Multi-vehicle navigation

1. Program Description

This function allows you to assign two cars different navigation points in RVIZ. The two cars will plan routes based on their respective positions on the map and navigate to the designated locations, avoiding obstacles in real time.

1.1. Functional Requirements

For more information, please refer to [12.Multi-vehicle Course] -- [1.Multi-vehicle control] -- [1.1. Functional Requirements]

1.2. Navigation Map

Before starting a multi-car platoon, you need to place the map file on the mainboard or in a virtual machine.

```
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps/yahboomcar.yaml
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps/yahboomcar.pgm
```

The map file includes a .yaml parameter file and a .pgm image file.

Note: Both cars require the same map file

2. Program Startup

Using the Raspberry Pi 5 as an example, due to performance issues, a virtual machine is required for visualization.

The virtual machine must be on the same local area network as both robots, and the ROS_DOMAIN_ID must be set to the same for both robots. Modify the contents of ~/.bashrc and refresh the environment variables after the modification.

For Raspberry Pi and Jetson-Nano boards, open a terminal on the host machine 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 machine, refer to **[01. Robot Configuration and Operation Guide] -- [5.Enter Docker (For JETSON Nano and RPi 5)]**.

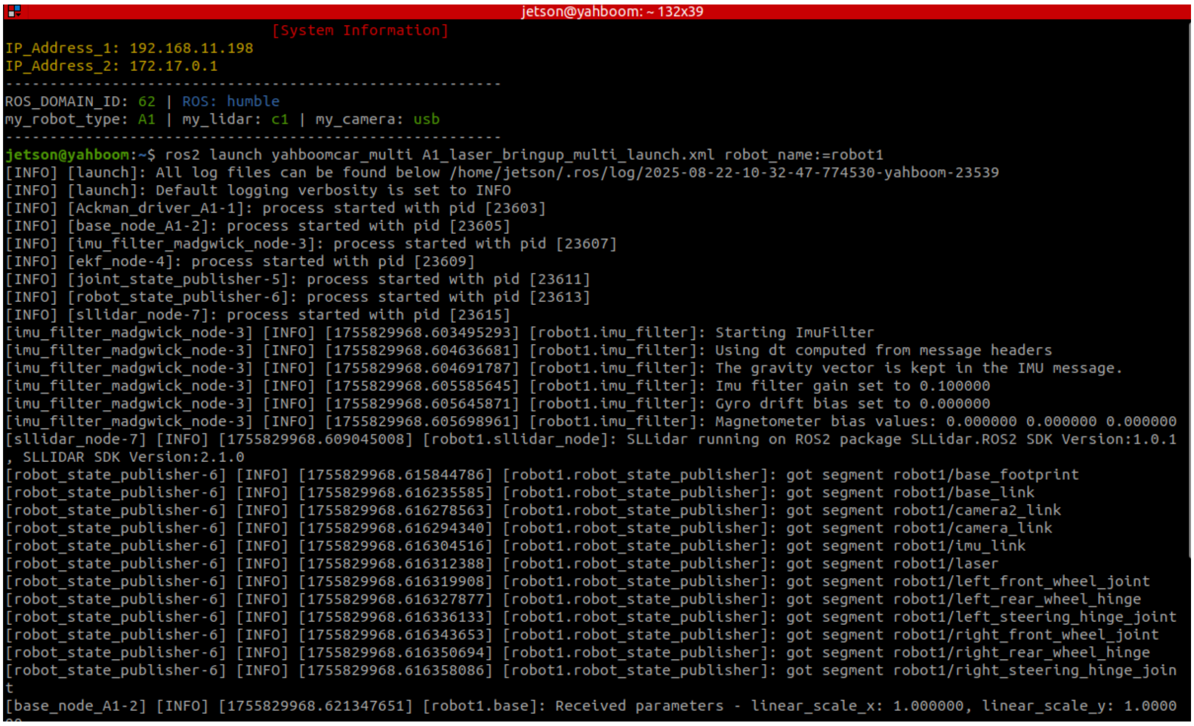
For Orin boards, simply open a terminal and enter the commands mentioned in this lesson.

This example uses two robots, Robot1 and Robot2. Please place them nearby in advance to ensure smooth network connectivity. This example requires a high data transfer rate.

2.1. Starting Chassis Data

On Robot1, open the terminal and enter the following command to start chassis data fusion, including radar data, filtered IMU and ODOM data, and EKF fusion.

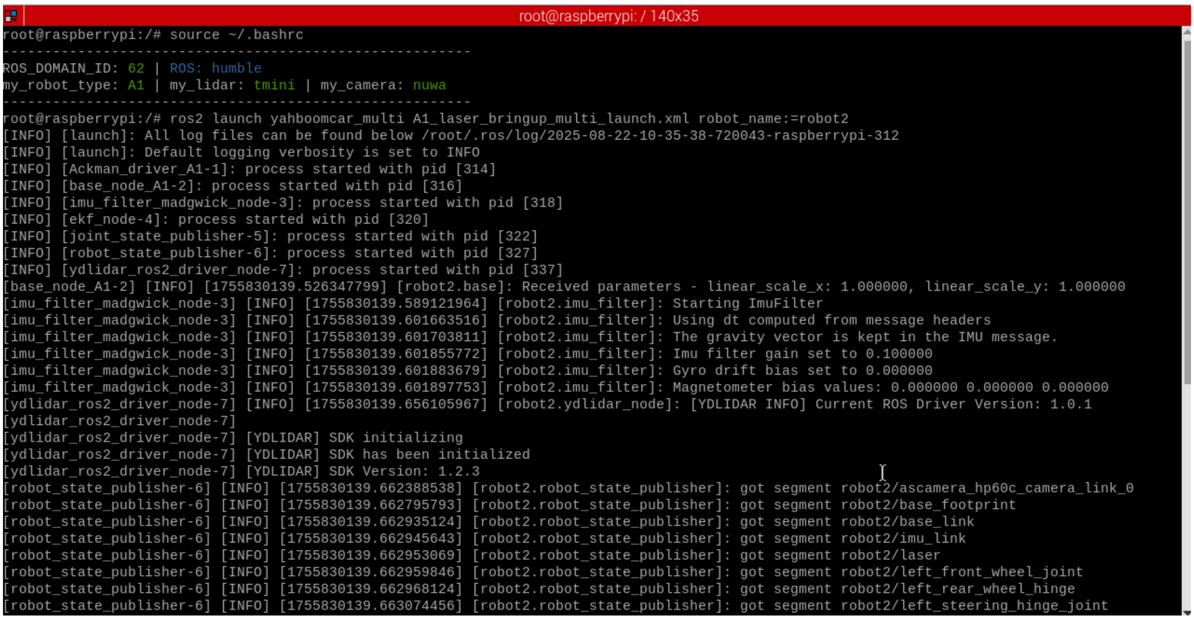
```
ros2 launch yahboomcar_multi A1_laser_bringup_multi_launch.xml
robot_name:=robot1
```



```
                              [System Information]
IP_Address_1: 192.168.11.198
IP_Address_2: 172.17.0.1
-------------------------------------------------------
ROS_DOMAIN_ID: 62 | ROS: humble
my_robot_type: A1 | my_lidar: c1 | my_camera: usb
-------------------------------------------------------
jetson@yahboom:~$ ros2 launch yahboomcar_multi A1_laser_bringup_multi_launch.xml robot_name:=robot1
[INFO] [launch]: All log files can be found below /home/jetson/.ros/log/2025-08-22-10-32-47-774530-yahboom-23539
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [Ackman_driver_A1-1]: process started with pid [23603]
[INFO] [base_node_A1-2]: process started with pid [23605]
[INFO] [imu_filter_madgwick_node-3]: process started with pid [23607]
[INFO] [ekf_node-4]: process started with pid [23609]
[INFO] [joint_state_publisher-5]: process started with pid [23611]
[INFO] [robot_state_publisher-6]: process started with pid [23613]
[INFO] [sllidar_node-7]: process started with pid [23615]
[imu_filter_madgwick_node-3] [INFO] [1755829968.603495293] [robot1.imu_filter]: Starting ImuFilter
[imu_filter_madgwick_node-3] [INFO] [1755829968.604636681] [robot1.imu_filter]: Using dt computed from message headers
[imu_filter_madgwick_node-3] [INFO] [1755829968.604691787] [robot1.imu_filter]: The gravity vector is kept in the IMU message.
[imu_filter_madgwick_node-3] [INFO] [1755829968.605585645] [robot1.imu_filter]: Imu filter gain set to 0.100000
[imu_filter_madgwick_node-3] [INFO] [1755829968.605645871] [robot1.imu_filter]: Gyro drift bias set to 0.000000
[imu_filter_madgwick_node-3] [INFO] [1755829968.605698961] [robot1.imu_filter]: Magnetometer bias values: 0.000000 0.000000 0.000000
[sllidar_node-7] [INFO] [1755829968.609045008] [robot1.sllidar_node]: SLLidar running on ROS2 package SLLidar.ROS2 SDK Version:1.0.1
, SLLIDAR SDK Version:2.1.0
[robot_state_publisher-6] [INFO] [1755829968.615844786] [robot1.robot_state_publisher]: got segment robot1/base_footprint
[robot_state_publisher-6] [INFO] [1755829968.616235585] [robot1.robot_state_publisher]: got segment robot1/base_link
[robot_state_publisher-6] [INFO] [1755829968.616278563] [robot1.robot_state_publisher]: got segment robot1/camera2_link
[robot_state_publisher-6] [INFO] [1755829968.616294340] [robot1.robot_state_publisher]: got segment robot1/camera_link
[robot_state_publisher-6] [INFO] [1755829968.616304516] [robot1.robot_state_publisher]: got segment robot1/imu_link
[robot_state_publisher-6] [INFO] [1755829968.616312388] [robot1.robot_state_publisher]: got segment robot1/laser
[robot_state_publisher-6] [INFO] [1755829968.616319908] [robot1.robot_state_publisher]: got segment robot1/left_front_wheel_joint
[robot_state_publisher-6] [INFO] [1755829968.616327877] [robot1.robot_state_publisher]: got segment robot1/left_rear_wheel_hinge
[robot_state_publisher-6] [INFO] [1755829968.616336133] [robot1.robot_state_publisher]: got segment robot1/left_steering_hinge_joint
[robot_state_publisher-6] [INFO] [1755829968.616343653] [robot1.robot_state_publisher]: got segment robot1/right_front_wheel_joint
[robot_state_publisher-6] [INFO] [1755829968.616350694] [robot1.robot_state_publisher]: got segment robot1/right_rear_wheel_hinge
[robot_state_publisher-6] [INFO] [1755829968.616358086] [robot1.robot_state_publisher]: got segment robot1/right_steering_hinge_join
t
[base_node_A1-2] [INFO] [1755829968.621347651] [robot1.base]: Received parameters - linear_scale_x: 1.000000, linear_scale_y: 1.0000
```

Similarly, on Robot2, open the terminal and enter the following command to start chassis data fusion.

```
ros2 launch yahboomcar_multi A1_laser_bringup_multi_launch.xml
robot_name:=robot2
```



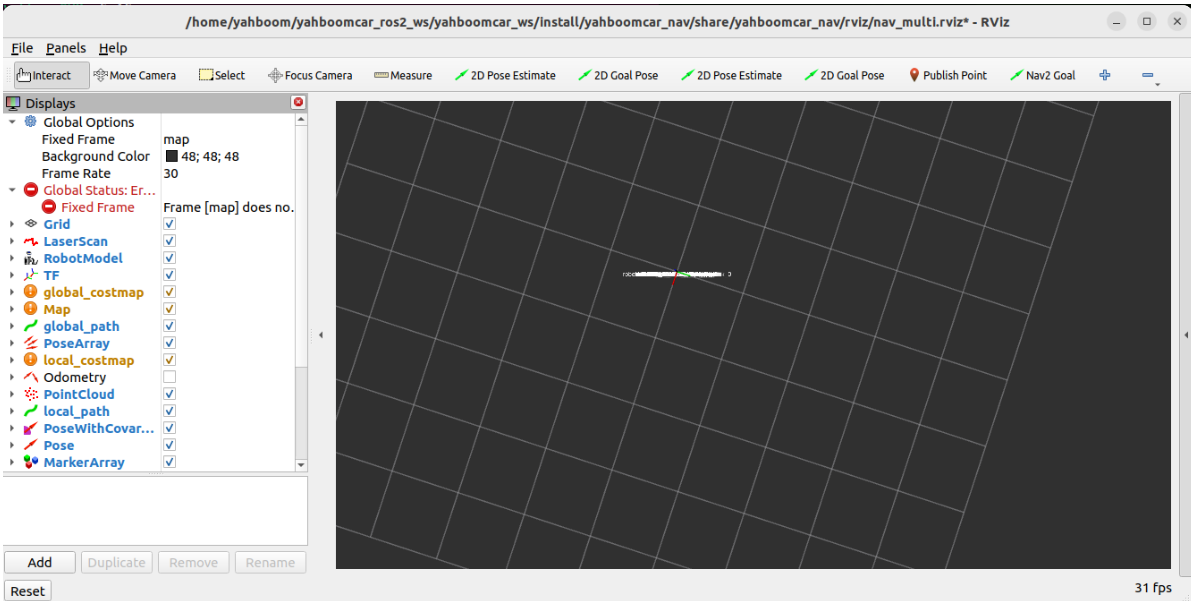
```
root@raspberrypi:/# source ~/.bashrc
-------------------------------------------------------
ROS_DOMAIN_ID: 62 | ROS: humble
my_robot_type: A1 | my_lidar: tmini | my_camera: nuwa
-------------------------------------------------------
root@raspberrypi:/# ros2 launch yahboomcar_multi A1_laser_bringup_multi_launch.xml robot_name:=robot2
[INFO] [launch]: All log files can be found below /root/.ros/log/2025-08-22-10-35-38-720043-raspberrypi-312
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [Ackman_driver_A1-1]: process started with pid [314]
[INFO] [base_node_A1-2]: process started with pid [316]
[INFO] [imu_filter_madgwick_node-3]: process started with pid [318]
[INFO] [ekf_node-4]: process started with pid [320]
[INFO] [joint_state_publisher-5]: process started with pid [322]
[INFO] [robot_state_publisher-6]: process started with pid [327]
[INFO] [ydlidar_ros2_driver_node-7]: process started with pid [337]
[base_node_A1-2] [INFO] [1755830139.526347799] [robot2.base]: Received parameters - linear_scale_x: 1.000000, linear_scale_y: 1.000000
[imu_filter_madgwick_node-3] [INFO] [1755830139.589121964] [robot2.imu_filter]: Starting ImuFilter
[imu_filter_madgwick_node-3] [INFO] [1755830139.601663516] [robot2.imu_filter]: Using dt computed from message headers
[imu_filter_madgwick_node-3] [INFO] [1755830139.601703811] [robot2.imu_filter]: The gravity vector is kept in the IMU message.
[imu_filter_madgwick_node-3] [INFO] [1755830139.601855772] [robot2.imu_filter]: Imu filter gain set to 0.100000
[imu_filter_madgwick_node-3] [INFO] [1755830139.601883679] [robot2.imu_filter]: Gyro drift bias set to 0.000000
[imu_filter_madgwick_node-3] [INFO] [1755830139.601897753] [robot2.imu_filter]: Magnetometer bias values: 0.000000 0.000000 0.000000
[ydlidar_ros2_driver_node-7] [INFO] [1755830139.656105967] [robot2.ydlidar_node]: [YDLIDAR INFO] Current ROS Driver Version: 1.0.1
[ydlidar_ros2_driver_node-7]
[ydlidar_ros2_driver_node-7] [YDLIDAR] SDK initializing
[ydlidar_ros2_driver_node-7] [YDLIDAR] SDK has been initialized
[ydlidar_ros2_driver_node-7] [YDLIDAR] SDK Version: 1.2.3
[robot_state_publisher-6] [INFO] [1755830139.662388538] [robot2.robot_state_publisher]: got segment robot2/ascamera_hp60c_camera_link_0
[robot_state_publisher-6] [INFO] [1755830139.662795793] [robot2.robot_state_publisher]: got segment robot2/base_footprint
[robot_state_publisher-6] [INFO] [1755830139.662935124] [robot2.robot_state_publisher]: got segment robot2/base_link
[robot_state_publisher-6] [INFO] [1755830139.662945643] [robot2.robot_state_publisher]: got segment robot2/imu_link
[robot_state_publisher-6] [INFO] [1755830139.662953069] [robot2.robot_state_publisher]: got segment robot2/laser
[robot_state_publisher-6] [INFO] [1755830139.662959846] [robot2.robot_state_publisher]: got segment robot2/left_front_wheel_joint
[robot_state_publisher-6] [INFO] [1755830139.662968124] [robot2.robot_state_publisher]: got segment robot2/left_rear_wheel_hinge
[robot_state_publisher-6] [INFO] [1755830139.663074456] [robot2.robot_state_publisher]: got segment robot2/left_steering_hinge_joint
```

2.2. Launching rviz and Publishing Map Data

In the virtual machine configured for multi-machine communication, open a terminal and enter the following command to launch rviz.

```
ros2 launch yahboomcar_multi display_multi_nav_launch.py
```

After launching, the image below appears.



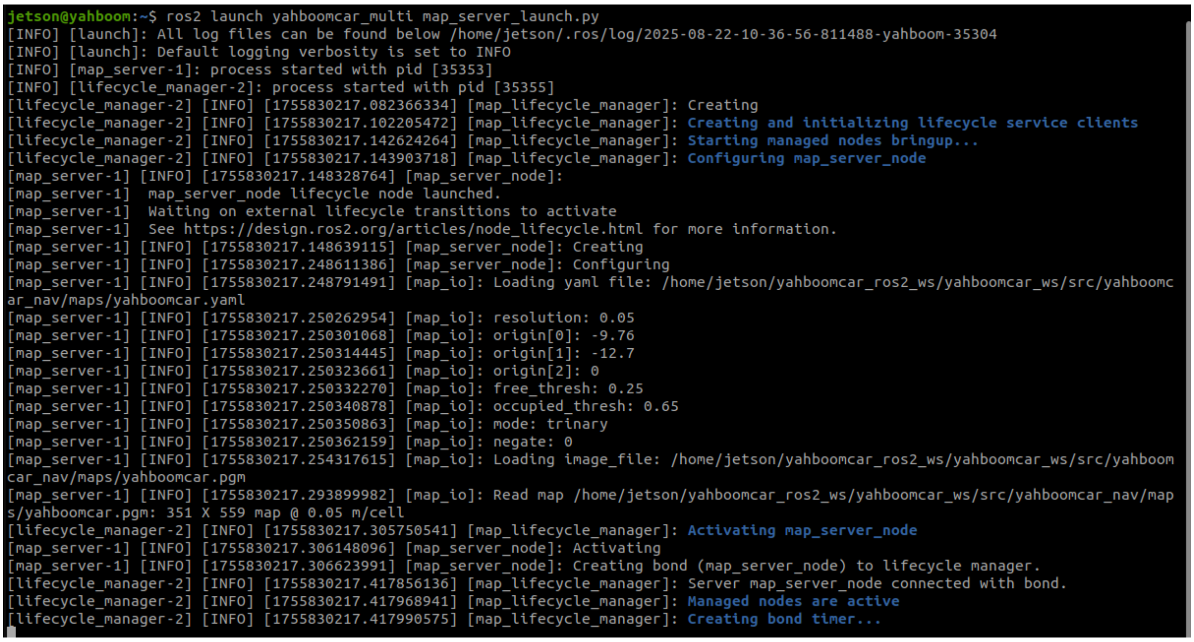
```
Global Options
    Fixed Frame          map
    Background Color     48; 48; 48
    Frame Rate           30
Global Status: Er...
    Fixed Frame          Frame [map] does no.
Grid                 ☑
LaserScan            ☑
RobotModel           ☑
TF                   ☑
global_costmap       ☑
Map                  ☑
global_path          ☑
PoseArray            ☑
local_costmap        ☑
Odometry             □
PointCloud           ☑
local_path           ☑
PoseWithCovar...     ☑
Pose                 ☑
MarkerArray          ☑
```

Then launch the map loading program. The default map launched is yahboomcar.yaml, and the file path is:

```
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps/yahboomcar.yaml
```

Enter the following command in the terminal to launch (you only need to load the map service in one robot's terminal).

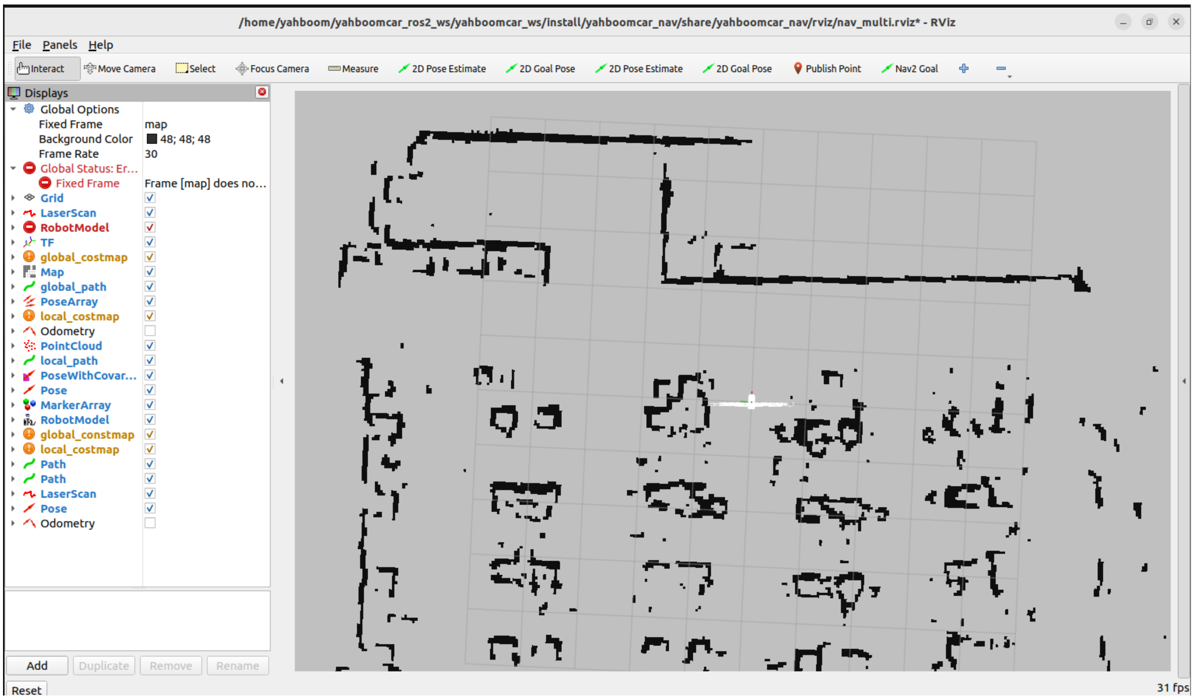
```
ros2 launch yahboomcar_multi map_server_launch.py
```



```
jetson@yahboom:~$ ros2 launch yahboomcar_multi map_server_launch.py
[INFO] [launch]: All log files can be found below /home/jetson/.ros/log/2025-08-22-10-36-56-811488-yahboom-35304
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [map_server-1]: process started with pid [35353]
[INFO] [lifecycle_manager-2]: process started with pid [35355]
[lifecycle_manager-2] [INFO] [1755830217.082366334] [map_lifecycle_manager]: Creating
[lifecycle_manager-2] [INFO] [1755830217.102205472] [map_lifecycle_manager]: Creating and initializing lifecycle service clients
[lifecycle_manager-2] [INFO] [1755830217.142624264] [map_lifecycle_manager]: Starting managed nodes bringup...
[lifecycle_manager-2] [INFO] [1755830217.143903718] [map_lifecycle_manager]: Configuring map_server_node
[map_server-1] [INFO] [1755830217.148328764] [map_server_node]:
[map_server-1]  map_server_node lifecycle node launched.
[map_server-1]  Waiting on external lifecycle transitions to activate
[map_server-1]  See https://design.ros2.org/articles/node_lifecycle.html for more information.
[map_server-1] [INFO] [1755830217.148639115] [map_server_node]: Creating
[map_server-1] [INFO] [1755830217.248611386] [map_server_node]: Configuring
[map_server-1] [INFO] [1755830217.248791491] [map_io]: Loading yaml file: /home/jetson/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomc
ar_nav/maps/yahboomcar.yaml
[map_server-1] [INFO] [1755830217.250262954] [map_io]: resolution: 0.05
[map_server-1] [INFO] [1755830217.250301068] [map_io]: origin[0]: -9.76
[map_server-1] [INFO] [1755830217.250314445] [map_io]: origin[1]: -12.7
[map_server-1] [INFO] [1755830217.250323661] [map_io]: origin[2]: 0
[map_server-1] [INFO] [1755830217.250332270] [map_io]: free_thresh: 0.25
[map_server-1] [INFO] [1755830217.250340878] [map_io]: occupied_thresh: 0.65
[map_server-1] [INFO] [1755830217.250350863] [map_io]: mode: trinary
[map_server-1] [INFO] [1755830217.250362159] [map_io]: negate: 0
[map_server-1] [INFO] [1755830217.254317615] [map_io]: Loading image_file: /home/jetson/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboom
car_nav/maps/yahboomcar.pgm
[map_server-1] [INFO] [1755830217.293899982] [map_io]: Read map /home/jetson/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/map
s/yahboomcar.pgm: 351 X 559 map @ 0.05 m/cell
[lifecycle_manager-2] [INFO] [1755830217.305750541] [map_lifecycle_manager]: Activating map_server_node
[map_server-1] [INFO] [1755830217.306148096] [map_server_node]: Activating
[map_server-1] [INFO] [1755830217.306623991] [map_server_node]: Creating bond (map_server_node) to lifecycle manager.
[lifecycle_manager-2] [INFO] [1755830217.417856136] [map_lifecycle_manager]: Server map_server_node connected with bond.
[lifecycle_manager-2] [INFO] [1755830217.417968941] [map_lifecycle_manager]: Managed nodes are active
[lifecycle_manager-2] [INFO] [1755830217.417990575] [map_lifecycle_manager]: Creating bond timer...
```

Next, on the virtual machine, our rviz visualization loads the created map.

(Due to network issues, the map may not be transferred to the virtual machine. Press Ctrl+C to shut it down and then restart the map loading service.)

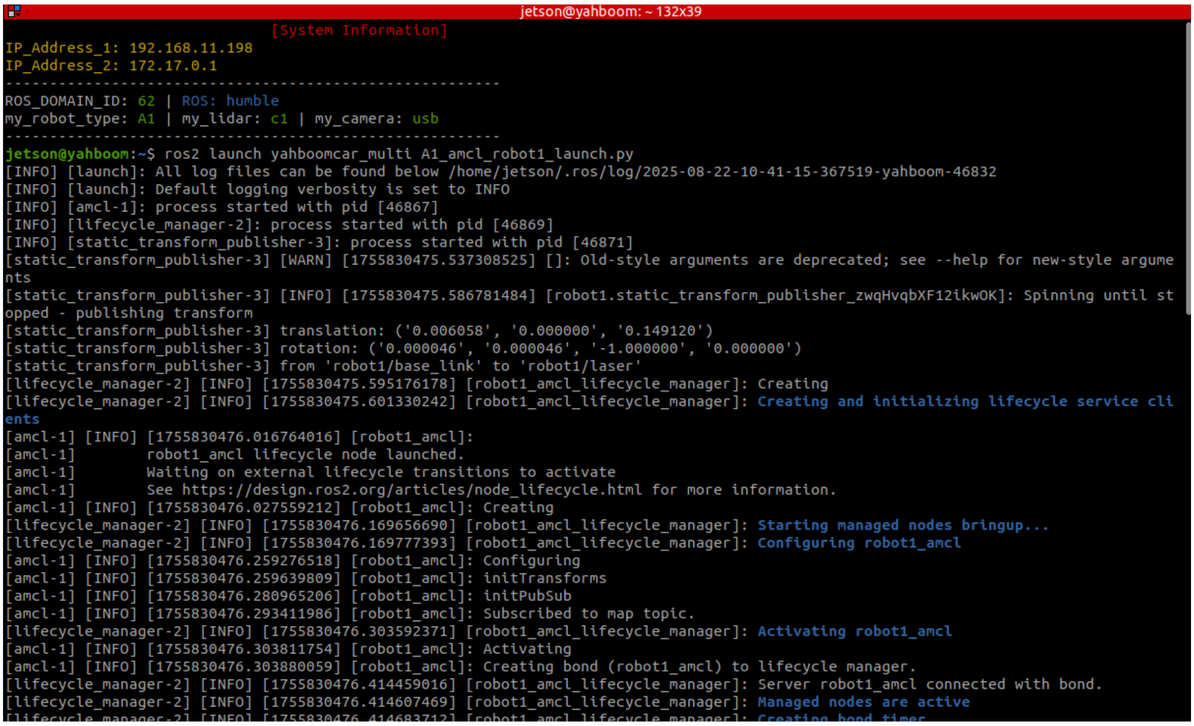


Fixed Frame map
Background Color 48; 48; 48
Frame Rate 30
Fixed Frame Frame [map] does no...

2.3. Start AMD positioning

Open the terminal on robot1 and enter the following command to start AMD positioning.

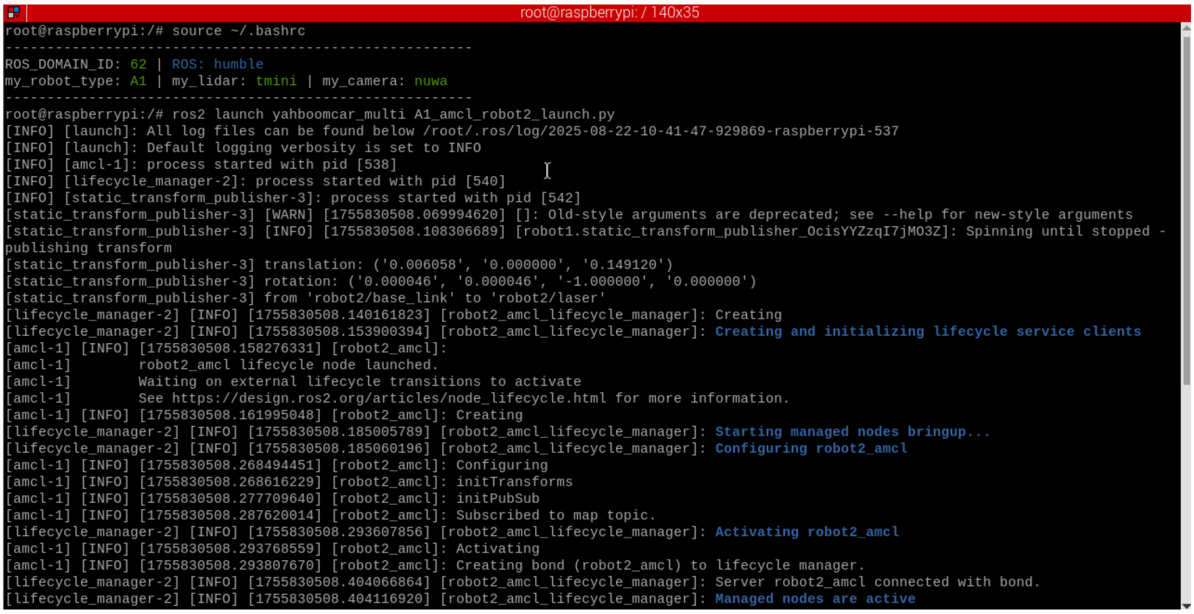
```
ros2 launch yahboomcar_multi A1_amcl_robot1_launch.py
```



```
jetson@yahboom: ~ 132x39
                                                  [System Information]
IP_Address_1: 192.168.11.198
IP_Address_2: 172.17.0.1
--------------------------------------------------------
ROS_DOMAIN_ID: 62 | ROS: humble
my_robot_type: A1 | my_lidar: c1 | my_camera: usb
--------------------------------------------------------
jetson@yahboom:~$ ros2 launch yahboomcar_multi A1_amcl_robot1_launch.py
[INFO] [launch]: All log files can be found below /home/jetson/.ros/log/2025-08-22-10-41-15-367519-yahboom-46832
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [amcl-1]: process started with pid [46867]
[INFO] [lifecycle_manager-2]: process started with pid [46869]
[INFO] [static_transform_publisher-3]: process started with pid [46871]
[static_transform_publisher-3] [WARN] [1755830475.537308525] []: Old-style arguments are deprecated; see --help for new-style argume
nts
[static_transform_publisher-3] [INFO] [1755830475.586781484] [robot1.static_transform_publisher_zwqHvqbXF12ikwOK]: Spinning until st
opped - publishing transform
[static_transform_publisher-3] translation: ('0.006058', '0.000000', '0.149120')
[static_transform_publisher-3] rotation: ('0.000046', '0.000046', '-1.000000', '0.000000')
[static_transform_publisher-3] from 'robot1/base_link' to 'robot1/laser'
[lifecycle_manager-2] [INFO] [1755830475.595176178] [robot1_amcl_lifecycle_manager]: Creating
[lifecycle_manager-2] [INFO] [1755830475.601330242] [robot1_amcl_lifecycle_manager]: Creating and initializing lifecycle service cli
ents
[amcl-1] [INFO] [1755830476.016764016] [robot1_amcl]:
[amcl-1]        robot1_amcl lifecycle node launched.
[amcl-1]        Waiting on external lifecycle transitions to activate
[amcl-1]        See https://design.ros2.org/articles/node_lifecycle.html for more information.
[amcl-1] [INFO] [1755830476.027559212] [robot1_amcl]: Creating
[lifecycle_manager-2] [INFO] [1755830476.169656690] [robot1_amcl_lifecycle_manager]: Starting managed nodes bringup...
[lifecycle_manager-2] [INFO] [1755830476.169777393] [robot1_amcl_lifecycle_manager]: Configuring robot1_amcl
[amcl-1] [INFO] [1755830476.259276518] [robot1_amcl]: Configuring
[amcl-1] [INFO] [1755830476.259639809] [robot1_amcl]: initTransforms
[amcl-1] [INFO] [1755830476.280965206] [robot1_amcl]: initPubSub
[amcl-1] [INFO] [1755830476.293411986] [robot1_amcl]: Subscribed to map topic.
[lifecycle_manager-2] [INFO] [1755830476.303592371] [robot1_amcl_lifecycle_manager]: Activating robot1_amcl
[amcl-1] [INFO] [1755830476.303811754] [robot1_amcl]: Activating
[amcl-1] [INFO] [1755830476.303880059] [robot1_amcl]: Creating bond (robot1_amcl) to lifecycle manager.
[lifecycle_manager-2] [INFO] [1755830476.414459016] [robot1_amcl_lifecycle_manager]: Server robot1_amcl connected with bond.
[lifecycle_manager-2] [INFO] [1755830476.414607469] [robot1_amcl_lifecycle_manager]: Managed nodes are active
[lifecycle_manager-2] [INFO] [1755830476.414683712] [robot1_amcl_lifecycle_manager]: Creating bond timer
```

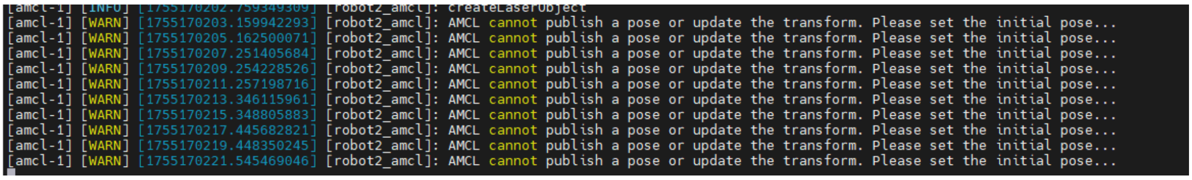
Open the terminal on the robot2 car and enter the following command to start AMCL positioning.

```
ros2 launch yahboomcar_multi A1_amcl_robot2_launch.py
```



```
root@raspberrypi:/# source ~/.bashrc
-------------------------------------------------------
ROS_DOMAIN_ID: 62 | ROS: humble
my_robot_type: A1 | my_lidar: tmini | my_camera: nuwa
-------------------------------------------------------
root@raspberrypi:/# ros2 launch yahboomcar_multi A1_amcl_robot2_launch.py
[INFO] [launch]: All log files can be found below /root/.ros/log/2025-08-22-10-41-47-929869-raspberrypi-537
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [amcl-1]: process started with pid [538]
[INFO] [lifecycle_manager-2]: process started with pid [540]
[INFO] [static_transform_publisher-3]: process started with pid [542]
[static_transform_publisher-3] [WARN] [1755830508.069994620] []: Old-style arguments are deprecated; see --help for new-style arguments
[static_transform_publisher-3] [INFO] [1755830508.108306689] [robot1.static_transform_publisher_OcisYYZzqI7jMO3Z]: Spinning until stopped -
publishing transform
[static_transform_publisher-3] translation: ('0.006058', '0.000000', '0.149120')
[static_transform_publisher-3] rotation: ('0.000046', '0.000046', '-1.000000', '0.000000')
[static_transform_publisher-3] from 'robot2/base_link' to 'robot2/laser'
[lifecycle_manager-2] [INFO] [1755830508.140161823] [robot2_amcl_lifecycle_manager]: Creating
[lifecycle_manager-2] [INFO] [1755830508.153900394] [robot2_amcl_lifecycle_manager]: Creating and initializing lifecycle service clients
[amcl-1] [INFO] [1755830508.158276331] [robot2_amcl]:
[amcl-1]        robot2_amcl lifecycle node launched.
[amcl-1]        Waiting on external lifecycle transitions to activate
[amcl-1]        See https://design.ros2.org/articles/node_lifecycle.html for more information.
[amcl-1] [INFO] [1755830508.161995048] [robot2_amcl]: Creating
[lifecycle_manager-2] [INFO] [1755830508.185005789] [robot2_amcl_lifecycle_manager]: Starting managed nodes bringup...
[lifecycle_manager-2] [INFO] [1755830508.185060196] [robot2_amcl_lifecycle_manager]: Configuring robot2_amcl
[amcl-1] [INFO] [1755830508.268494451] [robot2_amcl]: Configuring
[amcl-1] [INFO] [1755830508.268616229] [robot2_amcl]: initTransforms
[amcl-1] [INFO] [1755830508.277709640] [robot2_amcl]: initPubSub
[amcl-1] [INFO] [1755830508.287620014] [robot2_amcl]: Subscribed to map topic.
[lifecycle_manager-2] [INFO] [1755830508.293607856] [robot2_amcl_lifecycle_manager]: Activating robot2_amcl
[amcl-1] [INFO] [1755830508.293768559] [robot2_amcl]: Activating
[amcl-1] [INFO] [1755830508.293807670] [robot2_amcl]: Creating bond (robot2_amcl) to lifecycle manager.
[lifecycle_manager-2] [INFO] [1755830508.404066864] [robot2_amcl_lifecycle_manager]: Server robot2_amcl connected with bond.
[lifecycle_manager-2] [INFO] [1755830508.404116920] [robot2_amcl_lifecycle_manager]: Managed nodes are active
```

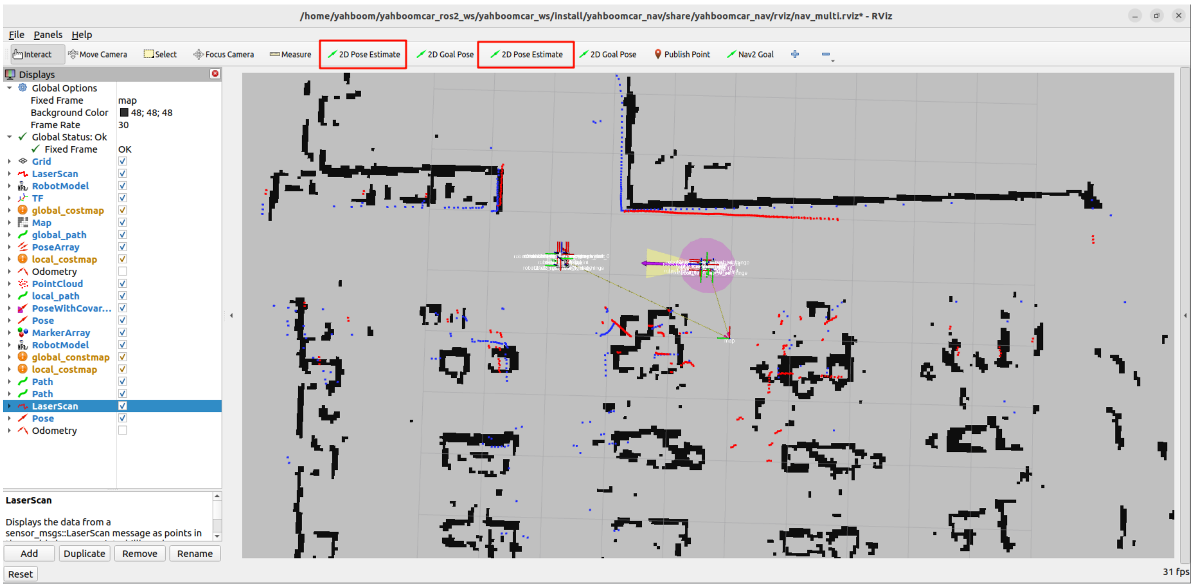
As shown in the figure below, the error message "**AMCL cannot publish a pose or update the transform. Please set the initial "pose...**" indicates that the program is running the amcl positioning program.



```
[amcl-1] [INFO] [1755170202.759349309] [robot2_amcl]: createLaserObject
[amcl-1] [WARN] [1755170203.159942293] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
[amcl-1] [WARN] [1755170205.162500071] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
[amcl-1] [WARN] [1755170207.251405684] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
[amcl-1] [WARN] [1755170209.254228526] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
[amcl-1] [WARN] [1755170211.257198716] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
[amcl-1] [WARN] [1755170213.346115961] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
[amcl-1] [WARN] [1755170215.348805883] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
[amcl-1] [WARN] [1755170217.445682821] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
[amcl-1] [WARN] [1755170219.448350245] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
[amcl-1] [WARN] [1755170221.545469046] [robot2_amcl]: AMCL cannot publish a pose or update the transform. Please set the initial pose...
```

Next, based on the car's position on the actual map, we use the [2D Pose Estimate] tool in rviz2 to give the car an initial pose. Here, rviz has two [2D Pose Estimate], the first tool is used to set the initial pose of robot 1, and the second tool is used to set the initial pose of robot 2. Use these two tools to set the initial poses for both robots.

As shown in the figure below, the areas scanned by the two radars overlap with the black area on the map. The blue point cloud represents the radar scan of robot 2, and the red point cloud represents the radar scan of robot 1.



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 ☑
PoseWithCovar... ☑
Pose ☑
MarkerArray ☑
RobotModel ☑
global_constmap ☑
local_costmap ☑
Path ☑
Path ☑
LaserScan ☑
Pose ☑
Odometry □

LaserScan

Displays the data from a sensor_msgs::LaserScan message as points in

2.4. Start nav2 Navigation

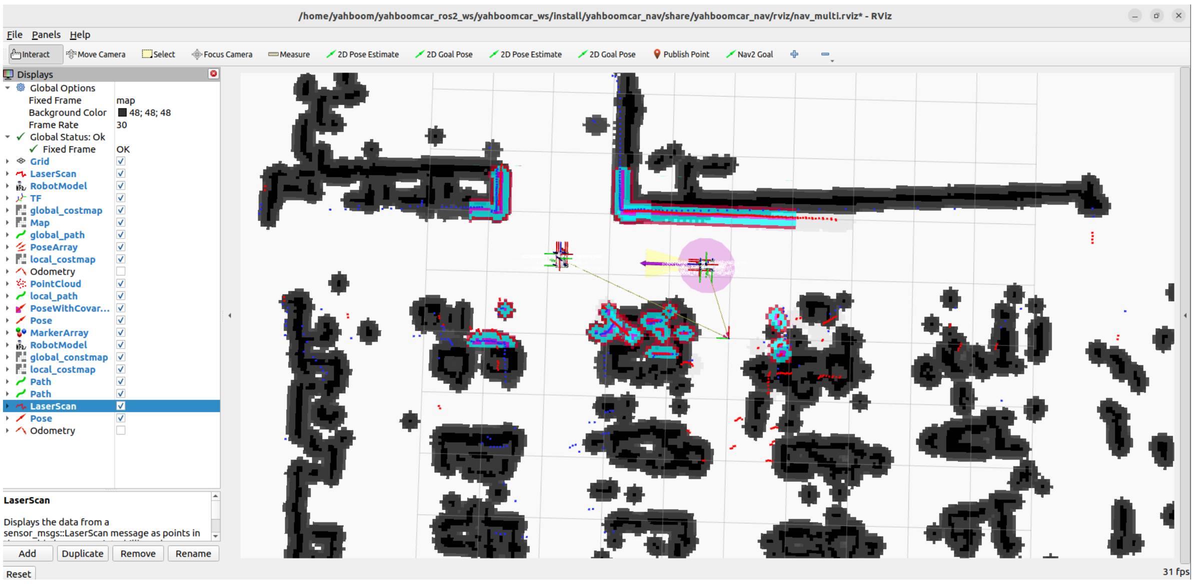
On robot 1, open the terminal and enter the following command to start nav2 navigation for robot 1.

```
ros2 launch yahboomcar_multi A1_nav_robot1_launch.py
```

On robot 2, open the terminal and enter the following command to start nav2 navigation for robot 2.

```
ros2 launch yahboomcar_multi A1_nav_robot2_launch.py
```

If both launch successfully, the expansion area shown in the image below will appear.

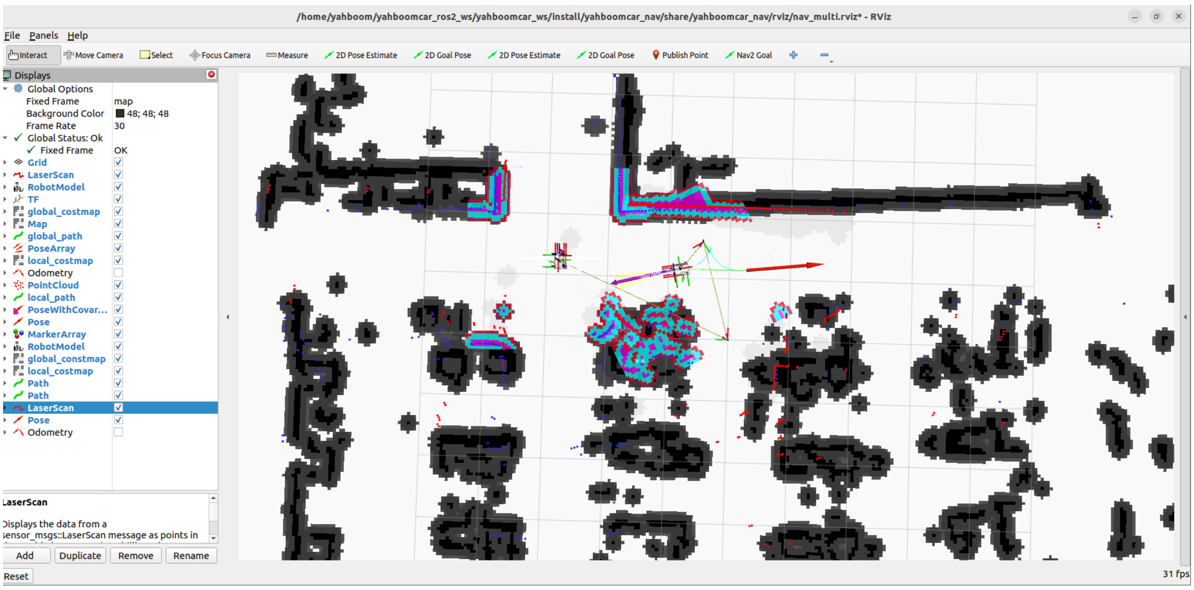


Fixed Frame: map
Background Color: 48; 48; 48
Frame Rate: 30
Global Status: Ok
Fixed Frame: OK

LaserScan

Displays the data from a sensor_msgs::LaserScan message as points in

You can use the [2D Goal Pose] tool as shown below to navigate car number one.

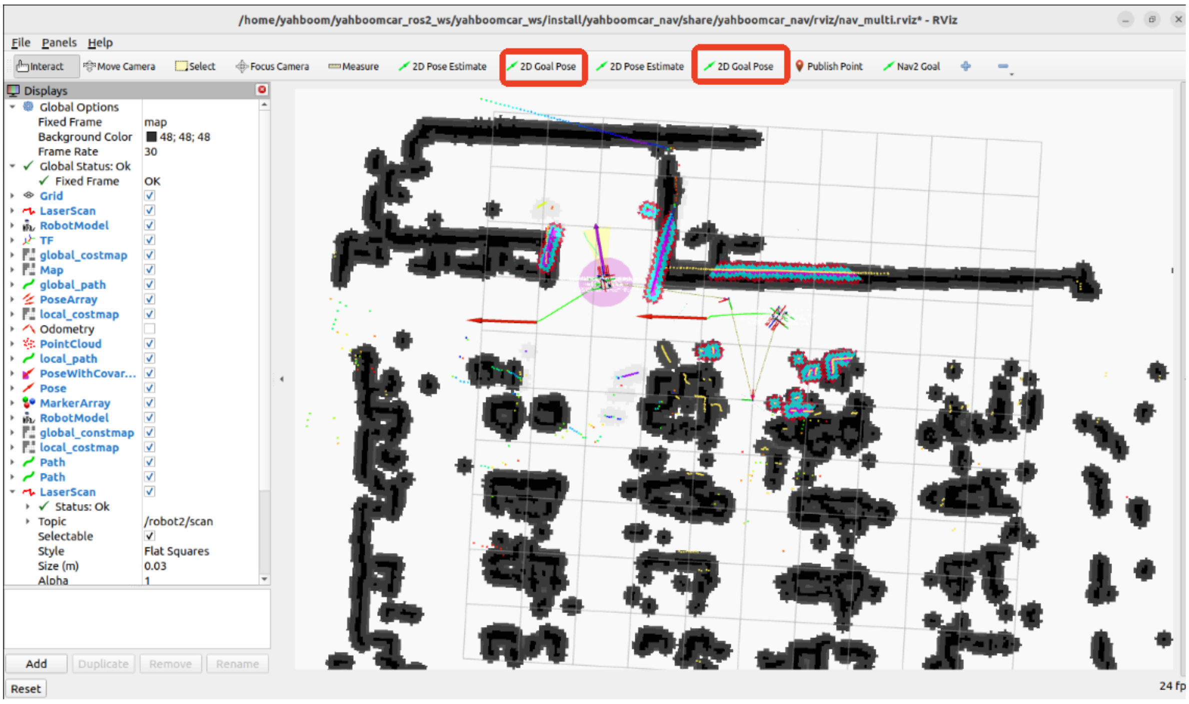


Fixed Frame: map
Background Color: 48; 48; 48
Frame Rate: 30
Global Status: Ok
Fixed Frame: OK

LaserScan

Displays the data from a sensor_msgs::LaserScan message as points in

Use the [2D Goal Pose] tool to assign a target point to the car. Here, there are two [2D Goal Pose] tools in the rviz file. Pose], the first tool is for assigning a target point to robot 1, and the second tool is for assigning a target point to robot 2. Use these two tools to assign target points to both robots, as shown in the figure below. After assigning a target point, the robots will plan a path and navigate to it. The figure below shows how both robots are navigated simultaneously.



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	☑
PoseWithCovar...	☑
Pose	☑
MarkerArray	☑
RobotModel	☑
global_constmap	☑
local_costmap	☑
Path	☑
Path	☑
LaserScan	☑
✓ Status: Ok	
Topic	/robot2/scan
Selectable	☑
Style	Flat Squares
Size (m)	0.03
Alpha	1