Planning simulation

Preparation

Download and unpack a sample map.

- You can also download the map manually.

```
gdown -O ~/autoware_map/ 'https://docs.google.com/uc?
export=download&id=1499_nsbUbIeturZaDj7jhUownh5fvXHd'
unzip -d ~/autoware_map ~/autoware_map/sample-map-planning.zip
```

Note

Sample map: Copyright 2020 TIER IV, Inc.

Check if you have `~/autoware_data` folder and files in it.

```
$ cd ~/autoware_data
$ ls -C -w 30
image_projection_based_fusion
lidar_apollo_instance_segmentation
lidar_centerpoint
tensorrt_yolo
tensorrt_yolox
traffic_light_classifier
traffic_light_fine_detector
traffic_light_ssd_fine_detector
yabloc_pose_initializer
```

If not, please, follow Manual downloading of artifacts.

Change the maximum velocity, that is 15km/h by default.

Basic simulations

Using Autoware Launch GUI

If you prefer a graphical user interface (GUI) over the command line for launching and managing your simulations, refer to the `Using Autoware Launch GUI` section at the end of this document for a step-by-step guide.

Lane driving scenario

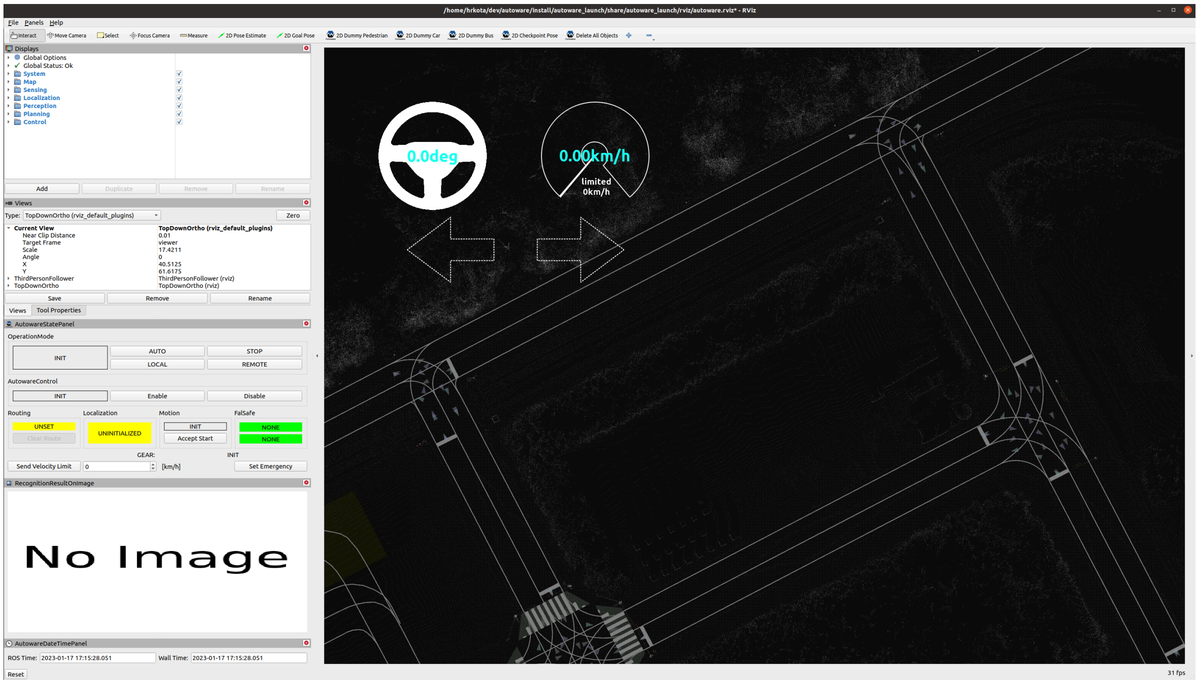
1. Launch Autoware

```
source ~/autoware/install/setup.bash
ros2 launch autoware_launch planning_simulator.launch.xml
map_path:=$HOME/autoware_map/sample-map-planning vehicle_model:=sample_vehicle
sensor_model:=sample_sensor_kit
```

Warning

Note that you cannot use `~` instead of `$HOME` here.

If `~` is used, the map will fail to load.



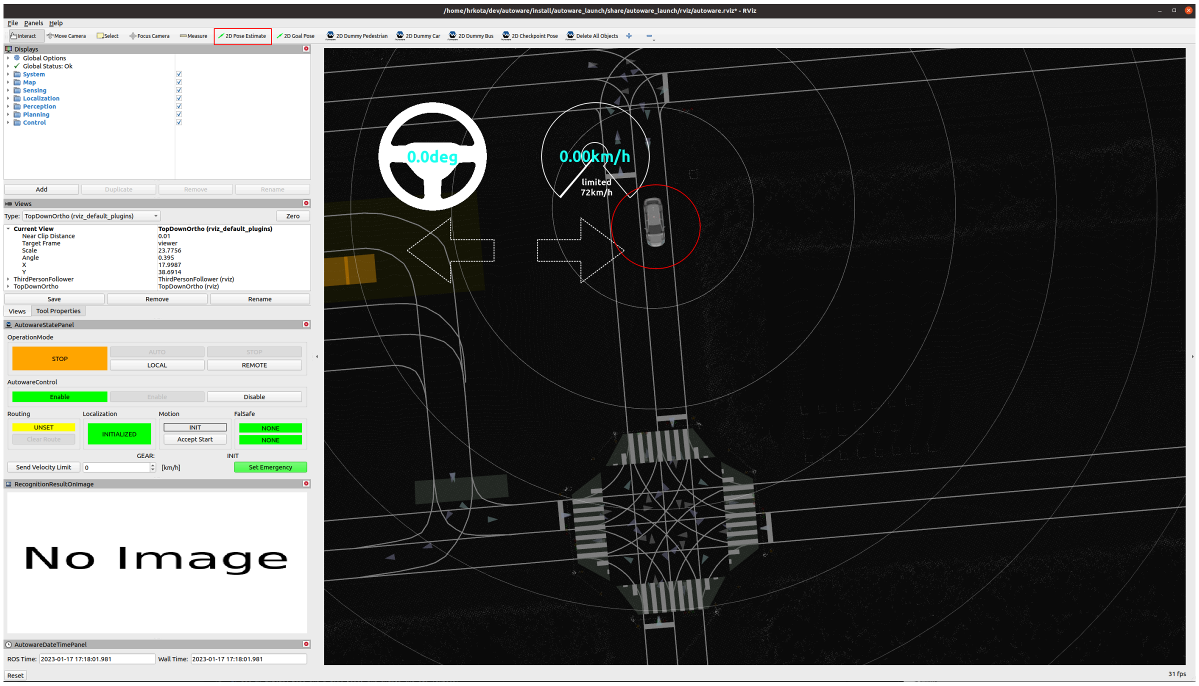
Type: TopDownOrtho (rviz_default_plugins)

Current View	TopDownOrtho (rviz_default_plugins)
Near Clip Distance	0.01
Target Frame	viewer
Scale	17.4211
Angle	0
X	40.5125
Y	61.6175
ThirdPersonFollower	ThirdPersonFollower (rviz)
TopDownOrtho	TopDownOrtho (rviz)

Send Velocity Limit: 0 [km/h]

ROS Time: 2023-01-17 17:15:28.051 Wall Time: 2023-01-17 17:15:28.051

2. Set an initial pose for the ego vehicle



Type: TopDownOrtho (rviz_default_plugins)

Current View: TopDownOrtho (rviz_default_plugins)
Near Clip Distance: 0.01
Target Frame: viewer
Scale: 23.7756
Angle: 0.395
X: 17.9987
Y: 38.6914
ThirdPersonFollower: ThirdPersonFollower (rviz)
TopDownOrtho: TopDownOrtho (rviz)

Send Velocity Limit: 0 [km/h]

ROS Time: 2023-01-17 17:18:01.981 Wall Time: 2023-01-17 17:18:01.981

a) Click the `2D Pose estimate` button in the toolbar, or hit the `P` key.

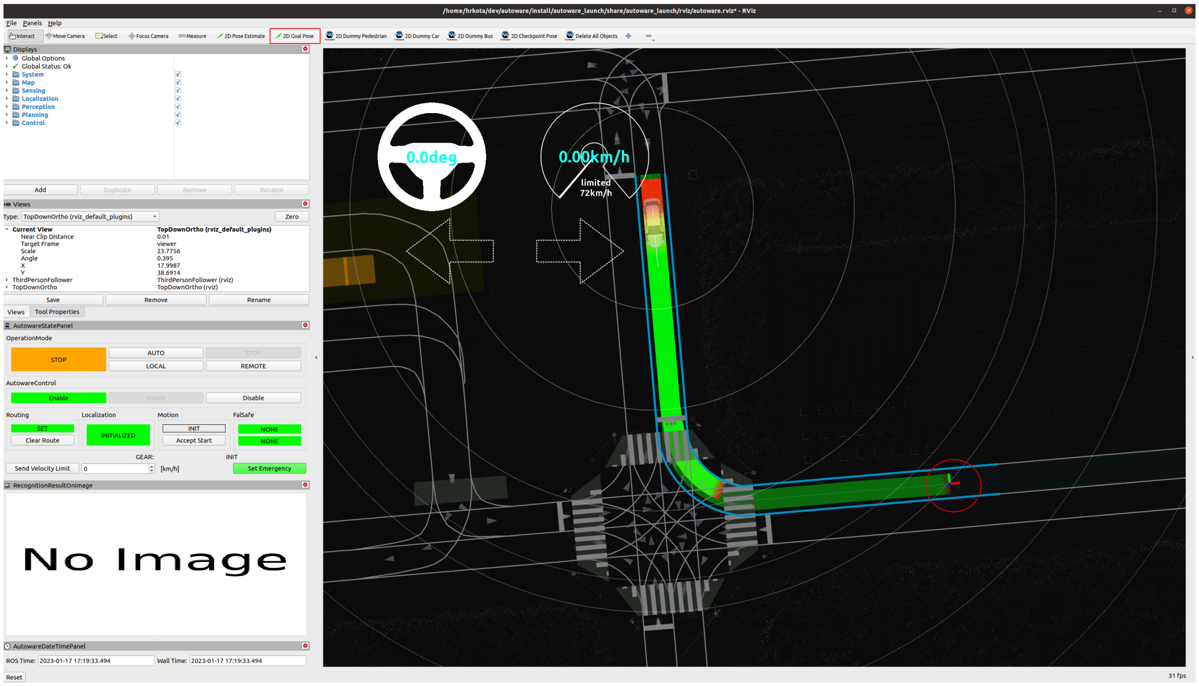
b) In the 3D View pane, click and hold the left-mouse button, and then drag to set the direction for the initial pose. An image representing the vehicle should now be displayed.

> **Warning**
>
> Remember to set the initial pose of the car in the same direction as the lane.
>
> To confirm the direction of the lane, check the arrowheads displayed on the map.

3. Set a goal pose for the ego vehicle

a) Click the `2D Goal Pose` button in the toolbar, or hit the `G` key.

b) In the 3D View pane, click and hold the left-mouse button, and then drag to set the direction for the goal pose. If done correctly, you will see a planned path from initial pose to goal pose.

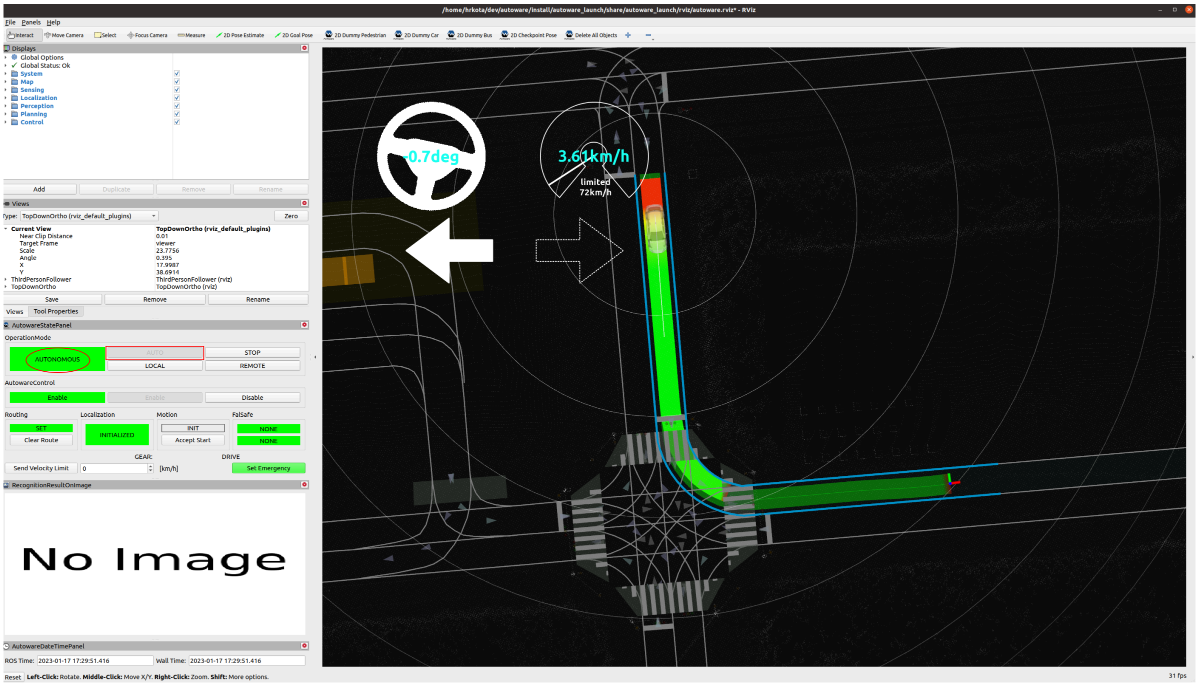


4. Start the ego vehicle

Now you can start the ego vehicle driving by clicking the `AUTO` button on `OperationMode` in `AutowareStatePanel`. Alteratively, you can manually start the vehicle by running the following command:

```
source ~/autoware/install/setup.bash
ros2 service call /api/operation_mode/change_to_autonomous
autoware_adapi_v1_msgs/srv/ChangeOperationMode {}
```

After that, you can see `AUTONOMOUS` sign on `OperationMode` and `AUTO` button is grayed out.


/home/hrkota/dev/autoware/install/autoware_launch/share/autoware_launch/rviz/autoware.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Goal Pose 2D Dummy Pedestrian 2D Dummy Car 2D Dummy Bus 2D Checkpoint Pose Delete All Objects
Displays
Global Options
Global Status: Ok
System
Map
Sensing
Localization
Perception
Planning
Control
Add Duplicate Remove Rename
Views
Zero
Save Remove Rename
Views Tool Properties
AutowareStatePanel
AUTO STOP LOCAL REMOTE
Disable
Clear Route
INIT
Accept Start
Send Velocity Limit
Set Emergency
RecognitionResultOnImage
No Image
AutowareDateTimePanel
Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Zoom. Shift: More options.
31 fps
0.7deg
3.61km/h
limited 72km/h

Type: TopDownOrtho (rviz_default_plugins)

Current View	TopDownOrtho (rviz_default_plugins)
Near Clip Distance	0.01
Target Frame	viewer
Scale	23.7756
Angle	0.395
X	17.9987
Y	38.6914
ThirdPersonFollower	ThirdPersonFollower (rviz)
TopDownOrtho	TopDownOrtho (rviz)

OperationMode: AUTONOMOUS

AutowareControl: Enable

Routing: SET

Localization: INITIALIZED

FalSafe: NONE NONE

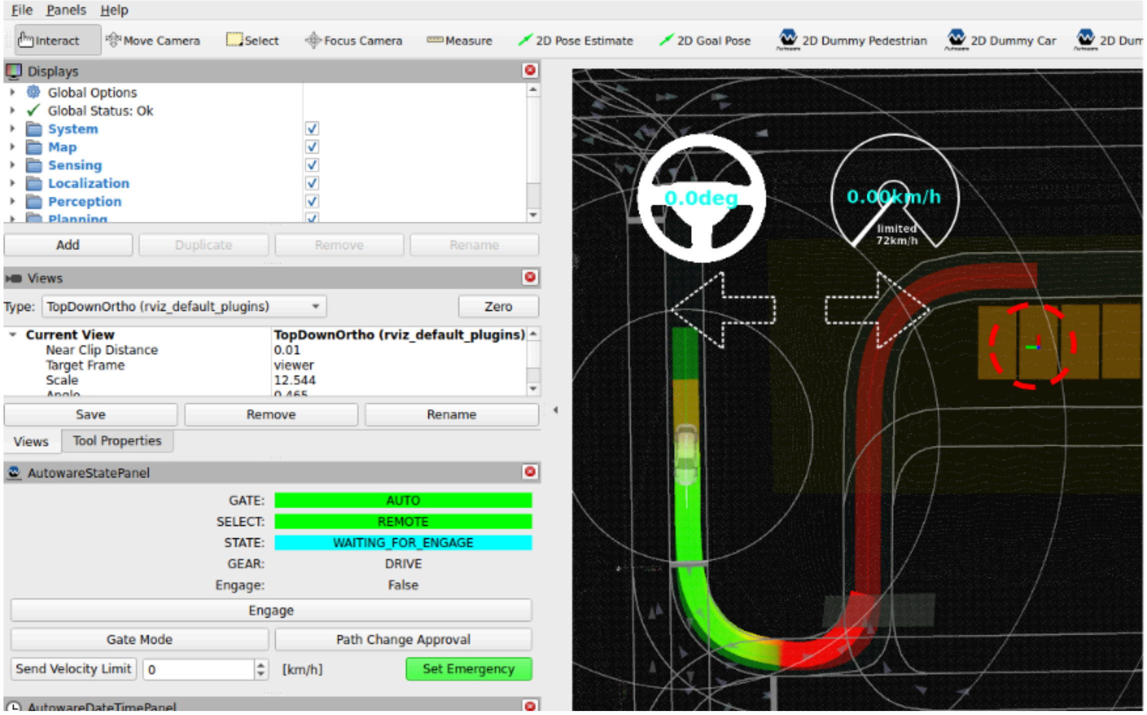
GEAR: DRIVE

Send Velocity Limit 0 [km/h]

ROS Time: 2023-01-17 17:29:51.416 Wall Time: 2023-01-17 17:29:51.416

Parking scenario

1. Set an initial pose and a goal pose, and engage the ego vehicle.



Type: TopDownOrtho (rviz_default_plugins)

Current View	TopDownOrtho (rviz_default_plugins)
Near Clip Distance	0.01
Target Frame	viewer
Scale	12.544

GATE: AUTO

SELECT: REMOTE

STATE: WAITING_FOR_ENGAGE

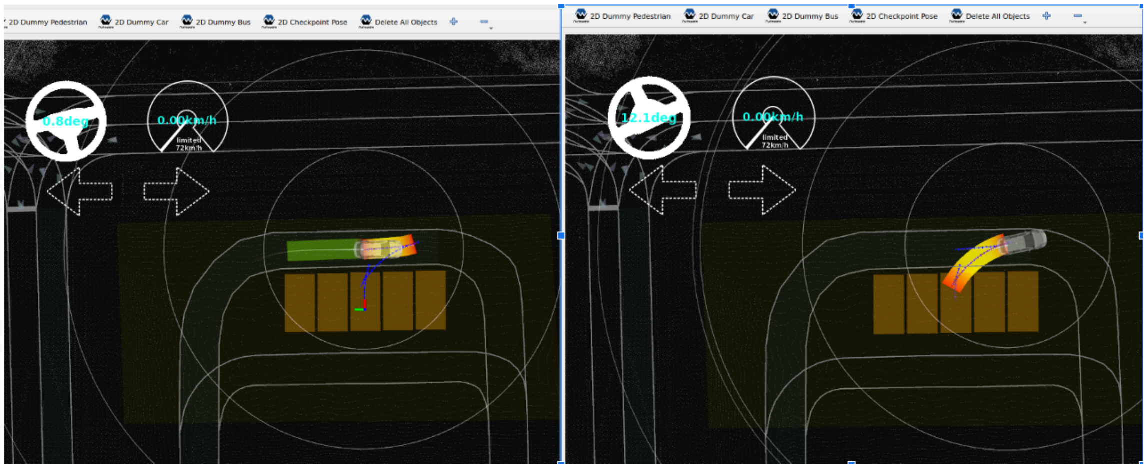
GEAR: DRIVE

Engage: False

Send Velocity Limit 0 [km/h]

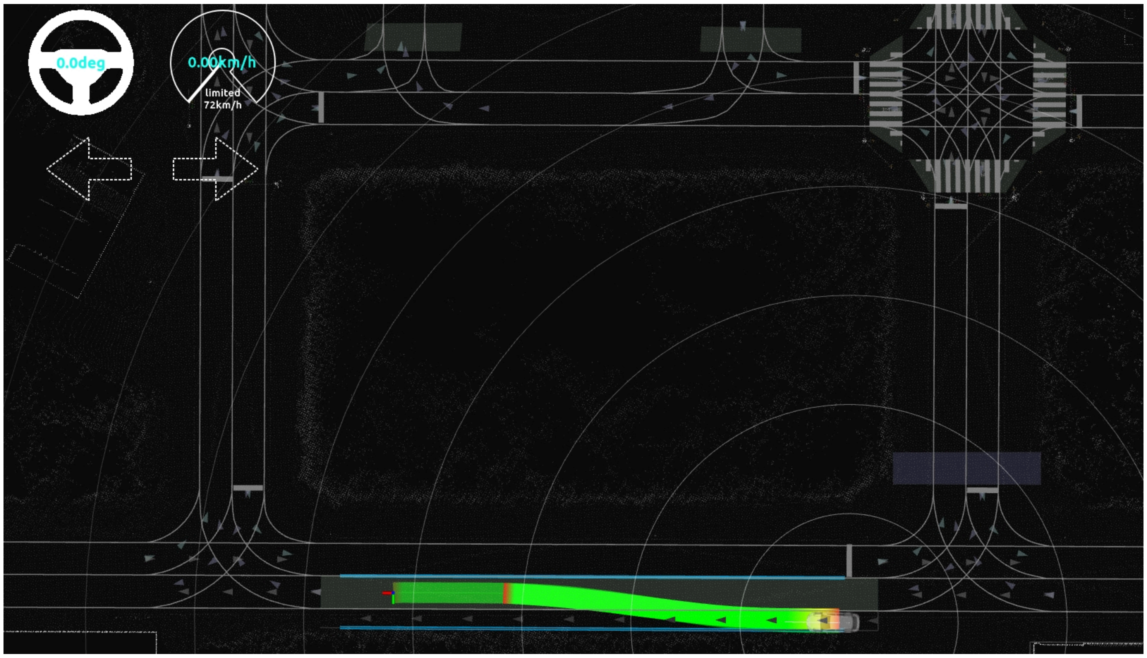
2. When the vehicle approaches the goal, it will switch from lane driving mode to parking mode.

3. After that, the vehicle will reverse into the destination parking spot.



Pull out and pull over scenario

1. In a pull out scenario, set the ego vehicle at the road shoulder.



2. Set a goal and then engage the ego vehicle.
3. In a pull over scenario, similarly set the ego vehicle in a lane and set a goal on the road shoulder.

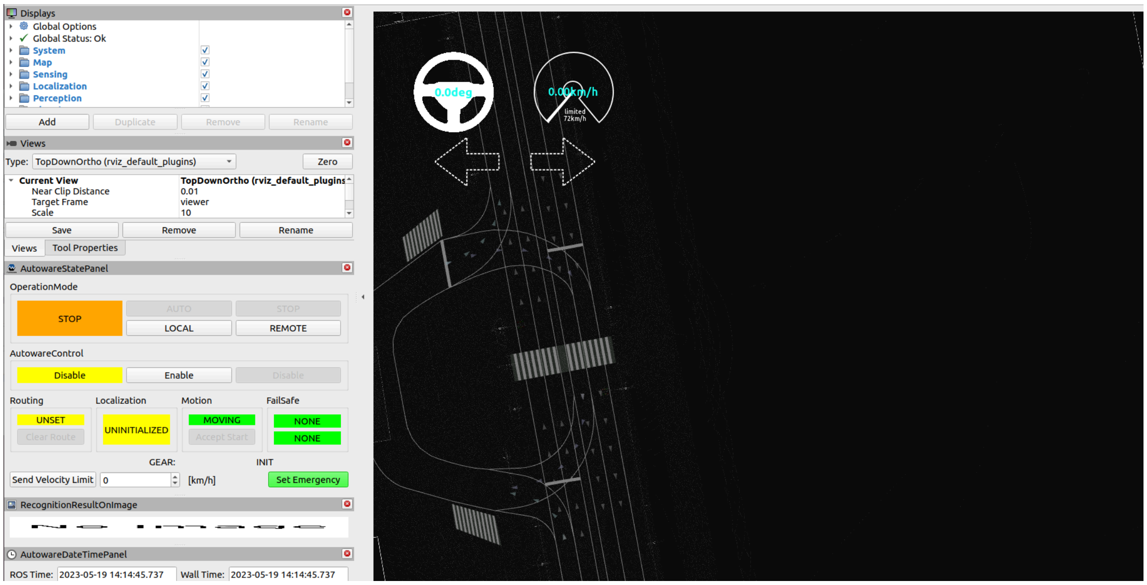
Lane change scenario

1. Download and unpack Nishishinjuku map.

```
gdown -O ~/autoware_map/
'https://github.com/tier4/AWSIM/releases/download/v1.1.0/nishishinjuku_autoware_map.zip'
unzip -d ~/autoware_map ~/autoware_map/nishishinjuku_autoware_map.zip
```

2. Launch autoware with Nishishinjuku map with following command:

```
source ~/autoware/install/setup.bash
ros2 launch autoware_launch planning_simulator.launch.xml
map_path:=$HOME/autoware_map/nishishinjuku_autoware_map
vehicle_model:=sample_vehicle sensor_model:=sample_sensor_kit
```



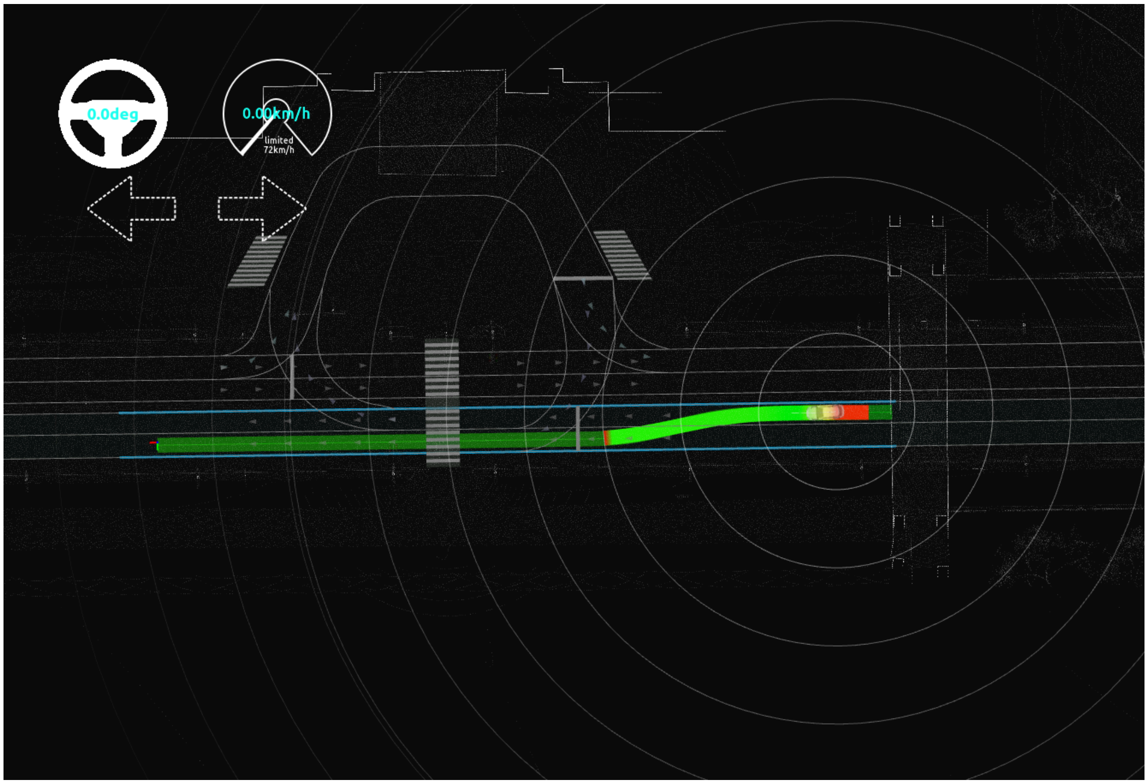
Type: TopDownOrtho (rviz_default_plugins)

Current View	TopDownOrtho (rviz_default_plugins
Near Clip Distance	0.01
Target Frame	viewer
Scale	10

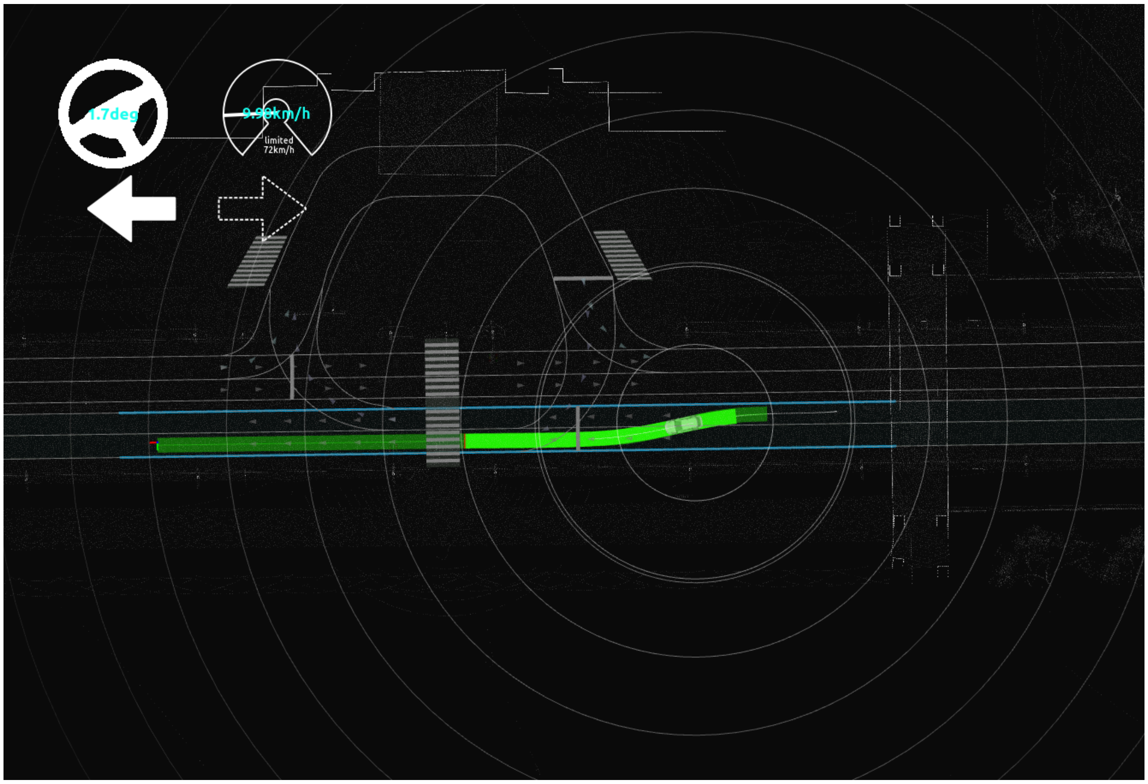
Send Velocity Limit 0 [km/h]

ROS Time: 2023-05-19 14:14:45.737 Wall Time: 2023-05-19 14:14:45.737

3. Set an initial pose and a goal pose in adjacent lanes.

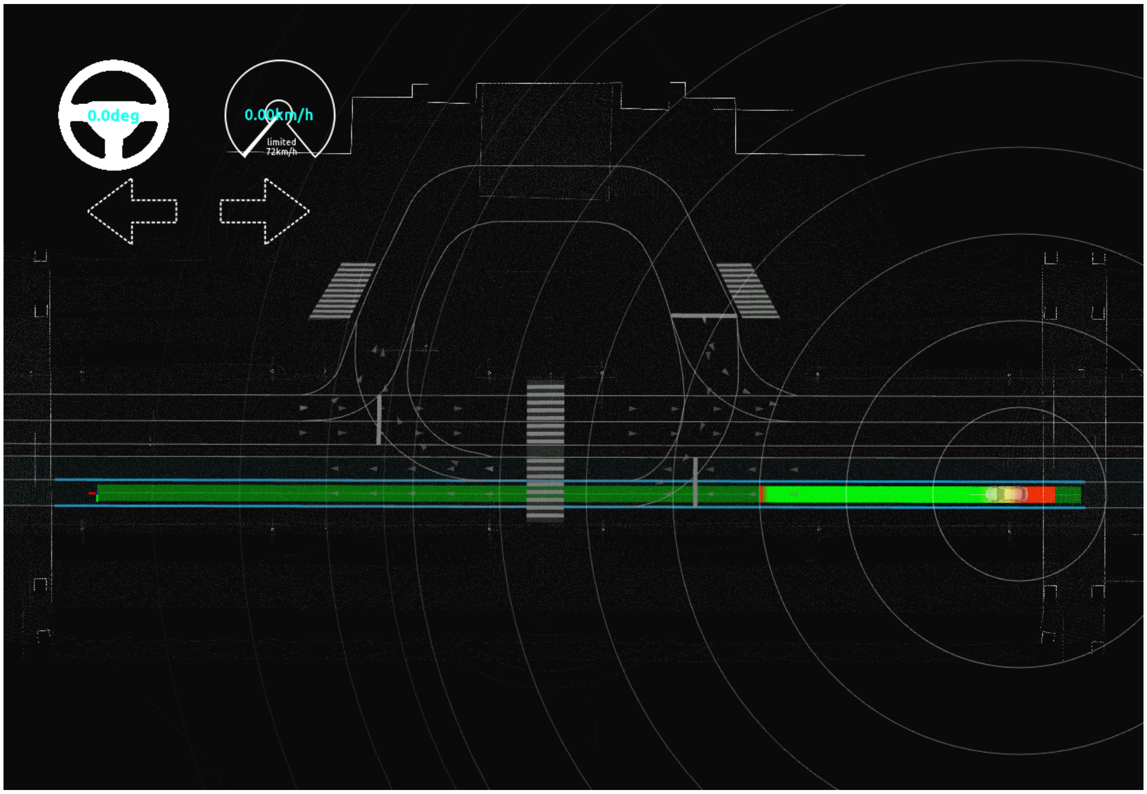


4. Engage the ego vehicle. It will make a lane change along the planned path.

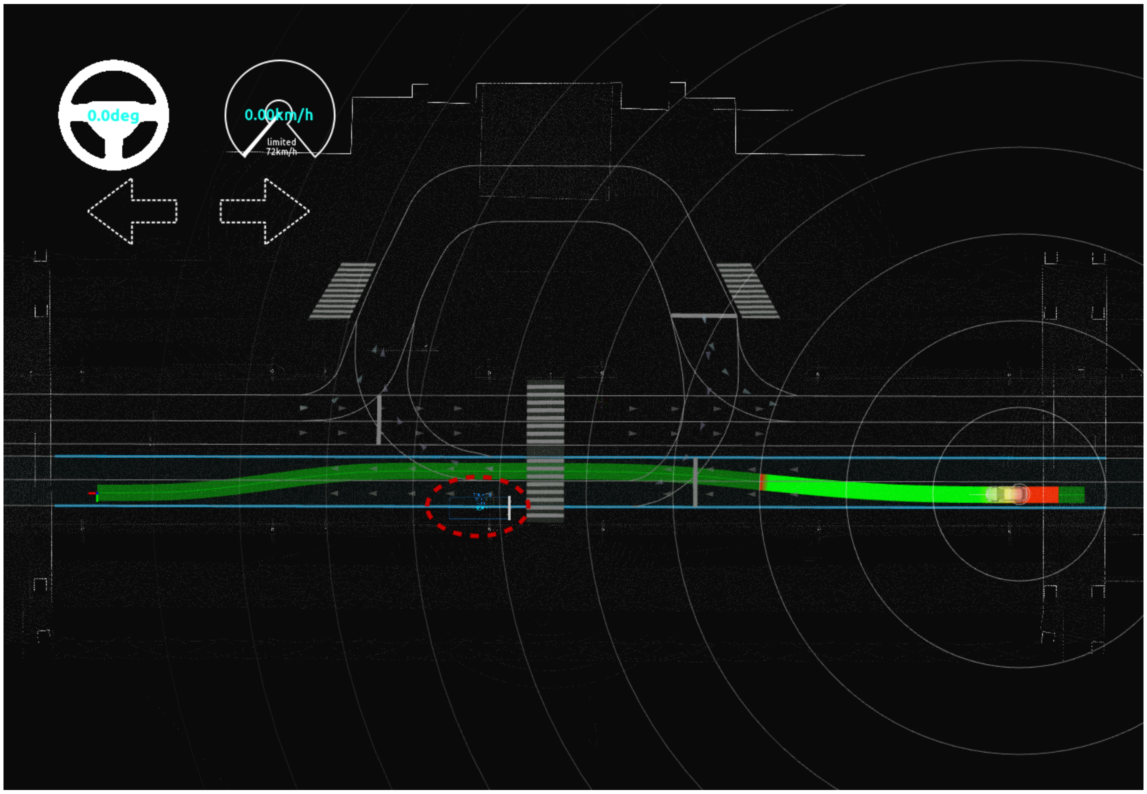


Avoidance scenario

1. Set an initial pose and a goal pose in the same lane. A path will be planned.



2. Set a "2D Dummy Bus" on the roadside. A new path will be planned.



3. Engage the ego vehicle. It will avoid the obstacle along the newly planned path.

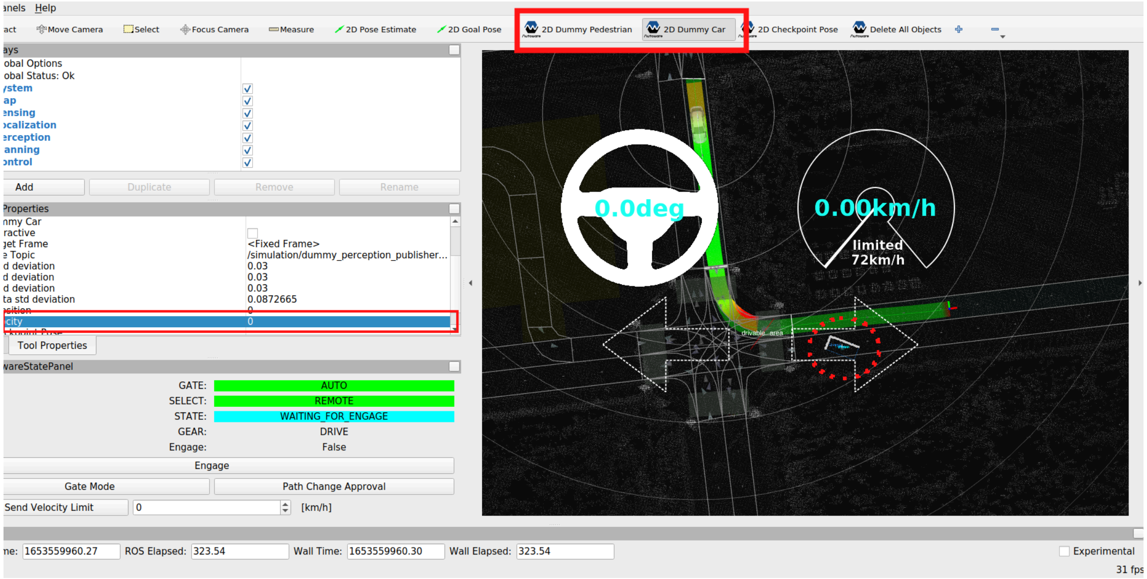
Advanced Simulations

Placing dummy objects

1. Click the `2D Dummy Car` or `2D Dummy Pedestrian` button in the toolbar.
2. Set the pose of the dummy object by clicking and dragging on the map.
3. Set the velocity of the object in `Tool Properties -> 2D Dummy Car/Pedestrian` panel.

   !!! note

   Changes to the `velocity` parameter will only affect objects placed after the parameter is changed.



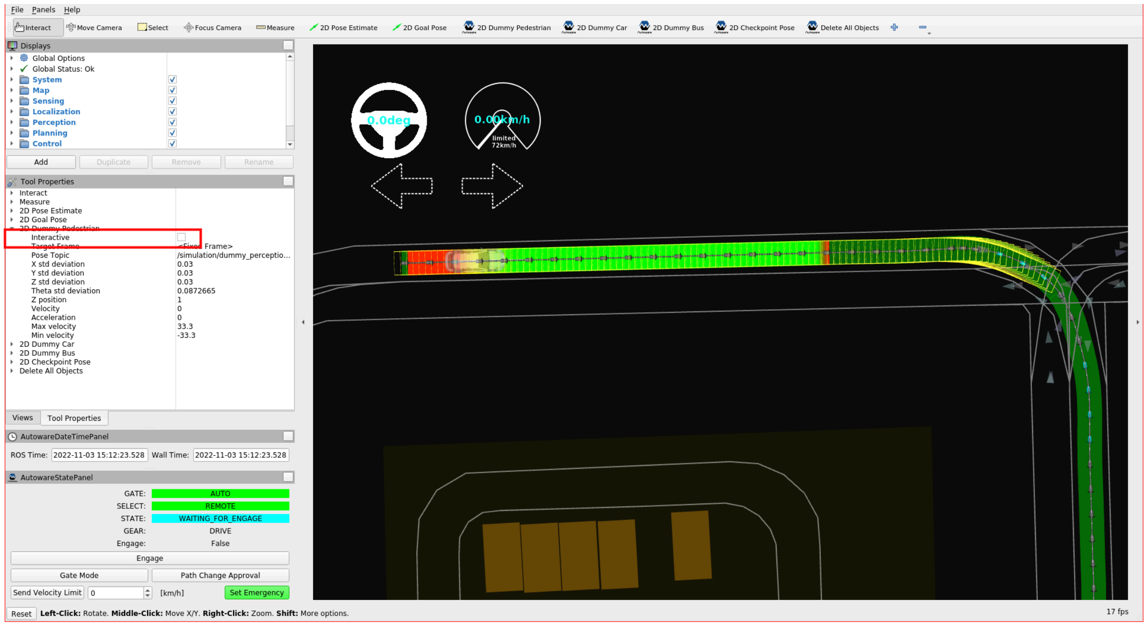
nmy Car	
ractive	□
jet Frame	<Fixed Frame>
e Topic	/simulation/dummy_perception_publisher...
d deviation	0.03
d deviation	0.03
d deviation	0.03
ta std deviation	0.0872665
sition	0
city	0

GATE: AUTO
SELECT: REMOTE
STATE: WAITING_FOR_ENGAGE
GEAR: DRIVE
Engage: False

Send Velocity Limit 0 [km/h]

ne: 1653559960.27 ROS Elapsed: 323.54 Wall Time: 1653559960.30 Wall Elapsed: 323.54

4. Delete any dummy objects placed in the view by clicking the `Delete All Objects` button in the toolbar.
5. Click the `Interactive` button in the toolbar to make the dummy object interactive.



Interactive	□
Target Frame	<Fixed Frame>
Pose Topic	/simulation/dummy_perceptio...
X std deviation	0.03
Y std deviation	0.03
Z std deviation	0.03
Theta std deviation	0.0872665
Z position	1
Velocity	0
Acceleration	0
Max velocity	33.3
Min velocity	-33.3

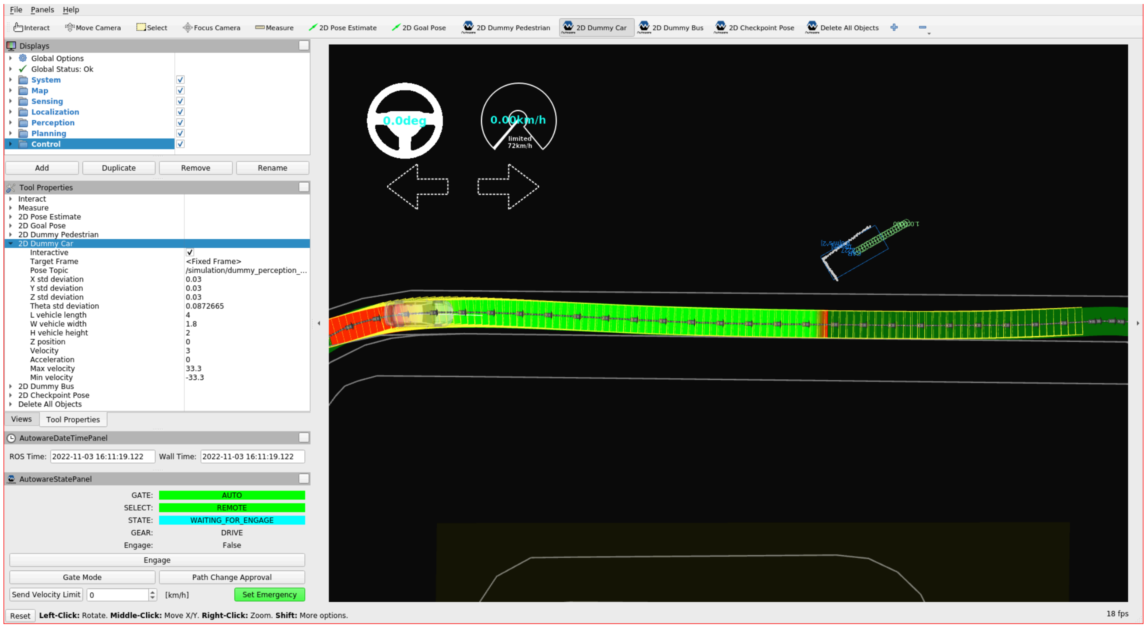
ROS Time: 2022-11-03 15:12:23.528 Wall Time: 2022-11-03 15:12:23.528

GATE: AUTO
SELECT: REMOTE
STATE: WAITING_FOR_ENGAGE
GEAR: DRIVE
Engage: False

Send Velocity Limit 0 [km/h]

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

6. For adding an interactive dummy object, press `SHIFT` and click the `right click`.
7. For deleting an interactive dummy object, press `ALT` and click the `right click`.
8. For moving an interactive dummy object, hold the `right click` drag and drop the object.



Tool Properties

- Interact
- Measure
- 2D Pose Estimate
- 2D Goal Pose
- 2D Dummy Pedestrian
- 2D Dummy Car
  - Interactive ☑
  - Target Frame <Fixed Frame>
  - Pose Topic /simulation/dummy_perception_...
  - X std deviation 0.03
  - Y std deviation 0.03
  - Z std deviation 0.03
  - Theta std deviation 0.0872665
  - L vehicle length 4
  - W vehicle width 1.8
  - H vehicle height 2
  - Z position 0
  - Velocity 3
  - Acceleration 0
  - Max velocity 33.3
  - Min velocity -33.3
- 2D Dummy Bus
- 2D Checkpoint Pose
- Delete All Objects

AutowareDateTimePanel

ROS Time: 2022-11-03 16:11:19.122 Wall Time: 2022-11-03 16:11:19.122

AutowareStatePanel

GATE: AUTO
SELECT: REMOTE
STATE: WAITING_FOR_ENGAGE
GEAR: DRIVE
Engage: False

Send Velocity Limit 0 [km/h]

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

Traffic light recognition simulation

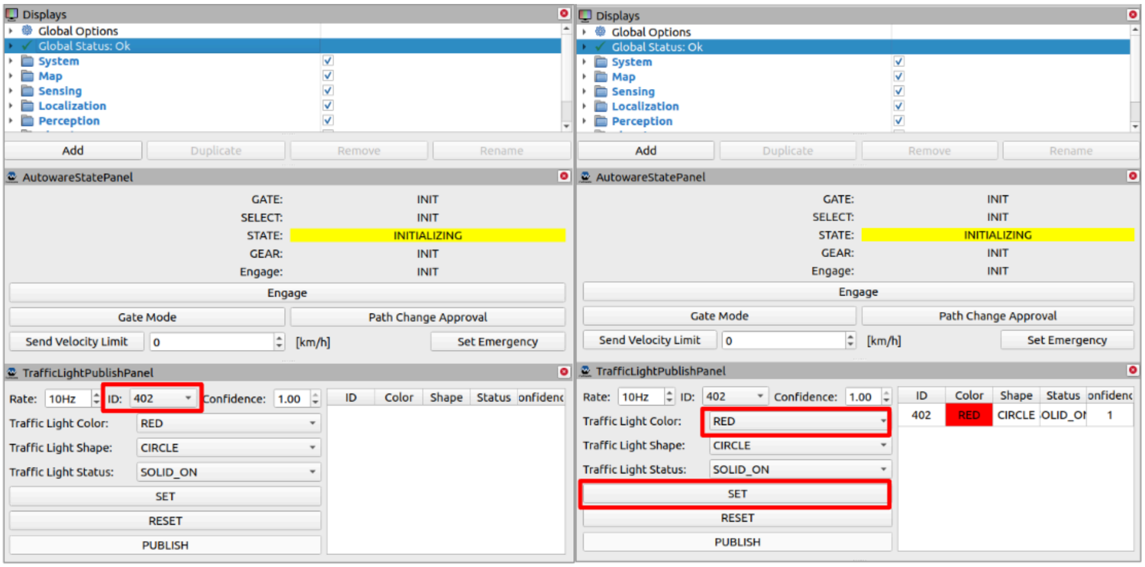
By default, traffic lights on the map are all treated as if they are set to green. As a result, when a path is created that passed through an intersection with a traffic light, the ego vehicle will drive through the intersection without stopping.

The following steps explain how to set and reset traffic lights in order to test how the Planning component will respond.

Set traffic light

1. Go to `Panels -> Add new panel`, select `TrafficLightPublishPanel`, and then press `OK`.
2. In `TrafficLightPublishPanel`, set the `ID` and color of the traffic light.

3. Click the `SET` button.



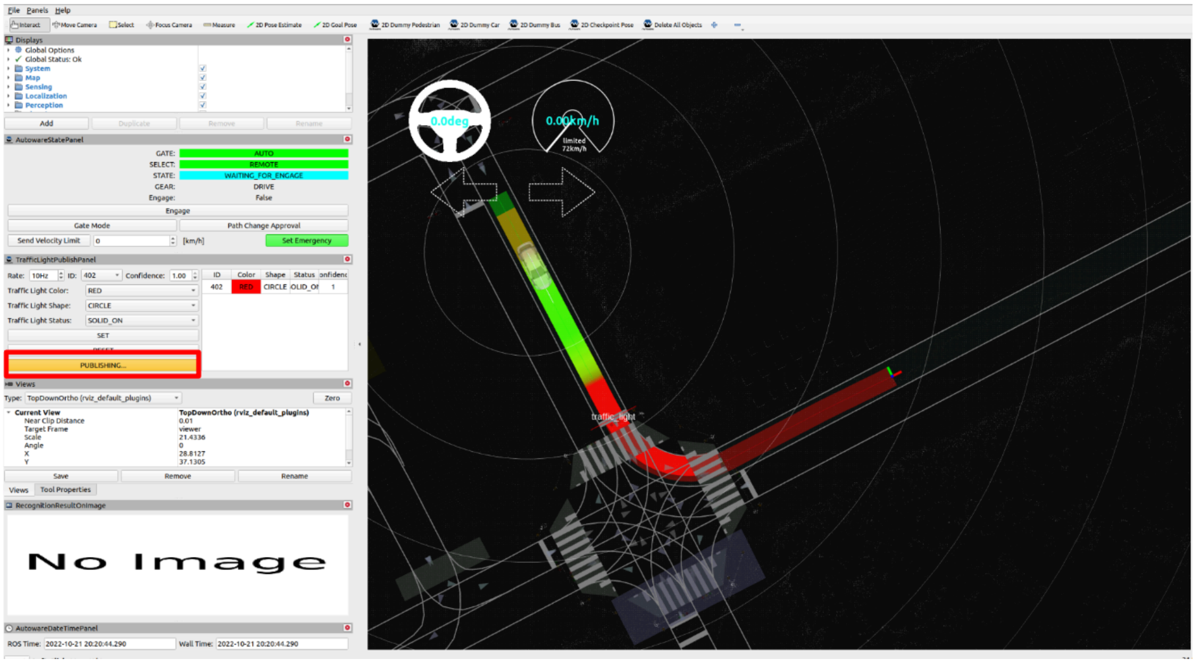
GATE: INIT
SELECT: INIT
STATE: INITIALIZING
GEAR: INIT
Engage: INIT

Send Velocity Limit: 0 [km/h]

Rate: 10Hz ID: 402 Confidence: 1.00
Traffic Light Color: RED
Traffic Light Shape: CIRCLE
Traffic Light Status: SOLID_ON

ID	Color	Shape	Status	onfidenc
402	RED	CIRCLE	OLID_OI	1

4. Finally, click the `PUBLISH` button to send the traffic light status to the simulator. Any planned path that goes past the selected traffic light will then change accordingly.



GATE: AUTO
SELECT: REMOTE
STATE: WAITING_FOR_ENGAGE
GEAR: DRIVE
Engage: False

Send Velocity Limit: 0 [km/h]

Rate: 10Hz ID: 402 Confidence: 1.00
Traffic Light Color: RED
Traffic Light Shape: CIRCLE
Traffic Light Status: SOLID_ON

ID	Color	Shape	Status	onfidenc
402	RED	CIRCLE	OLID_OI	1

Type: TopDownOrtho (rviz_default_plugins)

Current View: TopDownOrtho (rviz_default_plugins)
Near Clip Distance: 0.01
Target Frame: viewer
Scale: 21.4336
Angle: 0
X: 28.8127
Y: 37.1305

ROS Time: 2022-10-21 20:20:44.290 Wall Time: 2022-10-21 20:20:44.290

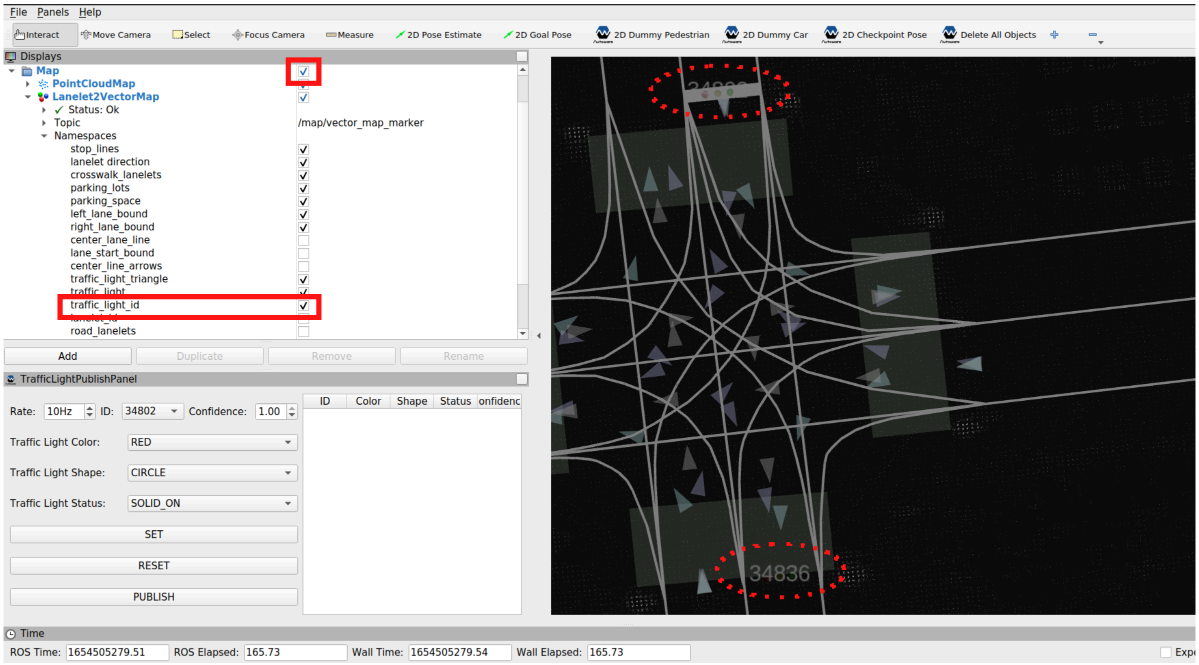
By default, Rviz should display the ID of each traffic light on the map. You can have a closer look at the IDs by zooming in the region or by changing the View type.

In case the IDs are not displayed, try the following troubleshooting steps:

a) In the `Displays` panel, find the `traffic_light_id` topic by toggling the triangle icons next to `Map > Lanelet2VectorMap > Namespaces`.

b) Check the `traffic_light_id` checkbox.

c) Reload the topic by clicking the `Map` checkbox twice.



Displays:
- Map ☑
  - PointCloudMap ☑
  - Lanelet2VectorMap ☑
    - Status: Ok
    - Topic /map/vector_map_marker
    - Namespaces
      - stop_lines ☑
      - lanelet direction ☑
      - crosswalk_lanelets ☑
      - parking_lots ☑
      - parking_space ☑
      - left_lane_bound ☑
      - right_lane_bound ☑
      - center_lane_line □
      - lane_start_bound □
      - center_line_arrows □
      - traffic_light_triangle ☑
      - traffic_light ☑
      - traffic_light_id ☑
      - lanelet_id □
      - road_lanelets □

Rate: 10Hz ID: 34802 Confidence: 1.00

Traffic Light Color: RED

Traffic Light Shape: CIRCLE

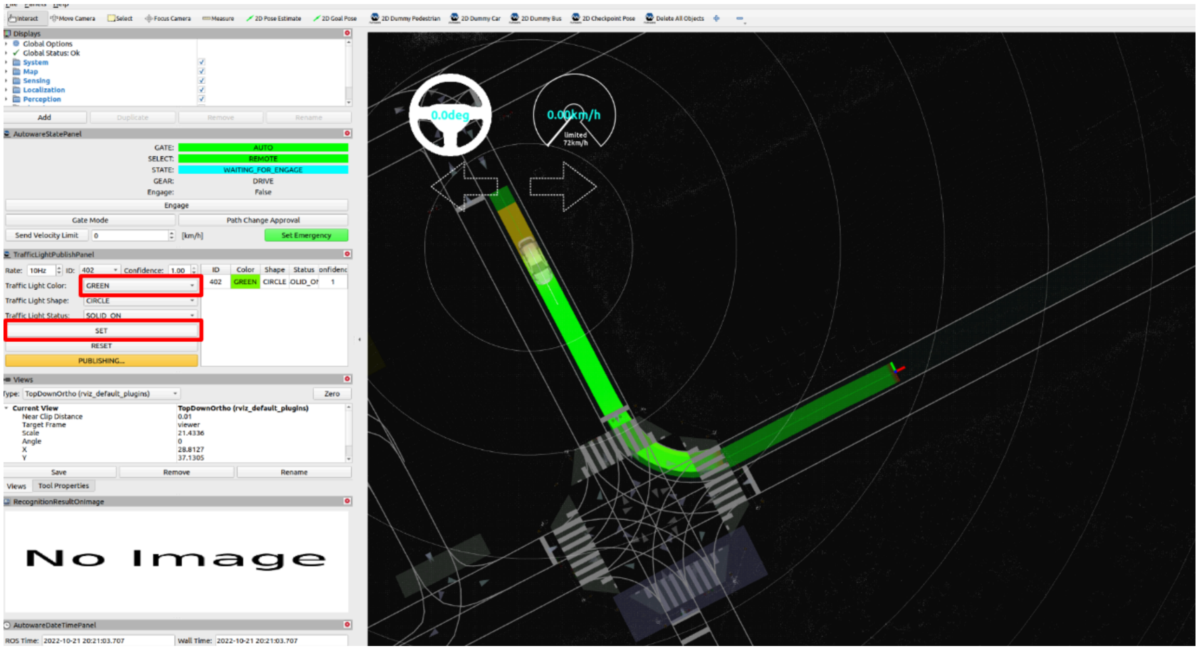
Traffic Light Status: SOLID_ON

ID	Color	Shape	Status	onfidenc

ROS Time: 1654505279.51 ROS Elapsed: 165.73 Wall Time: 1654505279.54 Wall Elapsed: 165.73

Update/Reset traffic light

You can update the color of the traffic light by selecting the next color (in the image it is `GREEN`) and clicking `SET` button. In the image the traffic light in front of the ego vehicle changed from `RED` to `GREEN` and the vehicle restarted.



Displays:
- Global Options
- Global Status: Ok
- System ☑
- Map ☑
- Sensing ☑
- Localization ☑
- Perception ☑

GATE: AUTO

SELECT: REMOTE

STATE: WAITING_FOR_ENGAGE

GEAR: DRIVE

Engage: False

Send Velocity Limit 0 [km/h]

Rate: 10Hz ID: 402 Confidence: 1.00

Traffic Light Color: GREEN

Traffic Light Shape: CIRCLE

Traffic Light Status: SOLID_ON

ID	Color	Shape	Status	onfidenc
402	GREEN	CIRCLE	OLID_ON	1

Type: TopDownOrtho (rviz_default_plugins)

Current View	TopDownOrtho (rviz_default_plugins)
Near Clip Distance	0.01
Target Frame	viewer
Scale	21.4336
Angle	0
X	28.8127
Y	37.1305

ROS Time: 2022-10-21 20:21:03.707 Wall Time: 2022-10-21 20:21:03.707

To remove a traffic light from `TrafficLightPublishPanel`, click the `RESET` button.

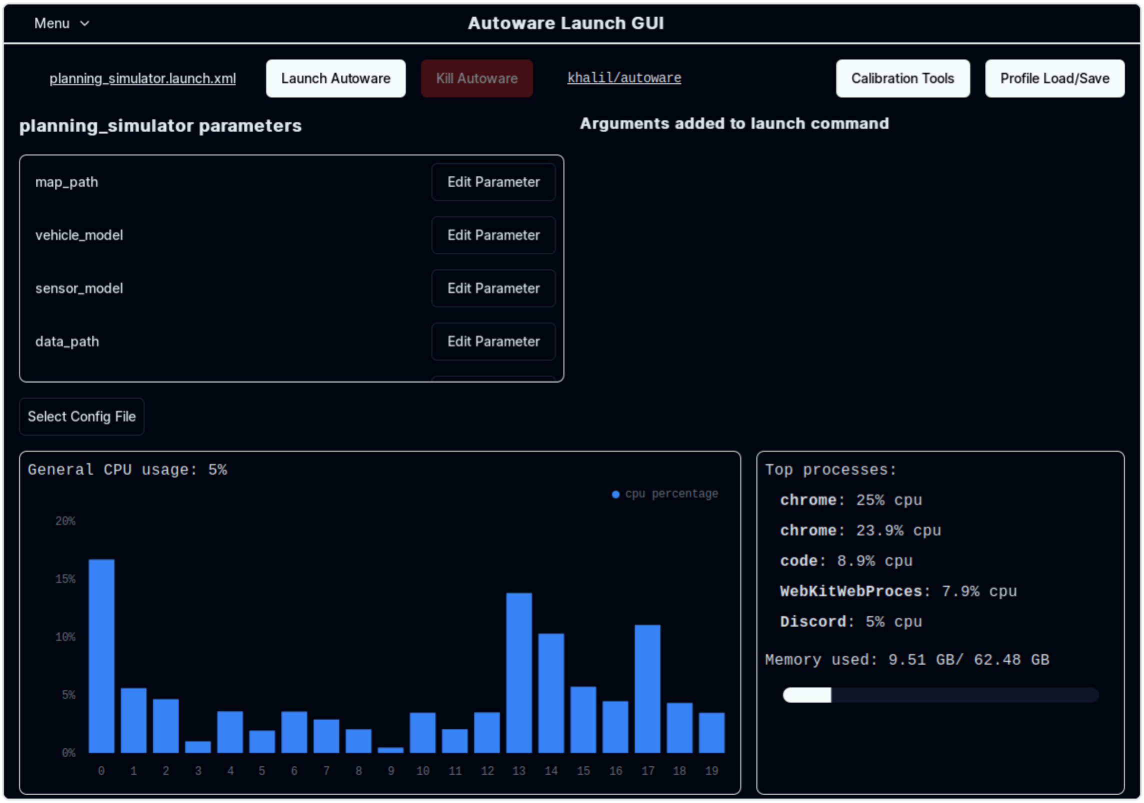
Reference video tutorials

Using Autoware Launch GUI

This section provides a step-by-step guide on using the Autoware Launch GUI for planning simulations, offering an alternative to the command-line instructions provided in the Basic simulations section.

Getting Started with Autoware Launch GUI

1. **Installation:** Ensure you have installed the Autoware Launch GUI. Installation instructions.
2. **Launching the GUI:** Open the Autoware Launch GUI from your applications menu.



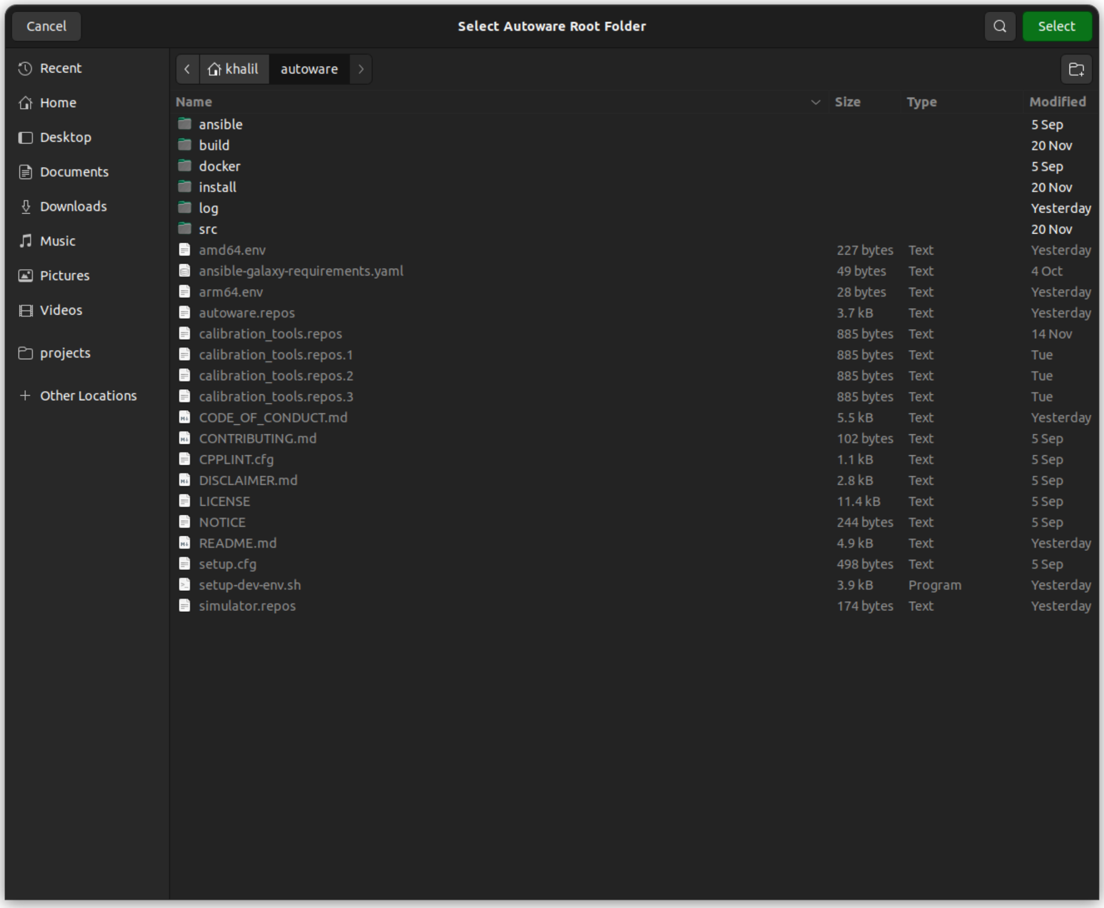
```
General CPU usage: 5%
```

```
Top processes:
  chrome: 25% cpu
  chrome: 23.9% cpu
  code: 8.9% cpu
  WebKitWebProces: 7.9% cpu
  Discord: 5% cpu
Memory used: 9.51 GB/ 62.48 GB
```

Launching a Planning Simulation

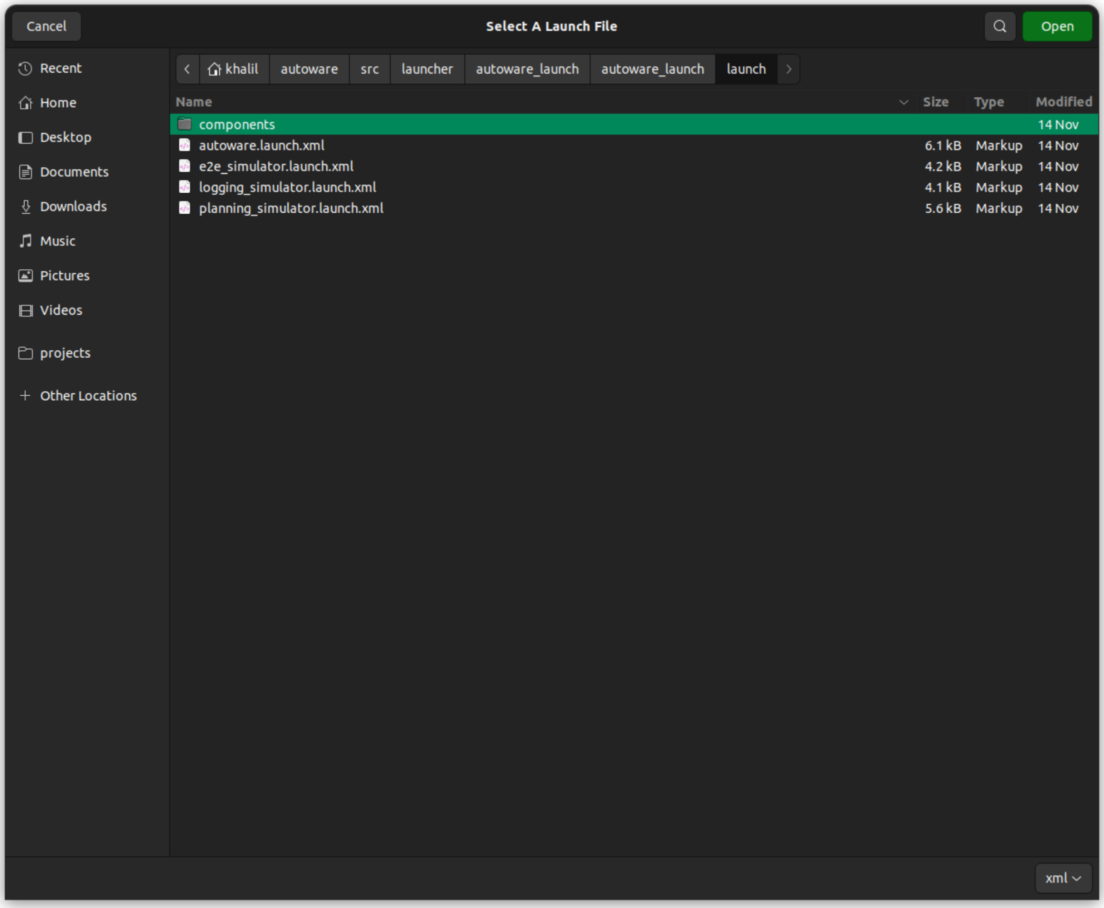
Lane Driving Scenario

1. **Set Autoware Path:** In the GUI, set the path to your Autoware installation.



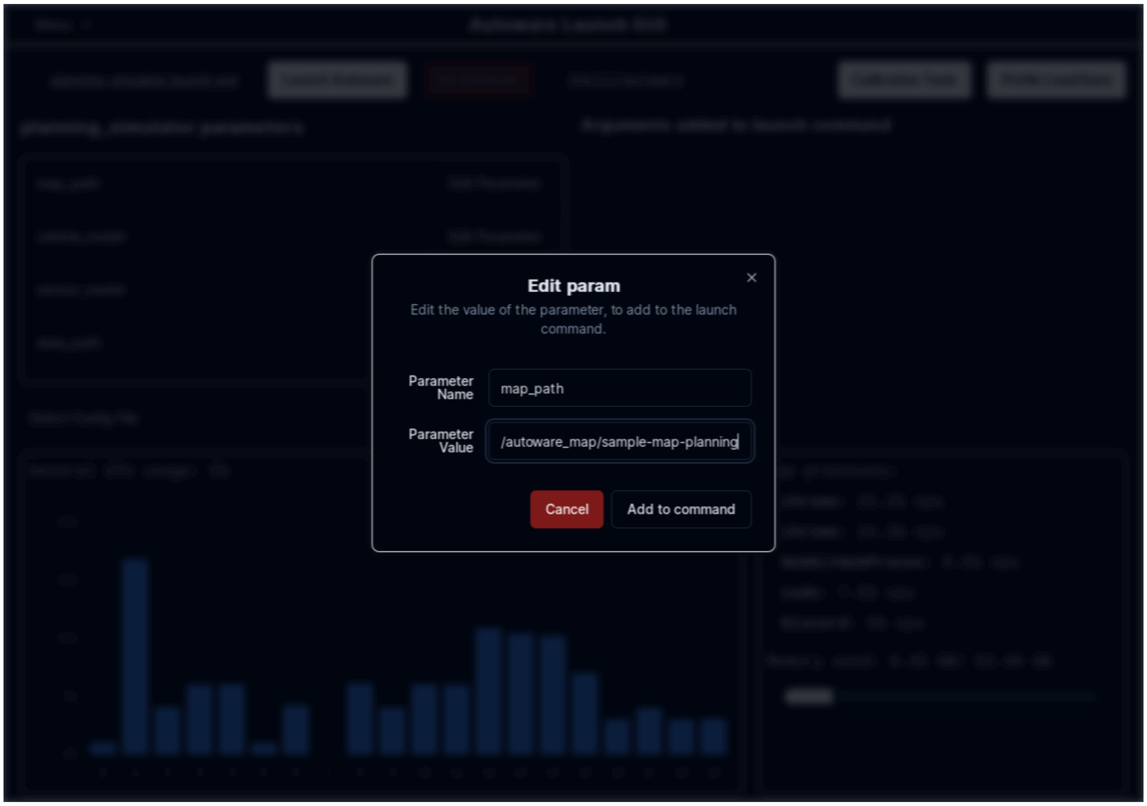
Name	Size	Type	Modified
ansible			5 Sep
build			20 Nov
docker			5 Sep
install			20 Nov
log			Yesterday
src			20 Nov
amd64.env	227 bytes	Text	Yesterday
ansible-galaxy-requirements.yaml	49 bytes	Text	4 Oct
arm64.env	28 bytes	Text	Yesterday
autoware.repos	3.7 kB	Text	Yesterday
calibration_tools.repos	885 bytes	Text	14 Nov
calibration_tools.repos.1	885 bytes	Text	Tue
calibration_tools.repos.2	885 bytes	Text	Tue
calibration_tools.repos.3	885 bytes	Text	Tue
CODE_OF_CONDUCT.md	5.5 kB	Text	Yesterday
CONTRIBUTING.md	102 bytes	Text	5 Sep
CPPLINT.cfg	1.1 kB	Text	5 Sep
DISCLAIMER.md	2.8 kB	Text	5 Sep
LICENSE	11.4 kB	Text	5 Sep
NOTICE	244 bytes	Text	5 Sep
README.md	4.9 kB	Text	Yesterday
setup.cfg	498 bytes	Text	5 Sep
setup-dev-env.sh	3.9 kB	Program	Yesterday
simulator.repos	174 bytes	Text	Yesterday

2. **Select Launch File:** Choose `planning_simulator.launch.xml` for the lane driving scenario.



Name	Size	Type	Modified
components			14 Nov
autoware.launch.xml	6.1 kB	Markup	14 Nov
e2e_simulator.launch.xml	4.2 kB	Markup	14 Nov
logging_simulator.launch.xml	4.1 kB	Markup	14 Nov
planning_simulator.launch.xml	5.6 kB	Markup	14 Nov

3. **Customize Parameters:** Adjust parameters such as `map_path`, `vehicle_model`, and `sensor_model` as needed.

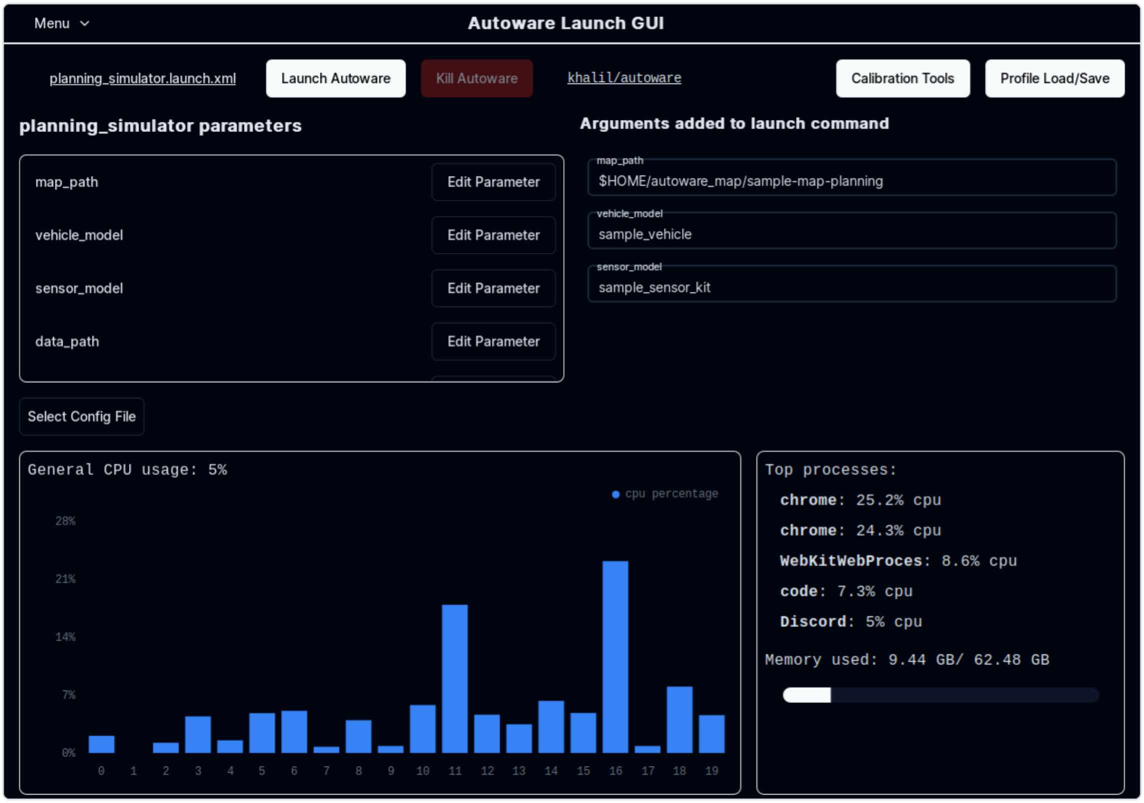


Edit param

Edit the value of the parameter, to add to the launch command.

Parameter Name: map_path

Parameter Value: /autoware_map/sample-map-planning



map_path: $HOME/autoware_map/sample-map-planning

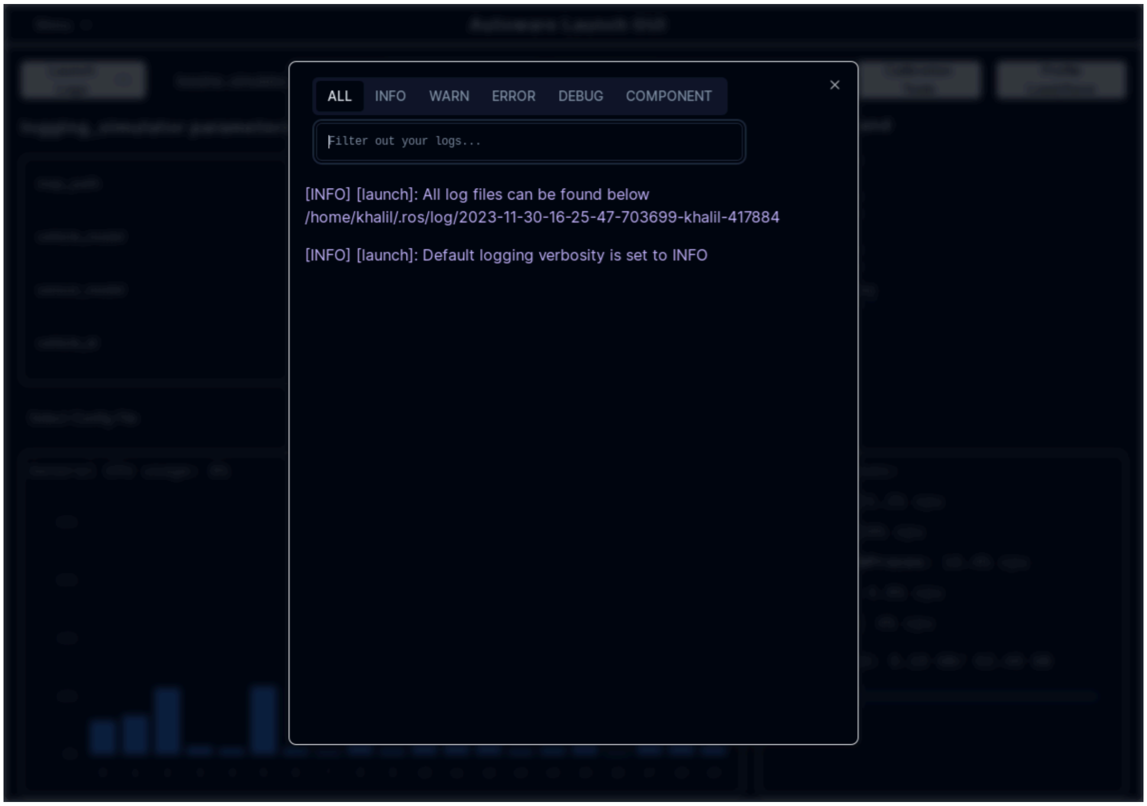
vehicle_model: sample_vehicle

sensor_model: sample_sensor_kit

```
General CPU usage: 5%
```

```
Top processes:
  chrome: 25.2% cpu
  chrome: 24.3% cpu
  WebKitWebProces: 8.6% cpu
  code: 7.3% cpu
  Discord: 5% cpu
Memory used: 9.44 GB/ 62.48 GB
```

4. **Start Simulation:** Click the launch button to start the simulation.



```
[INFO] [launch]: All log files can be found below
/home/khalil/.ros/log/2023-11-30-16-25-47-703699-khalil-417884

[INFO] [launch]: Default logging verbosity is set to INFO
```

5. **Any Scenario:** From here, you can follow the instructions in the

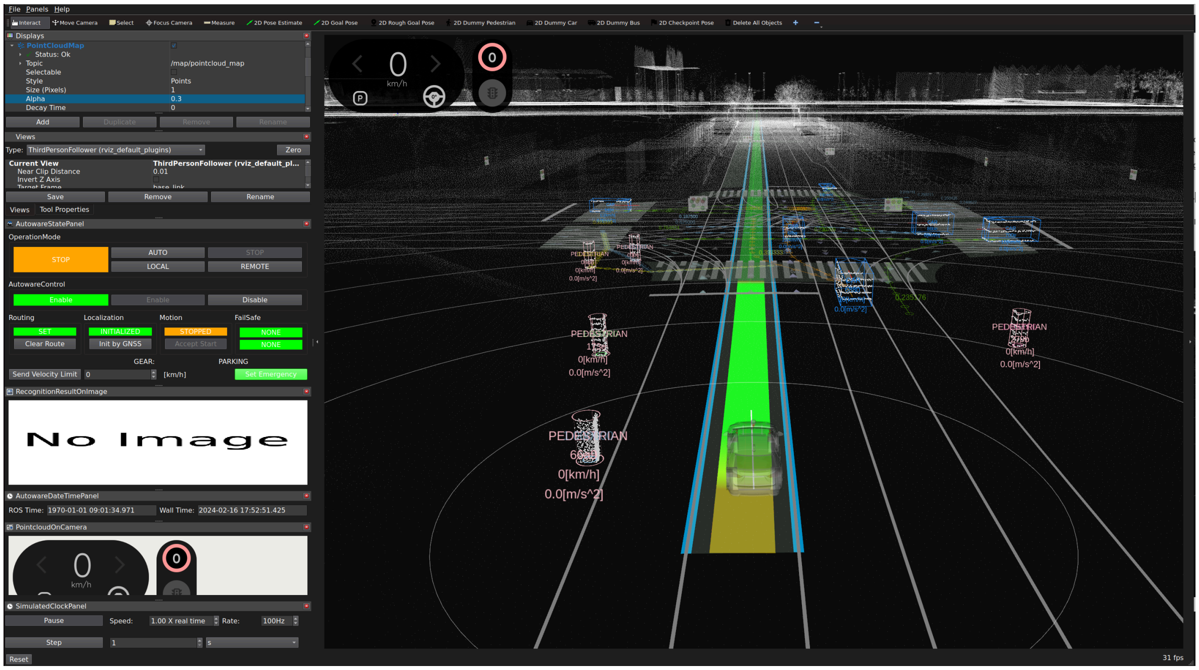
- Lane driving scenario: Lane Driving Scenario
- Parking scenario: Parking scenario
- Lane change scenario: Lane change scenario
- Avoidance scenario: Avoidance scenario
- Advanced Simulations: Advanced Simulations

Monitoring and Managing the Simulation

- **Real-Time Monitoring:** Use the GUI to monitor CPU/Memory usage and Autoware logs in real-time.
- **Profile Management:** Save your simulation profiles for quick access in future simulations.
- **Adjusting Parameters:** Easily modify simulation parameters on-the-fly through the GUI.

Want to Try Autoware with Your Custom Map?

The above content describes the process for conducting some operations in the planning simulator using a sample map. If you are interested in running Autoware with maps of your own environment, please visit the How to Create Vector Map section for guidance.



PointCloudMap
Status: Ok
Topic /map/pointcloud_map
Selectable
Style Points
Size (Pixels) 1
Alpha 0.3
Decay Time 0

Type: ThirdPersonFollower (rviz_default_plugins)

Current View ThirdPersonFollower (rviz_default_pl...
Near Clip Distance 0.01
Invert Z Axis
Target Frame base_link

Send Velocity Limit 0 [km/h]

ROS Time: 1970-01-01 09:01:34.971 Wall Time: 2024-02-16 17:52:51.425

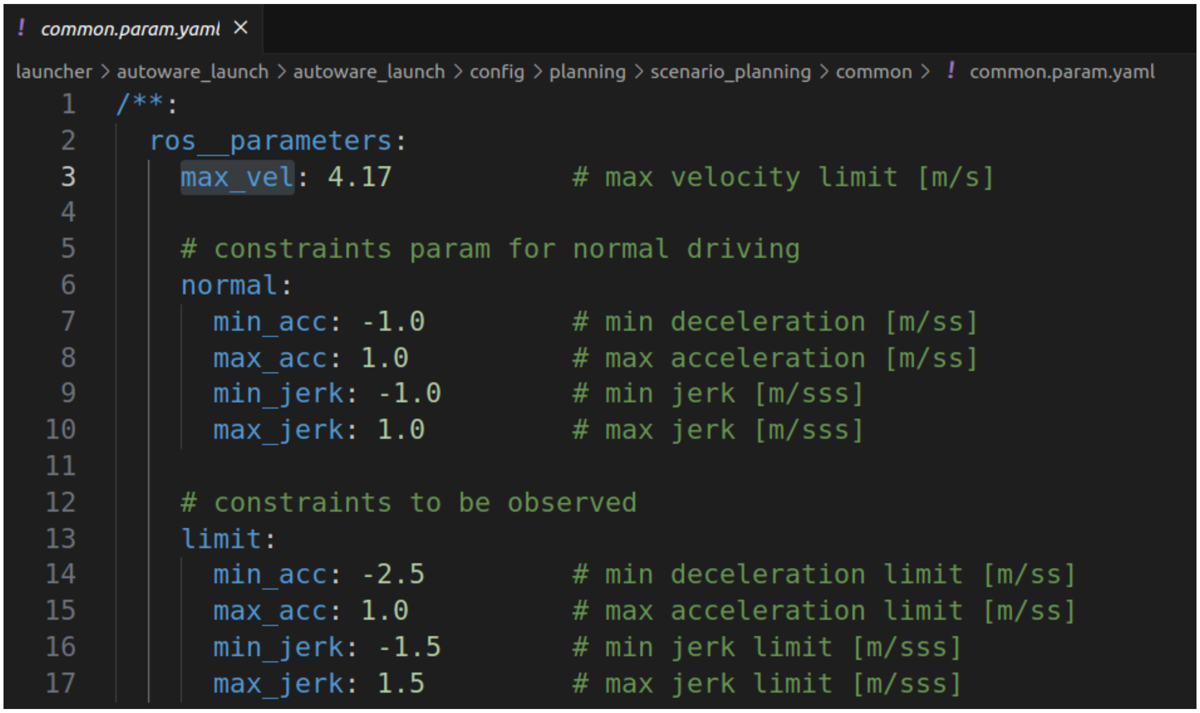
Speed: 1.00 X real time Rate: 100Hz

Step 1 s

Want to Try Autoware with Higher Speed?

The original Autoware is designed to operate at a wide speed range. But for safety reasons, the default maximum velocity has been limited to 15 km/h. In that case, even you drag the slider to a higher speed in rviz panel, the system will not allow it.

If you want to run Autoware at a higher speed, you can modify the `max_vel` parameter in the config file common.param.yaml located in the `config` directory of your autoware_launch.

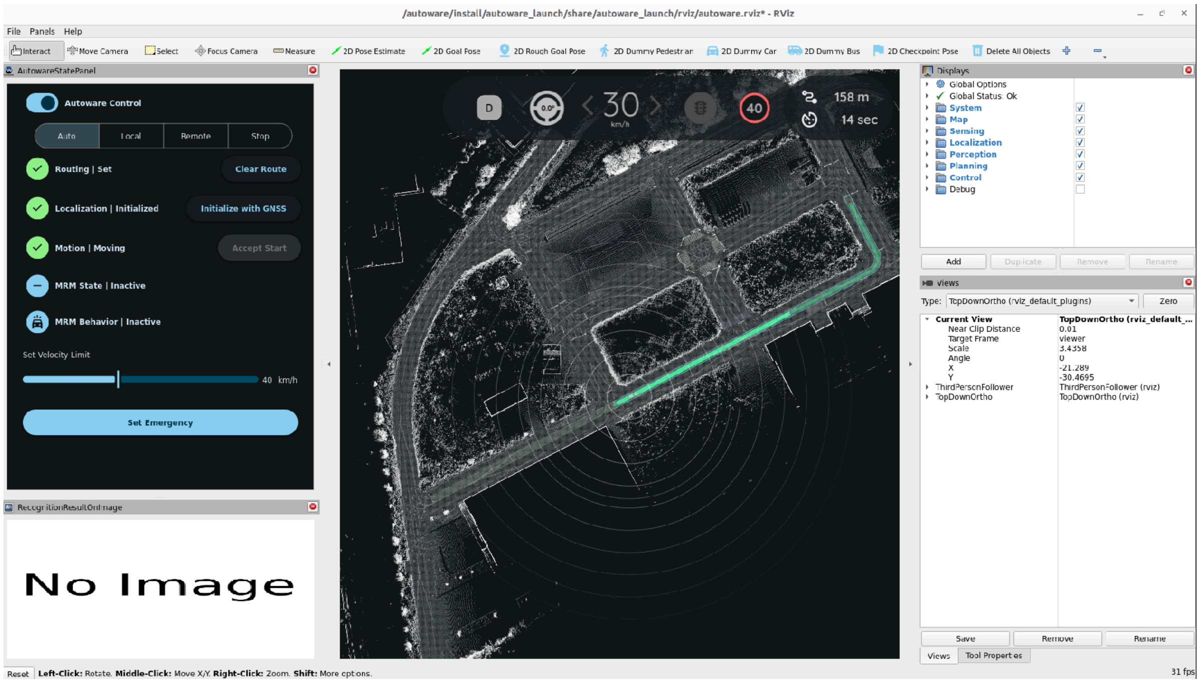

! common.param.yaml ×
launcher > autoware_launch > autoware_launch > config > planning > scenario_planning > common > ! common.param.yaml

```yaml
/**:
  ros__parameters:
    max_vel: 4.17          # max velocity limit [m/s]

    # constraints param for normal driving
    normal:
      min_acc: -1.0        # min deceleration [m/ss]
      max_acc: 1.0         # max acceleration [m/ss]
      min_jerk: -1.0       # min jerk [m/sss]
      max_jerk: 1.0        # max jerk [m/sss]

    # constraints to be observed
    limit:
      min_acc: -2.5        # min deceleration limit [m/ss]
      max_acc: 1.0         # max acceleration limit [m/ss]
      min_jerk: -1.5       # min jerk limit [m/sss]
      max_jerk: 1.5        # max jerk limit [m/sss]
```

For example, to set the maximum velocity to 72 km/h, you would change the value to `20.0` (since 72 km/h is equivalent to 20 m/s).



Routing | Set

Localization | Initialized

Motion | Moving

MRM State | Inactive

MRM Behavior | Inactive

Set Velocity Limit 40 km/h

Global Status: Ok

- System ☑
- Map ☑
- Sensing ☑
- Localization ☑
- Perception ☑
- Planning ☑
- Control ☑
- Debug □

Type: TopDownOrtho (rviz_default_plugins)

```
Current View             TopDownOrtho (rviz_default_...
  Near Clip Distance     0.01
  Target Frame           viewer
  Scale                  3.4358
  Angle                  0
  X                      -21.289
  Y                      -30.4695
ThirdPersonFollower      ThirdPersonFollower (rviz)
TopDownOrtho             TopDownOrtho (rviz)
```

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