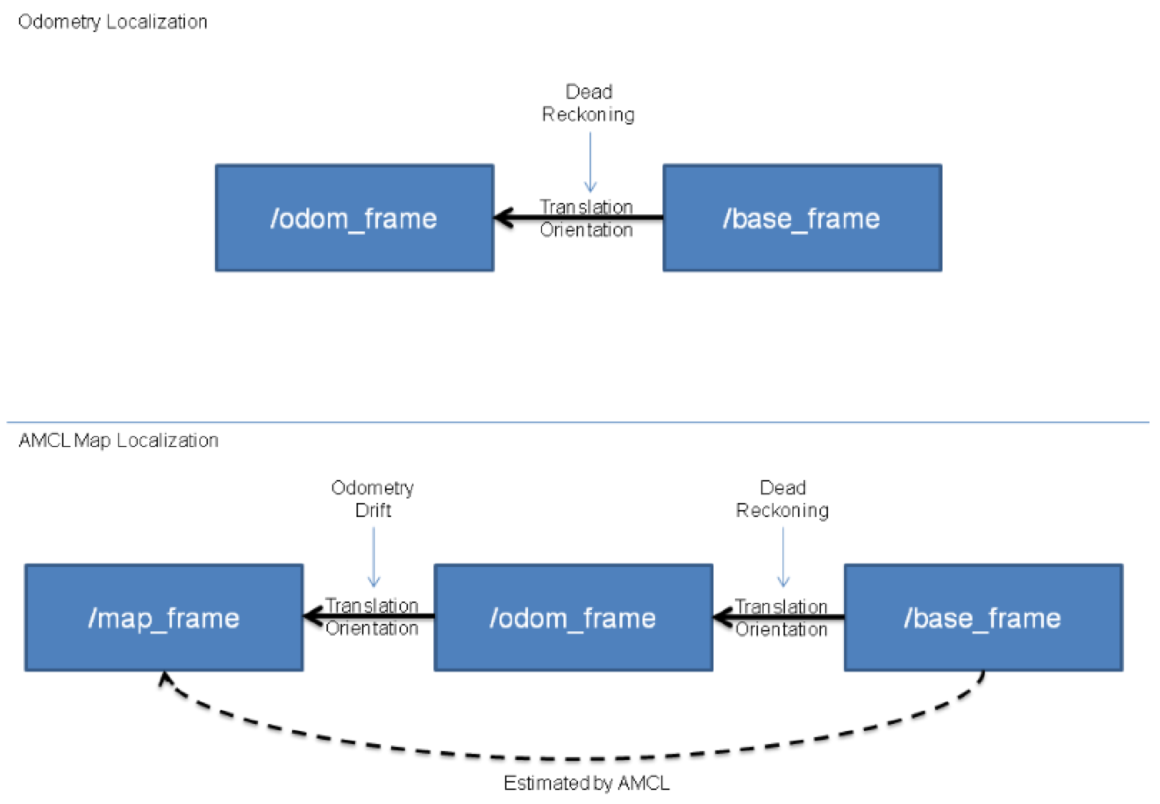
Local costmap:



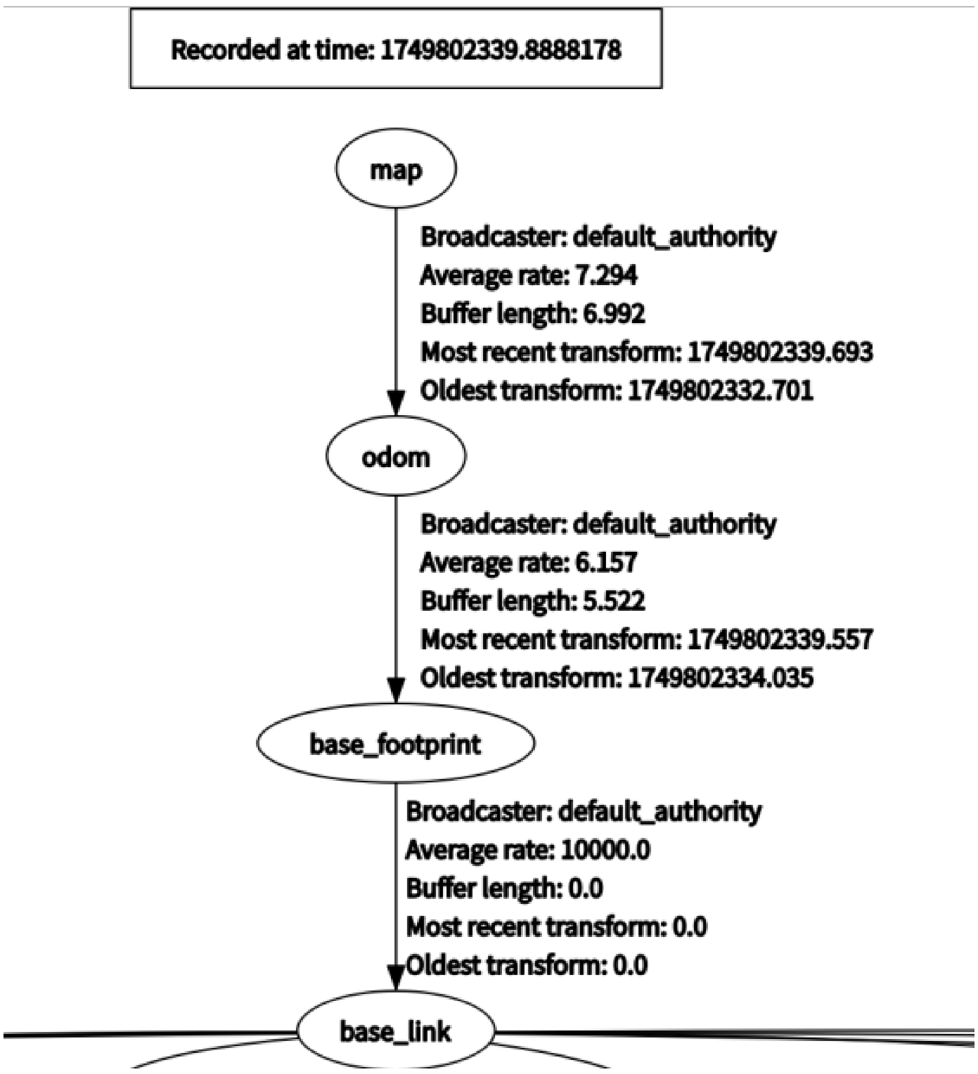
6.2.2 AMCL Localization Algorithm

AMCL (Adaptive Monte Carlo Localization) is a probabilistic localization system for 2D mobile robots. It implements an adaptive (or KLD sampling) Monte Carlo localization method that uses a particle filter to estimate the robot's position based on a given map.

As shown in the figure below, if the odometry is error-free, in a perfect world, we can directly use the odometry information (top half) to infer the robot's (base_frame) position relative to the odometry coordinate system. However, in reality, odometry drift and non-negligible cumulative errors exist. Therefore, AMCL uses the method in the bottom half. This method first uses the odometry information to preliminarily locate the base_frame. Then, using the measurement model, we determine the base_frame's position relative to the map_frame (global map coordinate system), thus determining the robot's position within the map. (Note that while the base-to-map transformation is estimated here, the final published transformation is the map-to-odom transformation, which can be understood as odometry drift.)



In the TF tree during robot navigation, the coordinate transformation between the map (map coordinate system) and the odom (odometry coordinate system) is published by the AMCL localization algorithm, while the coordinate transformation between the odom (odometry coordinate system) and the besel_footprint (robot chassis center projection coordinate system) is published by the ekf_filter_node node in the robot_localization package. Its function is to publish odometry information after fusion of the raw odometry from the IMU sensor and encoder.

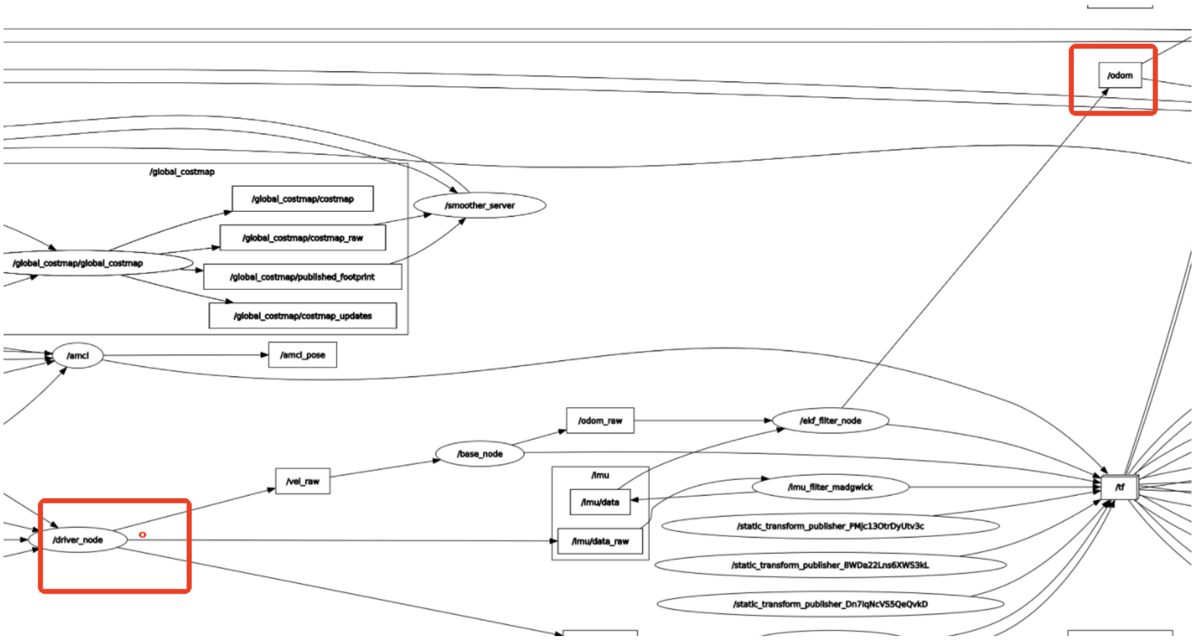

Recorded at time: 1749802339.8888178
map
Broadcaster: default_authority
Average rate: 7.294
Buffer length: 6.992
Most recent transform: 1749802339.693
Oldest transform: 1749802332.701
odom
Broadcaster: default_authority
Average rate: 6.157
Buffer length: 5.522
Most recent transform: 1749802339.557
Oldest transform: 1749802334.035
base_footprint
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
base_link

In the node communication diagram above, we can see the communication data flow for the ekf_filter_node.

The /driver_node node is the robot chassis node, publishing **/imu/data_raw** (raw IMU sensor data) and **/odom_raw** (raw encoder odometry data).

The imu_filter node subscribes to **/ imu/data_raw** (raw IMU sensor data). The imu_filter node filters the raw IMU sensor data and publishes it to the /imu/data topic.

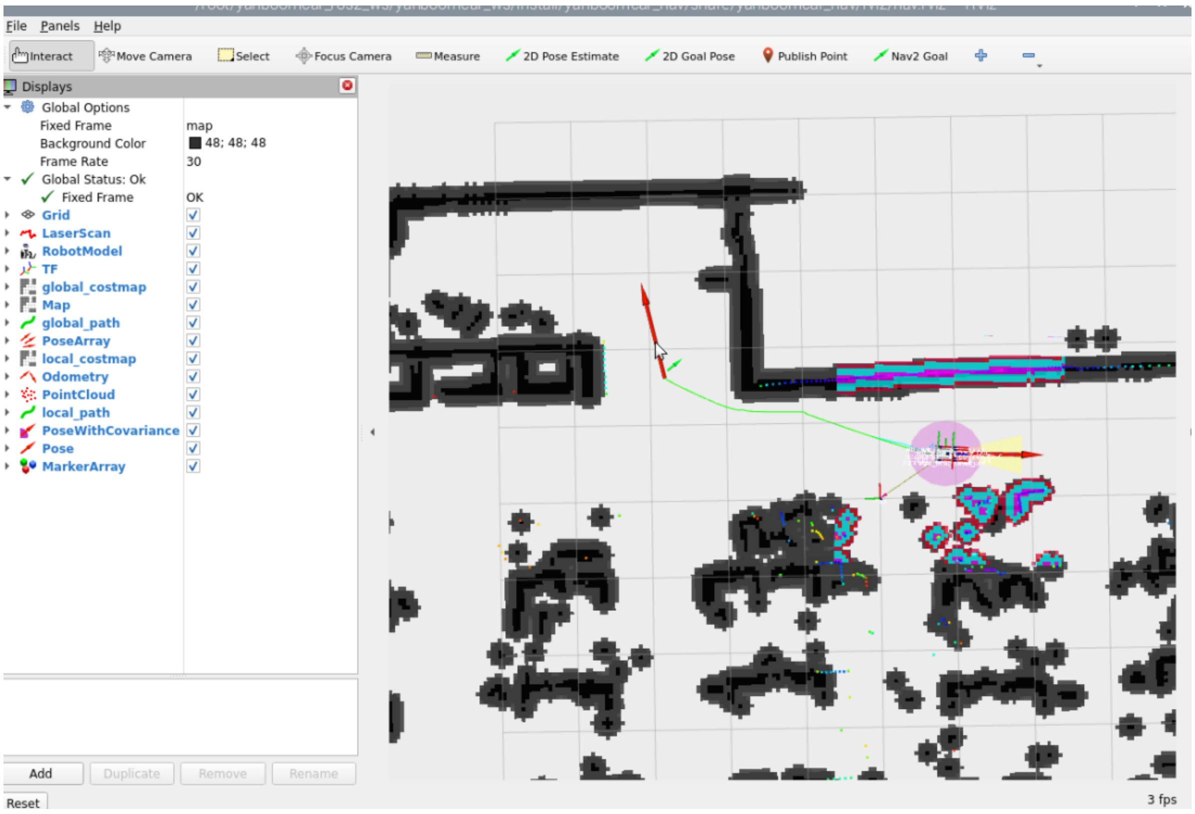
The **ekf_filter_node** node subscribes to the /**odom** and **/imu/data** topics, fuses the multi-sensor data, and publishes it to the **/odom** topic.


/odom
/global_costmap
/global_costmap/costmap
/smoother_server
/global_costmap/costmap_raw
/global_costmap/global_costmap
/global_costmap/published_footprint
/global_costmap/costmap_updates
/amcl
/amcl_pose
/odom_raw
/ekf_filter_node
/base_node
/vel_raw
/imu
/imu/data
/imu_filter_madgwick
/tf
/driver_node
/imu/data_raw
/static_transform_publisher_PMjc13OtrDyUtv3c
/static_transform_publisher_8WDa22Lns6XWS3kL
/static_transform_publisher_Dn7IqNcVS5QeQvkD

6.3 Path Planning

6.3.1 Global Path Planning

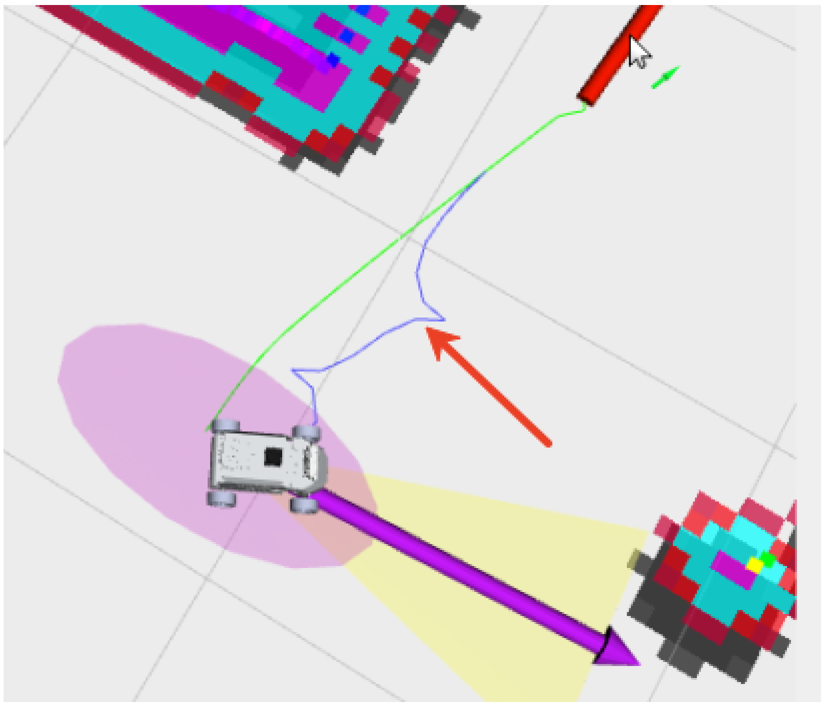
Global path planning calculates an optimal or feasible global path from a start point to a destination within a known environment map, disregarding real-time obstacles. The path from the start point to the destination is calculated with the global optimum as the goal. As shown in the figure below, the path connected by arrows is the global path.



Global Options
Fixed Frame map
Background Color 48; 48; 48
Frame Rate 30
Global Status: Ok
Fixed Frame OK
Grid ✓
LaserScan ✓
RobotModel ✓
TF ✓
global_costmap ✓
Map ✓
global_path ✓
PoseArray ✓
local_costmap ✓
Odometry ✓
PointCloud ✓
local_path ✓
PoseWithCovariance ✓
Pose ✓
MarkerArray ✓

6.3.2 Local Path Planning

Local path planning dynamically adjusts the robot's path based on real-time LiDAR sensor data during movement to avoid obstacles (obstacles are displayed in the costmap). As shown in the figure below, the blue path is the local path.



6.4 Navigation Parameter Configuration

Jetson Nano and Raspberry Pi require Docker first

Configuration file path:

```
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/params/teb_nav_params.yaml
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/params/dwa_nav_params.yaml
```

The default configuration parameters for the TEB algorithm navigation function are as follows:

```
amcl:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time
    alpha1: 0.1          # 里程计旋转误差模型参数 Odometry rotation error model
parameters
    alpha2: 0.1          # 里程计平移误差模型参数 Odometry translation error model
parameters
    alpha3: 0.3          # 里程计平移误差模型参数 Odometry translation error model
parameters
    alpha4: 0.3          # 里程计旋转误差模型参数 Odometry rotation error model
parameters
    alpha5: 0.1          # 里程计旋转误差模型参数 Odometry rotation error model
parameters
    base_frame_id: "base_footprint"  # 机器人基座坐标系 Robot base coordinate
system
    beam_skip_distance: 0.5          # 光束跳过距离阈值 Beam skip distance
threshold
    beam_skip_error_threshold: 0.9   # 光束跳过误差阈值 Beam skip error threshold
    beam_skip_threshold: 0.3         # 光束跳过阈值 Beam skip threshold
    do_beamskip: false               # 是否启用光束跳过 Enable beam skipping
    global_frame_id: "map"            # 全局坐标系 Global coordinate system
    lambda_short: 0.1                # 激光模型短距离权重 Laser model short distance
weight
    laser_likelihood_max_dist: 2.0   # 激光似然场最大距离 Maximum distance of the
laser likelihood field
```

```
    laser_max_range: 100.0          # 激光最大有效距离 Maximum effective laser
range
    laser_min_range: -1.0           # 激光最小有效距离 Minimum effective laser
range
    laser_model_type: "likelihood_field"  # 激光模型类型（似然场） Laser model type
(likelihood field)
    max_beams: 50                   # 每次处理的最大激光束数 Maximum number of
laser beams processed per shot
    max_particles: 4000             # 最大粒子数 Maximum number of particles
    min_particles: 1000             # 最小粒子数 Minimum number of particles
    odom_frame_id: "odom"           # 里程计坐标系 Odometry frame
    pf_err: 0.05                    # 粒子滤波器误差阈值 Particle filter error
threshold
    pf_z: 0.99                       # 粒子滤波器权重指数 Particle filter weight
exponent
    recovery_alpha_fast: 0.0         # 快速恢复参数 Fast recovery parameter
    recovery_alpha_slow: 0.0         # 慢速恢复参数 Slow recovery parameter
    resample_interval: 1            # 重采样间隔 Resampling interval
    robot_model_type: "nav2_amcl::DifferentialMotionModel"  # 机器人运动模型 Robot
motion model
    save_pose_rate: 0.5             # 保存位姿的速率 Save pose rate
    sigma_hit: 0.2                  # 激光模型标准差 Laser model standard
deviation
    tf_broadcast: true              # 是否广播TF变换 Broadcast TF transforms
    transform_tolerance: 1.0         # TF变换容差 TF transform tolerance
    update_min_a: 0.15              # 最小角度变化触发更新 Minimum angle change to
trigger an update
    update_min_d: 0.15              # 最小距离变化触发更新 Minimum distance change
to trigger an update
    z_hit: 0.5                      # 激光模型命中权重 Laser model hit weight
    z_max: 0.05                     # 激光模型最大距离权重 Laser model maximum
distance weight
    z_rand: 0.5                     # 激光模型随机权重 Laser model random weight
    z_short: 0.05                   # 激光模型短距离权重 Laser model short distance
weight
    scan_topic: scan                # 激光扫描话题 Laser scan topic

amcl_map_client:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Whether to use simulation time

amcl_rclcpp_node:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Whether to use simulation time

bt_navigator:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Whether to use simulation time
    global_frame: map    # 全局坐标系 Global coordinate system
    robot_base_frame: base_footprint  # 机器人基座坐标系 Robot base coordinate
system
    odom_topic: /odom   # 里程计话题 Odometry topic
    plugin_lib_names:   # 行为树插件库列表 Behavior tree plugin library list
      - nav2_compute_path_to_pose_action_bt_node
      - nav2_compute_path_through_poses_action_bt_node
      # ...（其他行为树节点插件 other behavior tree node plugins）

bt_navigator_rclcpp_node:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Whether to use simulation time

controller_server:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Whether to use simulation time
    controller_frequency: 30.0  # 控制器频率 Controller frequency
    min_x_velocity_threshold: 0.001  # X方向最小速度阈值 Minimum velocity threshold
in the X direction
    min_y_velocity_threshold: 0.0    # Y方向最小速度阈值 Minimum velocity threshold
in the Y direction
    min_theta_velocity_threshold: 0.001  # 旋转最小速度阈值 Minimum rotation
velocity threshold
    progress_checker_plugin: "progress_checker"  # 进度检查器插件 Progress checker
plugin
    goal_checker_plugin: "goal_checker"          # 目标检查器插件 Goal checker
plugin
    controller_plugins: ["FollowPath"]           # 控制器插件列表 List of
controller plugins
    # 进度检查器参数 Progress checker parameters
    progress_checker:
      plugin: "nav2_controller::SimpleProgressChecker"  # 简单进度检查器 Simple
progress checker
      required_movement_radius: 0.3  # 要求移动半径 Required movement radius
      movement_time_allowance: 3.5   # 移动时间容限 Movement time allowance
# Goal checker parameters
    # 目标检查器参数
    goal_checker:
      plugin: "nav2_controller::SimpleGoalChecker"  # 简单目标检查器 Simple Goal
Checker
      xy_goal_tolerance: 0.25  # XY位置容差 XY position tolerance
      yaw_goal_tolerance: 0.25  # 航向角容差 Heading angle tolerance
      stateful: True           # 是否保持状态 Maintain state
    # TEB局部规划器参数 TEB local planner parameters
    FollowPath:
      plugin: "teb_local_planner::TebLocalPlannerROS"  # TEB局部规划器 TEB local
planner
      footprint_model.type: circular  # 机器人足迹模型（圆形） Robot footprint model
(circular)
      footprint_model.radius: 0.1     # 机器人半径 Robot radius
      min_obstacle_dist: 0.1          # 最小障碍物距离 Minimum obstacle distance
      inflation_dist: 0.1             # 膨胀距离 Inflation distance
      costmap_converter_plugin: "costmap_converter::CostmapToPolygonsDBSMCCH"  #
代价地图转换插件 Costmap conversion plugin
      costmap_converter_spin_thread: True  # 转换器是否使用独立线程 Whether to use a
separate thread for the converter
      costmap_converter_rate: 20      # 转换器频率 Converter rate
      enable_homotopy_class_planning: True  # 启用同伦类规划 Enable homotopy class
planning
      enable_multithreading: True     # 启用多线程 Enable multithreading
      optimization_verbose: False    # 优化过程是否详细输出 Verbose output for the
optimization process
      teb_autoresize: True           # 自动调整轨迹点数量 Automatically adjust the
number of trajectory points
      min_samples: 3                 # 最小轨迹点数 Minimum number of trajectory
points
      max_samples: 20                 # 最大轨迹点数 Maximum number of trajectory
points
```

```
      max_global_plan_lookahead_dist: 1.0  # 全局规划前瞻距离 Global plan lookahead distance
      visualize_hc_graph: True        # 是否可视化同伦图 Visualize the homotopy graph
      max_vel_x: 0.26                 # 最大线速度Maximum linear velocity
      min_vel_x: -0.26                # 最小线速度（后退） Minimum linear velocity (reverse)
      max_vel_theta: 3.0              # 最大角速度 Maximum angular velocity
      min_vel_theta: -3.0             # 最小角速度 Minimum angular velocity
      acc_lim_x: 2.5                  # X方向加速度限制 X-axis acceleration limit
      acc_lim_theta: 3.2              # 旋转加速度限制 Rotational acceleration limit

costmap_converter:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time

controller_server_rclcpp_node:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time

local_costmap:
  local_costmap:
    ros__parameters:
      use_sim_time: False  # 是否使用仿真时间 Use simulation time
      update_frequency: 5.0  # 更新频率 Update frequency
      publish_frequency: 2.0  # 发布频率 Publish frequency
      global_frame: odom      # 全局坐标系 Global coordinate system
      robot_base_frame: base_footprint  # 机器人基座坐标系 Robot base coordinate system
      rolling_window: true     # 是否使用滚动窗口 Use rolling window
      width: 3                 # 代价地图宽度（米） Costmap width (meters)
      height: 3                # 代价地图高度（米） Costmap height (meters)
      resolution: 0.05          # 分辨率（米/像素） Resolution (meters/pixel)
      robot_radius: 0.1         # 机器人半径 Robot radius
      plugins: ["obstacle_layer", "voxel_layer", "inflation_layer"]  # 图层插件 Layer plugin
      inflation_layer:          # 膨胀层参数 Inflation layer parameters
        plugin: "nav2_costmap_2d::InflationLayer"  # 膨胀层插件 Inflation layer plugin
        cost_scaling_factor: 5.0  # 代价缩放因子 Cost scaling factor
        inflation_radius: 0.11    # 膨胀半径 Inflation radius
      obstacle_layer:           # 障碍物层参数 Obstacle layer parameters
        plugin: "nav2_costmap_2d::ObstacleLayer"  # 障碍物层插件 Obstacle layer plugin
        enabled: True           # 是否启用 Enable or disable
        observation_sources: scan  # 观测源（激光） Observation source (laser)
        scan:                   # 激光参数 Laser parameters
          topic: /scan           # 激光话题 Laser topic
          max_obstacle_height: 2.0  # 最大障碍物高度 Maximum obstacle height
          obstacle_min_range: 0.1    # 最小障碍物距离 Minimum obstacle distance
          clearing: True         # 是否用于清除空间 Clearing space
          marking: True          # 是否用于标记障碍 Whether to mark obstacles
          data_type: "LaserScan" # 数据类型 Data type
      voxel_layer:              # 体素层参数 Voxel layer parameters
        plugin: "nav2_costmap_2d::VoxelLayer"  # 体素层插件 Voxel layer plugin
        enabled: True           # 是否启用 whether is it enable
        publish_voxel_map: True # 是否发布体素地图 Whether to publish voxel map
        origin_z: 0.0           # Z轴原点 Z-axis origin
        z_resolution: 0.2       # Z轴分辨率 Z-axis resolution
        z_voxels: 10            # Z轴体素数 Z-axis voxel number
        max_obstacle_height: 2.0  # 最大障碍物高度 Maximum obstacle height
        mark_threshold: 0       # 标记阈值 Marking threshold
        observation_sources: pointcloud  # 观测源（点云） Observation source (point
cloud)
        pointcloud:              # 点云参数 Point cloud parameters
          topic: /intel_realsense_r200_depth/points  # 点云话题 Point cloud topic
          max_obstacle_height: 2.0  # 最大障碍物高度 Maximum obstacle height
          clearing: True          # 是否用于清除空间 Clear space
          marking: True           # 是否用于标记障碍 Whether to use it to mark
obstacles
          data_type: "PointCloud2"  # 数据类型 Data type
      static_layer:             # 静态层参数 Static layer parameters
        map_subscribe_transient_local: True  # 使用瞬态本地订阅 Use transient local
      always_send_full_costmap: True  # 总是发送完整代价地图 Always send the full
costmapv

local_costmap_client:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time

local_costmap_rclcpp_node:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time

global_costmap:
  global_costmap:
    ros__parameters:
      use_sim_time: False  # 是否使用仿真时间 Use simulation time
      footprint_padding: 0.03  # 足迹填充 Footprint padding
      update_frequency: 1.0    # 更新频率 Update frequency
      publish_frequency: 1.0    # 发布频率 Publish frequency
      global_frame: map         # 全局坐标系 Global coordinate system
      robot_base_frame: base_footprint  # 机器人基座坐标系 Robot base coordinate
system
      robot_radius: 0.1        # 机器人半径 Robot radius
      resolution: 0.05          # 分辨率 Resolution
      plugins: ["static_layer", "obstacle_layer", "voxel_layer",
"inflation_layer"]  # 图层插件 Layer plugin
      obstacle_layer:           # 障碍物层参数 Obstacle layer parameters
        plugin: "nav2_costmap_2d::ObstacleLayer"  # 障碍物层插件 Obstacle layer
plugin
        enabled: True           # 是否启用 Enable or disable
        observation_sources: scan  # 观测源（激光） Observation sources (laser)
        footprint_clearing_enabled: true  # 是否启用足迹清除 Enable footprint
clearing
        max_obstacle_height: 2.0  # 最大障碍物高度 Maximum obstacle height
        combination_method: 1    # 组合方法 Combination method
        scan:                   # 激光参数 Laser parameters
          topic: /scan           # 激光话题 Laser topic
          obstacle_range: 3.0    # 障碍物检测范围 Obstacle detection range
          obstacle_min_range: 0.3  # 最小障碍物距离 Minimum obstacle distance
          raytrace_range: 4.0    # 光线追踪范围 Ray tracing range
          max_obstacle_height: 1.0  # 最大障碍物高度 Maximum obstacle height
          min_obstacle_height: 0.0  # 最小障碍物高度 Minimum obstacle height
          clearing: True          # 是否用于清除空间 Clearing space
          marking: True           # 是否用于标记障碍 Marking obstacles
          data_type: "LaserScan" # 数据类型 Data type
          inf_is_valid: false     # 无限距离是否有效 Infinite distance valid
      voxel_layer:               # 体素层参数 Voxel layer parameters
        plugin: "nav2_costmap_2d::VoxelLayer"  # 体素层插件 Voxel layer plugin
        enabled: True            # 是否启用 Enable or disable
        footprint_clearing_enabled: true  # Enable or disable是否启用足迹清除
Whether to enable footprint clearing
        max_obstacle_height: 2.0  # 最大障碍物高度 Maximum obstacle height
        publish_voxel_map: True # 是否发布体素地图 Whether to publish voxel map
        origin_z: 0.0            # Z轴原点 Z-axis origin
        z_resolution: 0.05       # Z轴分辨率 Z-axis resolution
        z_voxels: 16             # Z轴体素数 Number of voxels on the Z axis
        unknown_threshold: 15    # 未知区域阈值 Unknown region threshold
        mark_threshold: 0        # 标记阈值 Marking threshold
        observation_sources: pointcloud  # 观测源（点云） Observation sources
(point cloud)
        combination_method: 1     # 组合方法 Combination method
        pointcloud:               # 点云参数 Point cloud parameters
          topic: /intel_realsense_r200_depth/points  # 点云话题 Point cloud topic
          max_obstacle_height: 2.0  # 最大障碍物高度 Maximum obstacle height
          min_obstacle_height: 0.0  # 最小障碍物高度 Minimum obstacle height
          obstacle_range: 2.5      # 障碍物检测范围 Obstacle detection range
          raytrace_range: 3.0      # 光线追踪范围 Ray tracing range
          clearing: True          # 是否用于清除空间 Clearing space
          marking: True           # 是否用于标记障碍 Marking obstacles
          data_type: "PointCloud2"  # 数据类型 Data type
      static_layer:              # 静态层参数 Static layer parameters
        plugin: "nav2_costmap_2d::StaticLayer"  # 静态层插件 Static layer plugin
        map_subscribe_transient_local: True  # 使用瞬态本地订阅 Use transient local
subscriptions
        enabled: true             # 是否启用 Enable or disable
        subscribe_to_updates: true  # 是否订阅更新 Whether to subscribe to updates
        transform_tolerance: 0.1  # 变换容差 Transformation tolerance
      inflation_layer:           # 膨胀层参数 Inflation layer parameters
        plugin: "nav2_costmap_2d::InflationLayer"  # 膨胀层插件 Inflation layer
plugin
        enabled: true             #  是否启用 Enable or disable
        cost_scaling_factor: 5.0  # 代价缩放因子 Cost scaling factor
        inflation_radius: 0.11    # 膨胀半径 Inflation radius
        inflate_unknown: false    # 是否膨胀未知区域 Whether to expand the unknown
area
        inflate_around_unknown: true  # 是否在未知区域周围膨胀 Whether to inflate
around unknown areas
      always_send_full_costmap: True  # 总是发送完整代价地图 Always send the full
costmap

global_costmap_client:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time

global_costmap_rclcpp_node:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time

map_server:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time
    yaml_filename: "yahboomcar.yaml"  # 地图YAML文件路径 Map YAML file path

map_saver:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time
    save_map_timeout: 5.0  # 保存地图超时时间 Map save timeout
    free_thresh_default: 0.25  # 空闲阈值默认值 Default idle threshold
    occupied_thresh_default: 0.65  # 占用阈值默认值 Default occupied threshold
    map_subscribe_transient_local: True  # 使用瞬态本地订阅 Use transient local
subscription

planner_server:
  ros__parameters:
    expected_planner_frequency: 20.0  # 期望规划器频率 Expected planner frequency
    use_sim_time: False  # 是否使用仿真时间 Use simulation time
    planner_plugins: ["GridBased"]  # 规划器插件列表 Planner plugin list
    GridBased:
      plugin: "nav2_navfn_planner/NavfnPlanner"  # Navfn规划器 Navfn planner
      tolerance: 0.5  # 目标点容差 Target point tolerance
      use_astar: false  # 是否使用A*算法 Use A* algorithm
      allow_unknown: true  # 是否允许未知区域 Whether to allow unknown regions

planner_server_rclcpp_node:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time

behavior_server:
  ros__parameters:
    costmap_topic: local_costmap/costmap_raw  # 代价地图话题 Costmap topic
    footprint_topic: local_costmap/published_footprint  # 足迹话题 Footprint topic
    cycle_frequency: 10.0  # 行为执行频率 Behavior execution frequency
    behavior_plugins: ["backup"]  # 行为插件列表 Behavior plugin list
    backup:  # 后退行为参数 Backup behavior parameters
      plugin: "nav2_behaviors/BackUp"  # 后退行为插件 Backup behavior plugin
      backup_speed: -1.0  # 后退速度 Backup speed
      backup_duration: 1.5  # 后退持续时间 Backup duration
      simulate_ahead_time: 2.0  # 模拟提前时间 Simulation advance time
    global_frame: odom  # 全局坐标系 Global coordinate system
    robot_base_frame: base_footprint  # 机器人基座坐标系 Robot base coordinate
system
    transform_timeout: 0.1  # 变换超时 Transformation timeout
    use_sim_time: False  # 是否使用仿真时间 Use simulation time
    simulate_ahead_time: 2.0  # 模拟提前时间 Simulation advance time
    max_rotational_vel: 1.0  # 最大旋转速度 Maximum rotational velocity
    min_rotational_vel: 0.4  # 最小旋转速度 Minimum rotational velocity
    rotational_acc_lim: 3.2  # 旋转加速度限制 Rotational acceleration limit

robot_state_publisher:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 Use simulation time

waypoint_follower:
  ros__parameters:
    loop_rate: 20  # 循环频率 Loop rate
    stop_on_failure: false  # 失败时是否停止 Stop on failure
    waypoint_task_executor_plugin: "wait_at_waypoint"  # 航点任务执行器插件 Waypoint
task executor plugin
    wait_at_waypoint:  # 航点等待参数 Waypoint wait parameters
```

```
      plugin: "nav2_waypoint_follower::WaitAtWaypoint"  # 航点等待插件 Waypoint Wait Plugin
      enabled: True  #  是否启用 Enable or disable
      waypoint_pause_duration: 200  # 航点暂停时间（毫秒） Waypoint pause duration (milliseconds)
```

The default configuration parameters for the DWA algorithm navigation function are as follows:

```
amcl:
  ros__parameters:
    use_sim_time: False  # 是否使用仿真时间 (Gazebo等) Whether to use simulation time (Gazebo, etc.)
    alpha1: 0.2          # 运动模型中的平移噪声参数 (x方向) Translational noise parameters (x-direction) in the motion model
    alpha2: 0.2          # 运动模型中的平移噪声参数 (y方向) Translational noise parameters (y-direction) in the motion model
    alpha3: 0.2          # 运动模型中的旋转噪声参数 Rotational noise parameters in the motion model
    alpha4: 0.2          # 运动模型中的平移噪声参数 (末端旋转) Translational noise parameters (end-end rotation) in the motion model
    alpha5: 0.2          # 运动模型中的旋转噪声参数 (末端旋转) Rotational noise parameters (end-effector rotation) in the motion model
    base_frame_id: "base_footprint"  # 机器人基座坐标系 Robot base coordinate system
    beam_skip_distance: 0.5          # 光束跳过距离(米) Beam jump distance (meters)
    beam_skip_error_threshold: 0.9   # 光束跳过错误阈值 Beam skips error threshold
    beam_skip_threshold: 0.3          # 光束跳过阈值 Beam skip threshold
    do_beamskip: false                # 是否启用光束跳过功能 Enable beam skip function?
    global_frame_id: "map"            # 全局地图坐标系 Global map coordinate system
    lambda_short: 0.1                 # 短距离测量的权重 Weighting of short-distance measurements
    laser_likelihood_max_dist: 2.0    # 激光似然场最大距离(米) Maximum distance of laser likelihood field (meters)
    laser_max_range: 100.0            # 激光最大测距范围(米) Maximum laser ranging range (meters)
    laser_min_range: -1.0             # 激光最小测距范围(米) Minimum laser ranging range (meters)
    laser_model_type: "likelihood_field"  # 激光模型类型 (似然场模型) Laser model type (likelihood field model)
    max_beams: 50                     # 每次更新使用的最大激光束数 Maximum number of laser beams used per update
    max_particles: 4000               # 最大粒子数 Maximum number of particles
    min_particles: 1000               # 最小粒子数 Minimum number of particles
    odom_frame_id: "odom"             # 里程计坐标系 Odometry coordinate system
    pf_err: 0.05                      # 粒子滤波器错误阈值 Particle filter error threshold
    pf_z: 0.99                        # 粒子滤波器z值 Particle filter z-value
    recovery_alpha_fast: 0.0          # 快速恢复alpha值 Fast alpha recovery
    recovery_alpha_slow: 0.0          # 慢速恢复alpha值 Slow alpha recovery
    resample_interval: 1              # 重采样间隔 Resampling interval
    robot_model_type: "nav2_amcl::DifferentialMotionModel"  # 机器人运动模型 (差速驱动) Robot motion model (differential drive)
    save_pose_rate: 0.5               # 保存位姿的速率(Hz) Pose saving rate (Hz)
    sigma_hit: 0.2                    # 激光测量噪声标准差 Standard deviation of laser measurement noise
    tf_broadcast: true                 # 是否广播TF变换 Whether to broadcast TF transform
    transform_tolerance: 1.0          # TF变换容忍时间(秒) TF transform tolerance time (seconds)
    update_min_a: 0.2                 # 最小角度变化触发更新(弧度) Minimum angle change triggers update (radians)
    update_min_d: 0.25                # 最小距离变化触发更新(米) Minimum distance change triggers update (meters)
    z_hit: 0.5                        # 命中测量的权重 Weight of hit measurement
    z_max: 0.05                       # 最大范围测量的权重 Weight of maximum range measurement
    z_rand: 0.5                       # 随机测量的权重 Weight of random measurement
    z_short: 0.05                     # 短距离测量的权重 Weight of short-range measurement
    scan_topic: scan                  # 激光扫描数据的话题名称 Topic name of laser scan data

    # 初始位置设置 (示例，当前被注释) Initial position settings (example, currently commented out)
    # set_initial_pose: true
    # initial_pose:
    #   x: 0.0
    #   y: 0.0
    #   z: 0.0
    #   yaw: 0.0

# AMCL相关组件的参数 Parameters of AMCL related components
amcl_map_client:
  ros__parameters:
    use_sim_time: False

amcl_rclcpp_node:
  ros__parameters:
    use_sim_time: False

# 行为树导航器配置 Behavior Tree Navigator Configuration
bt_navigator:
  ros__parameters:
    use_sim_time: False
    global_frame: map                  # 全局坐标系 Global coordinate system
    robot_base_frame: base_footprint   # 机器人基座坐标系 Robot base coordinate system
    odom_topic: /odom                  # 里程计数据的话题 The topic of odometer data
    default_bt_xml_filename: "navigate_to_pose_w_replanning_and_recovery.xml"  # 默认行为树文件 Default behavior tree file
    plugin_lib_names:                  # 行为树插件库列表 Behavior Tree Plugin Library List
      - 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

# 行为树导航器节点参数 Behavior Tree Navigator Node Parameters
bt_navigator_rclcpp_node:
  ros__parameters:
    use_sim_time: False

# 控制器服务器配置 Controller server configuration
controller_server:
  ros__parameters:
    use_sim_time: False
    controller_frequency: 10.0        # 控制器运行频率(Hz) Controller operating
frequency (Hz)
    min_x_velocity_threshold: 0.001    # X方向最小速度阈值(m/s) Minimum velocity
threshold in the X direction (m/s)
    min_y_velocity_threshold: 0.5       # Y方向最小速度阈值(m/s) Minimum velocity
threshold in the Y direction (m/s)
    min_theta_velocity_threshold: 0.001  # 旋转最小速度阈值(rad/s) Minimum
rotational speed threshold (rad/s)
    controller_plugins: ["FollowPath"]  # 使用的控制器插件列表 List of controller
plugins used
```

```
    # DWB控制器参数 DWB controller parameters
    FollowPath:
      plugin: "dwb_core::DWBLocalPlanner"  # 控制器类型 Controller Type
      debug_trajectory_details: True    # 是否调试轨迹细节 Debug track details
      min_vel_x: 0.0                    # 最小X方向速度(m/s) Minimum X-direction
velocity (m/s)
      min_vel_y: 0.0                    # 最小Y方向速度(m/s) Minimum Y-direction
velocity (m/s)
      max_vel_x: 0.3                    # 最大X方向速度(m/s) Maximum X-direction
velocity (m/s)
      max_vel_y: 0.0                    # 最大Y方向速度(m/s) Maximum Y-direction
velocity (m/s)
      max_vel_theta: 0.6                # 最大旋转速度(rad/s) Maximum rotational
speed (rad/s)
      min_speed_xy: 0.0                 # 最小XY平面速度(m/s) Minimum XY-plane
velocity (m/s)
      max_speed_xy: 0.3                 # 最大XY平面速度(m/s) Maximum XY-plane
velocity (m/s)
      min_speed_theta: 0.0              # 最小旋转速度(rad/s) Minimum rotational
speed (rad/s)
      acc_lim_x: 0.25                   # X方向加速度限制(m/s²) X-direction
acceleration limit (m/s²)
      acc_lim_y: 0.0                    # Y方向加速度限制(m/s²) Y-direction
acceleration limit (m/s²)
      acc_lim_theta: 0.4                # 旋转加速度限制(rad/s²) Rotational
acceleration limit (rad/s²)
      decel_lim_x: -0.25                # X方向减速度限制(m/s²) X-direction
deceleration limit (m/s²)
      decel_lim_y: 0.0                  # Y方向减速度限制(m/s²) Y-direction
deceleration limit (m/s²)
      decel_lim_theta: -0.4             # 旋转减速度限制(rad/s²) Rotational
deceleration limit (rad/s²)
      vx_samples: 20                    # X方向速度采样数 Number of X-direction
velocity samples
      vy_samples: 0                     # Y方向速度采样数 Number of Y-direction
velocity samples
      vtheta_samples: 40                # 旋转速度采样数 Number of rotational
velocity samples
      sim_time: 1.5                     # 轨迹模拟时间(s) Trajectory simulation
time (s)
      linear_granularity: 0.05          # 线速度粒度(m/s) Linear velocity
granularity (m/s)
      angular_granularity: 0.025        # 角速度粒度(rad/s) Angular velocity
granularity (rad/s)
      transform_tolerance: 0.2           # TF变换容忍时间(s) TF transform tolerance
time (s)
      xy_goal_tolerance: 0.15           # XY目标容差(m) XY target tolerance (m)
      trans_stopped_velocity: 0.25      # 停止时的平移速度(m/s) Translational
velocity at stop (m/s)
      short_circuit_trajectory_evaluation: True  # 是否使用短路轨迹评估 Whether to
use short-circuit trajectory evaluation
      stateful: True                    # 是否保持状态 Whether to maintain state
      critics:                          # 轨迹评估器列表 List of trajectory
evaluators
        - "RotateToGoal"
        - "Oscillation"
        - "BaseObstacle"
        - "GoalAlign"
        - "PathAlign"
        - "PathDist"
        - "GoalDist"
      BaseObstacle.scale: 0.02          # 基础障碍物评估器缩放因子 Base obstacle
estimator scaling factor
      PathAlign.scale: 16.0             # 路径对齐评估器缩放因子 Path alignment
estimator scaling factor
      PathAlign.forward_point_distance: 0.1  # 路径对齐前视点距离(m) Path alignment
forward point distance (m)
      GoalAlign.scale: 24.0             # 目标对齐评估器缩放因子 Goal alignment
estimator scaling factor
      GoalAlign.forward_point_distance: 0.1  # 目标对齐前视点距离(m) Goal alignment
forward point distance (m)
      PathDist.scale: 32.0              # 路径距离评估器缩放因子 Path distance
estimator scaling factor
      GoalDist.scale: 24.0              # 目标距离评估器缩放因子 Goal distance
estimator scaling factor
      RotateToGoal.scale: 64.0          # 旋转到目标评估器缩放因子 Rotate to Goal
estimator scaling factor
      RotateToGoal.slowing_factor: 5.0  # 旋转到目标减速因子 Rotate to Goal slowing
factor
      RotateToGoal.lookahead_time: -1.0 # 旋转到目标前视时间(s) Rotate to Goal
forward lookahead time (s)

# 控制器服务器节点参数 Controller server node parameters
controller_server_rclcpp_node:
  ros__parameters:
    use_sim_time: False

# 局部代价地图配置 Local cost map configuration
local_costmap:
  local_costmap:
    ros__parameters:
      update_frequency: 5.0            # 更新频率(Hz) Update frequency (Hz)
      publish_frequency: 2.0           # 发布频率(Hz) Publish frequency (Hz)
      global_frame: odom                # 全局坐标系 Global coordinate system
      robot_base_frame: base_footprint  # 机器人基座坐标系 Robot base coordinate
system
      use_sim_time: False
      rolling_window: true              # 是否使用滚动窗口 Whether to use a
scrolling window
      width: 3                          # 地图宽度(m) Map width (m)
      height: 3                         # 地图高度(m)Map height (m)
      resolution: 0.05                  # 分辨率(m/像素) Resolution (m/pixel)
      robot_radius: 0.1                 # 机器人半径(m) Robot radius (m)
      plugins: ["obstacle_layer", "inflation_layer"]  # 使用的图层 Layers used
      inflation_layer:
        plugin: "nav2_costmap_2d::InflationLayer"  # 膨胀层插件 Inflation layer
plugin
        inflation_radius: 0.22          # 膨胀半径(m) Inflation radius (m)
        cost_scaling_factor: 6.0        # 成本缩放因子 Cost scaling factor
      obstacle_layer:
        plugin: "nav2_costmap_2d::ObstacleLayer"  # 障碍物层插件 Obstacle layer
plugin
        enabled: True                   # 是否启用障碍物层Whether to enable the
obstacle layer
        observation_sources: scan       # 观测源 Observation sources
        scan:
          topic: /scan                  # 激光扫描话题 Laser scan topic
          max_obstacle_height: 2.0      # 最大障碍物高度(m) Maximum obstacle height
(m)
          obstacle_min_range: 0.1       # 障碍物最小范围(m) Minimum obstacle range
(m)
          clearing: True                # 是否清除障碍物 Whether to clear obstacles
          marking: True                 # 是否标记障碍物 Whether to mark obstacles
          data_type: "LaserScan"        # 数据类型 Data type
      static_layer:
        map_subscribe_transient_local: True  # 静态地图订阅传输方式 Static map
subscription transmission method
      always_send_full_costmap: True    # 是否始终发送完整代价地图 Whether to always
send the full cost map

  # 局部代价地图客户端参数 Local cost map client parameters
  local_costmap_client:
    ros__parameters:
      use_sim_time: False

  # 局部代价地图节点参数 Local cost map node parameters
  local_costmap_rclcpp_node:
    ros__parameters:
      use_sim_time: False

# 全局代价地图配置 Global Cost Map Configuration
global_costmap:
  global_costmap:
    ros__parameters:
      footprint_padding: 0.03           # 机器人轮廓填充(m) Robot contour filling
(m)
      update_frequency: 1.0             # 更新频率(Hz) Update frequency (Hz)
      publish_frequency: 1.0            # 发布频率(Hz) Publish frequency (Hz)
      global_frame: map                  # 全局坐标系 Global coordinate system
      robot_base_frame: base_footprint   # 机器人基座坐标系 Robot base coordinate
system
      use_sim_time: False
      robot_radius: 0.1                 # 机器人半径(m) Robot radius (m)
      resolution: 0.05                  # 分辨率(m/像素)  Resolution (m/pixel)
      plugins: ["static_layer", "obstacle_layer", "inflation_layer"]  # 使用的图层
Layers used
      obstacle_layer:
        plugin: "nav2_costmap_2d::ObstacleLayer"  # 障碍物层插件 Obstacle layer
plugin
        enabled: True                   # 是否启用障碍物层 Whether the obstacle
layer is enabled
        observation_sources: scan       # 观测源 Observation sources
        scan:
          topic: /scan                  # 激光扫描话题 Laser scan topic
          raytrace_range: 3.0           # 光线追踪范围(m) Raytrace range (m)
          obstacle_min_range: 0.1        # 障碍物最小范围(m) Minimum obstacle range
(m)
          max_obstacle_height: 2.0      # 最大障碍物高度(m) Maximum obstacle height
(m)
          clearing: True                # 是否清除障碍物 Whether to clear obstacles
          marking: True                 # 是否标记障碍物 Whether to mark obstacles
          data_type: "LaserScan"        # 数据类型 Data type
      static_layer:
        plugin: "nav2_costmap_2d::StaticLayer"  # 静态层插件 Static layer plugin
        map_subscribe_transient_local: True  # 静态地图订阅传输方式 Static map subscription transmission method
      inflation_layer:
        plugin: "nav2_costmap_2d::InflationLayer"  # 膨胀层插件 Inflation layer plugin
        cost_scaling_factor: 6.0        # 成本缩放因子 Cost scaling factor
        inflation_radius: 0.22          # 膨胀半径(m) Inflation radius (m)
      always_send_full_costmap: True    # 是否始终发送完整代价地图 Whether to always send the full cost map

  # 全局代价地图客户端参数 Global Cost Map Client Parameters
  global_costmap_client:
    ros__parameters:
      use_sim_time: False

  # 全局代价地图节点参数 Global cost map node parameters
  global_costmap_rclcpp_node:
    ros__parameters:
      use_sim_time: False

# 地图服务器配置 Map server configuration
map_server:
  ros__parameters:
    use_sim_time: False
    yaml_filename: "yahboomcar.yaml"    # 地图文件路径 Map file path

# 地图保存器配置 Map saver configuration
map_saver:
  ros__parameters:
    use_sim_time: False
    save_map_timeout: 5000              # 保存地图超时时间(ms) Map save timeout (ms)
    free_thresh_default: 0.25           # 自由空间阈值 Free space threshold
    occupied_thresh_default: 0.65        # 占用空间阈值 Space occupied threshold

# 规划器服务器配置 Planner server configuration
planner_server:
  ros__parameters:
    expected_planner_frequency: 20.0    # 预期规划器频率(Hz) Expected planner frequency (Hz)
    use_sim_time: False
    planner_plugins: ["GridBased"]      # 使用的规划器插件 Planner plugin used
    GridBased:
      plugin: "nav2_navfn_planner/NavfnPlanner"  # 规划器类型 Planner type
      tolerance: 0.5                    # 目标点容差(m) Target point tolerance (m)
      use_astar: false                  # 是否使用A*算法 Should the A* algorithm be used?
      allow_unknown: true               # 是否允许在未知区域规划 Is planning permitted in unknown areas?

# 规划器服务器节点参数 Planner server node parameters
planner_server_rclcpp_node:
  ros__parameters:
    use_sim_time: False

# 行为服务器配置 Behavior server configuration
behavior_server:
  ros__parameters:
    costmap_topic: local_costmap/costmap_raw  # 代价地图话题 Cost Map Topic
    footprint_topic: local_costmap/published_footprint  # 机器人轮廓话题 Robot
outline topic
    cycle_frequency: 10.0                # 行为执行频率(Hz) Behavior execution
frequency (Hz)
    behavior_plugins: ["spin", "backup", "wait"]  # 行为插件列表 List of behavior
plugins
    spin:
      plugin: "nav2_behaviors/Spin"     # 旋转行为插件 Rotation behavior plugi
      spin_type: "SPIN_IN_PLACE"        # 旋转类型 (原地旋转)Rotation type
(rotation in place)
      rotation_velocity: 1.0            # 旋转速度(rad/s) Rotation speed (rad/s
      max_rotational_vel: 1.0           # 最大旋转速度(rad/s) Maximum rotation
speed (rad/s)
      min_rotational_vel: 0.4           # 最小旋转速度(rad/s) Minimum rotation
speed (rad/s
      rotational_acc_lim: 3.2           # 旋转加速度限制(rad/s²)Rotation
acceleration limit (rad/s²)
    backup:
      plugin: "nav2_behaviors/BackUp"   # 后退行为插件 Back-up behavior plugin
      backup_speed: -0.2                # 后退速度(m/s) Back-up speed (m/s)
      backup_duration: 2.0              # 后退持续时间(s) Back-up duration (s)
      simulate_ahead_time: 2.0          # 模拟提前时间(s) imulation lead time (s)
    wait:
      plugin: "nav2_behaviors/Wait"     # 等待行为插件 it behavior plugin
      wait_duration: 1.0                # 等待时间(s) ait time (s)
    global_frame: odom                  # 全局坐标系 obal coordinate system
    robot_base_frame: base_footprint    # 机器人基座坐标系 Robot base coordinate
system
    transform_timeout: 0.1              # 变换超时时间(s) ransformation timeout (s)
    use_sim_time: False
    simulate_ahead_time: 2.0            # 模拟提前时间(s) imulation lead time (s)
    max_rotational_vel: 1.0             # 最大旋转速度(rad/s) Maximum rotational
speed (rad/s)
    min_rotational_vel: 0.4             # 最小旋转速度(rad/s) Minimum rotational
speed (rad/s)
    rotational_acc_lim: 3.2             # 旋转加速度限制(rad/s²) Rotational
acceleration limit (rad/s²)

# 机器人状态发布器参数 Robot Status Publisher Parameters
robot_state_publisher:
  ros__parameters:
    use_sim_time: False

# 航点跟随器配置 Waypoint follower configuration
waypoint_follower:
  ros__parameters:
    use_sim_time: False
    loop_rate: 20                       # 循环频率(Hz) Loop rate (Hz)
    stop_on_failure: false              # 失败时是否停止 Whether to stop on failure
    waypoint_task_executor_plugin: "wait_at_waypoint"  # 航点任务执行器插件 Waypoint
task executor plugin
    wait_at_waypoint:
      plugin: "nav2_waypoint_follower::WaitAtWaypoint"  # 等待插件 Waiting plugin
      enabled: True                     # 是否启用 Whether to enable
      waypoint_pause_duration: 200      # 航点暂停时间(ms) Waypoint pause time (ms)
```