/ Tutorials / Advanced / Simulators / Gazebo / Setting up a robot simulation (Gazebo)

You're reading the documentation for an older, but still supported, version of ROS 2. For information on the latest version, please have a look at Iron.

Setting up a robot simulation (Gazebo)

Goal: Launch a Simulation with Gazebo and ROS 2

Tutorial level: Advanced

Time: 20 minutes

Contents

Prerequisites

First of all you should install ROS 2 and Gazebo. You have two options:

- Install from deb packages. To check which versions are available from deb packages please check this table.
- Compile from sources:
  - ROS 2 install instructions
  - Gazebo install instructions

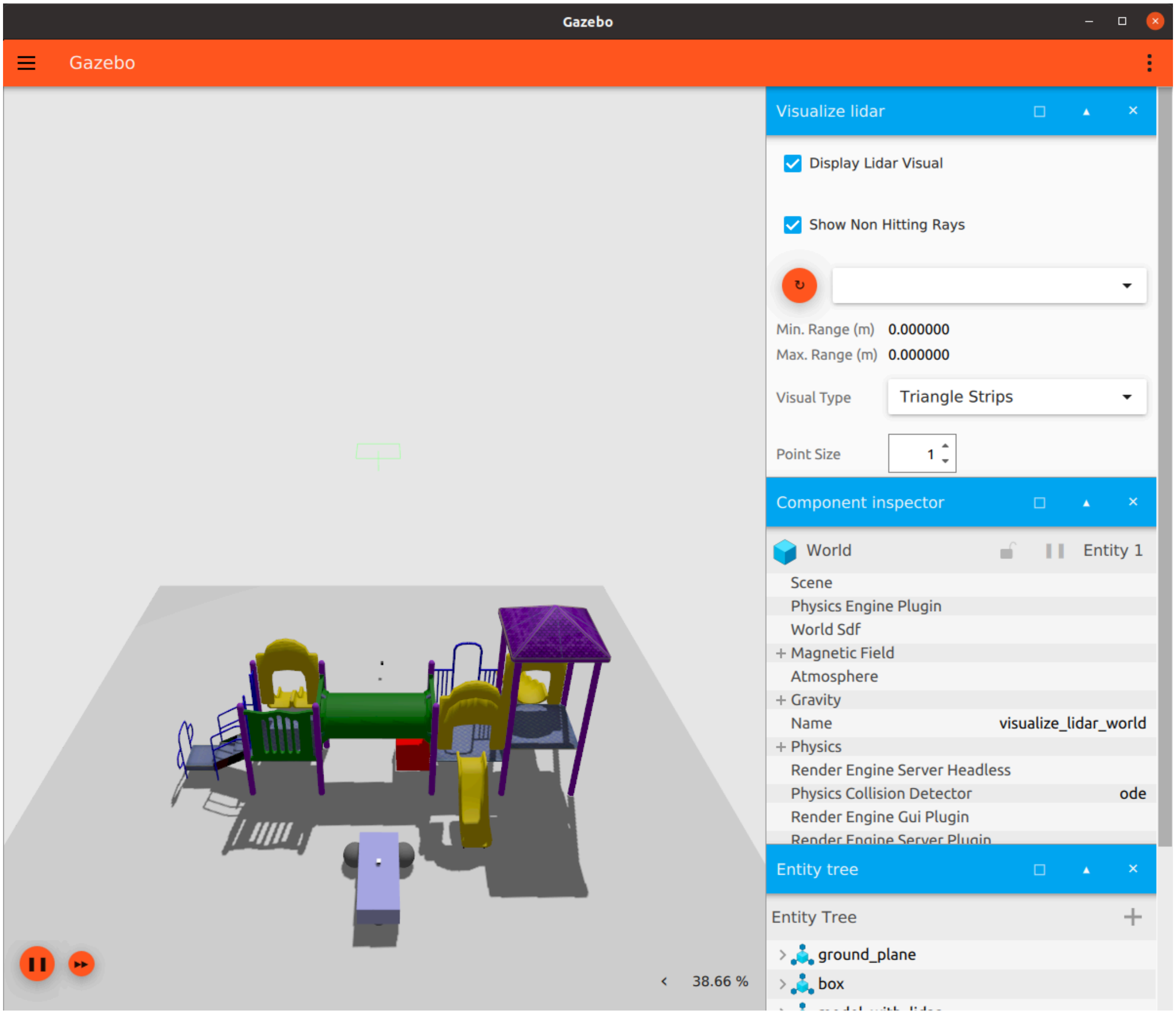
Tasks

1 Launch the simulation

In this demo you are going to simulate a simple diff drive robot in Gazebo. You are going to use one of the worlds defined in the Gazebo examples called visualize_lidar.sdf. To run this example you should execute the following command in a terminal:

Linux

```
ign gazebo -v 4 -r visualize_lidar.sdf
```



☑ Display Lidar Visual

☑ Show Non Hitting Rays

Min. Range (m) 0.000000
Max. Range (m) 0.000000

Visual Type Triangle Strips

Point Size 1

World Entity 1

Scene
Physics Engine Plugin
World Sdf
+ Magnetic Field
Atmosphere
+ Gravity
Name visualize_lidar_world
+ Physics
Render Engine Server Headless
Physics Collision Detector ode
Render Engine Gui Plugin
Render Engine Server Plugin

ground_plane
box

When the simulation is running you can check the topics provided by Gazebo with the `ign` command line tool:

Linux

```
ign topic -l
```

Which should show:

```
/clock
/gazebo/resource_paths
/gui/camera/pose
/gui/record_video/stats
/model/vehicle_blue/odometry
/model/vehicle_blue/tf
/stats
/world/visualize_lidar_world/clock
/world/visualize_lidar_world/dynamic_pose/info
/world/visualize_lidar_world/pose/info
/world/visualize_lidar_world/scene/deletion
/world/visualize_lidar_world/scene/info
/world/visualize_lidar_world/state
/world/visualize_lidar_world/stats
```

Since you have not launched an ROS 2 nodes yet, the output from `ros2 topic list` should be free of any robot topics:

Linux

```
ros2 topic list
```

Which should show:

```
/parameter_events
/rosout
```

2 Configuring ROS 2

To be able to communicate our simulation with ROS 2 you need to use a package called `ros_gz_bridge`. This package provides a network bridge which enables the exchange of messages between ROS 2 and Gazebo Transport. You can install this package by typing:

Linux

```
sudo apt-get install ros-humble-ros-ign-bridge
```

At this point you are ready to launch a bridge from ROS to Gazebo. In particular you are going to create a bridge for the topic `/model/vehicle_blue/cmd_vel`:

Linux

```
source /opt/ros/humble/setup.bash
ros2 run ros_gz_bridge parameter_bridge
/model/vehicle_blue/cmd_vel@geometry_msgs/msg/Twist]ignition.msgs.Twist
```

For more details about the `ros_gz_bridge` please check this README .

Once the bridge is running the robot is able to follow your motor commands. There are two options:

- Send a command to the topic using `ros2 topic pub`

  Linux

  ```
  ros2 topic pub /model/vehicle_blue/cmd_vel geometry_msgs/Twist "linear: { x: 0.1 }"
  ```

- `teleop_twist_keyboard` package. This node takes keypresses from the keyboard and publishes them as Twist messages. You can install it typing:

Linux

```
sudo apt-get install ros-humble-teleop-twist-keyboard
```

The default topic where `teleop_twist_keyboard` is publishing Twist messages is `/cmd_vel` but you can remap this topic to make use of the topic used in the bridge:

Linux

```
source /opt/ros/humble/setup.bash
ros2 run teleop_twist_keyboard teleop_twist_keyboard --ros-args -r
/cmd_vel:=/model/vehicle_blue/cmd_vel
```

Which will show:

```
This node takes keypresses from the keyboard and publishes them
as Twist messages. It works best with a US keyboard layout.
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

For Holonomic mode (strafing), hold down the shift key:
---------------------------
   U    I    O
   J    K    L
   M    <    >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5       turn 1.0
```

3 Visualizing lidar data in ROS 2

The diff drive robot has a lidar. To send the data generated by Gazebo to ROS 2, you need to launch another bridge. In the case the data from the lidar is provided in the Gazebo Transport topic `/lidar2`, which you are going to remap in the bridge. This topic will be available under the topic `/lidar_scan`:

Linux

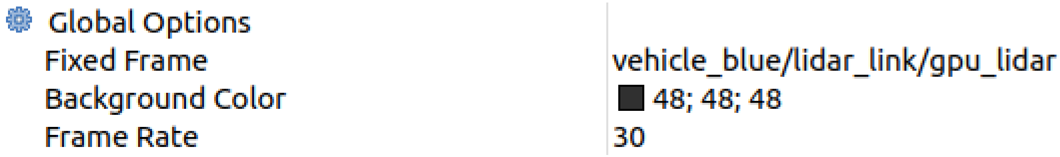
```
source /opt/ros/humble/setup.bash
ros2 run ros_gz_bridge parameter_bridge
/lidar2@sensor_msgs/msg/LaserScan[ignition.msgs.LaserScan --ros-args -r /lidar2:=/laser_scan
```

To visualize the data from the lidar in ROS 2 you can use Rviz2:

Linux

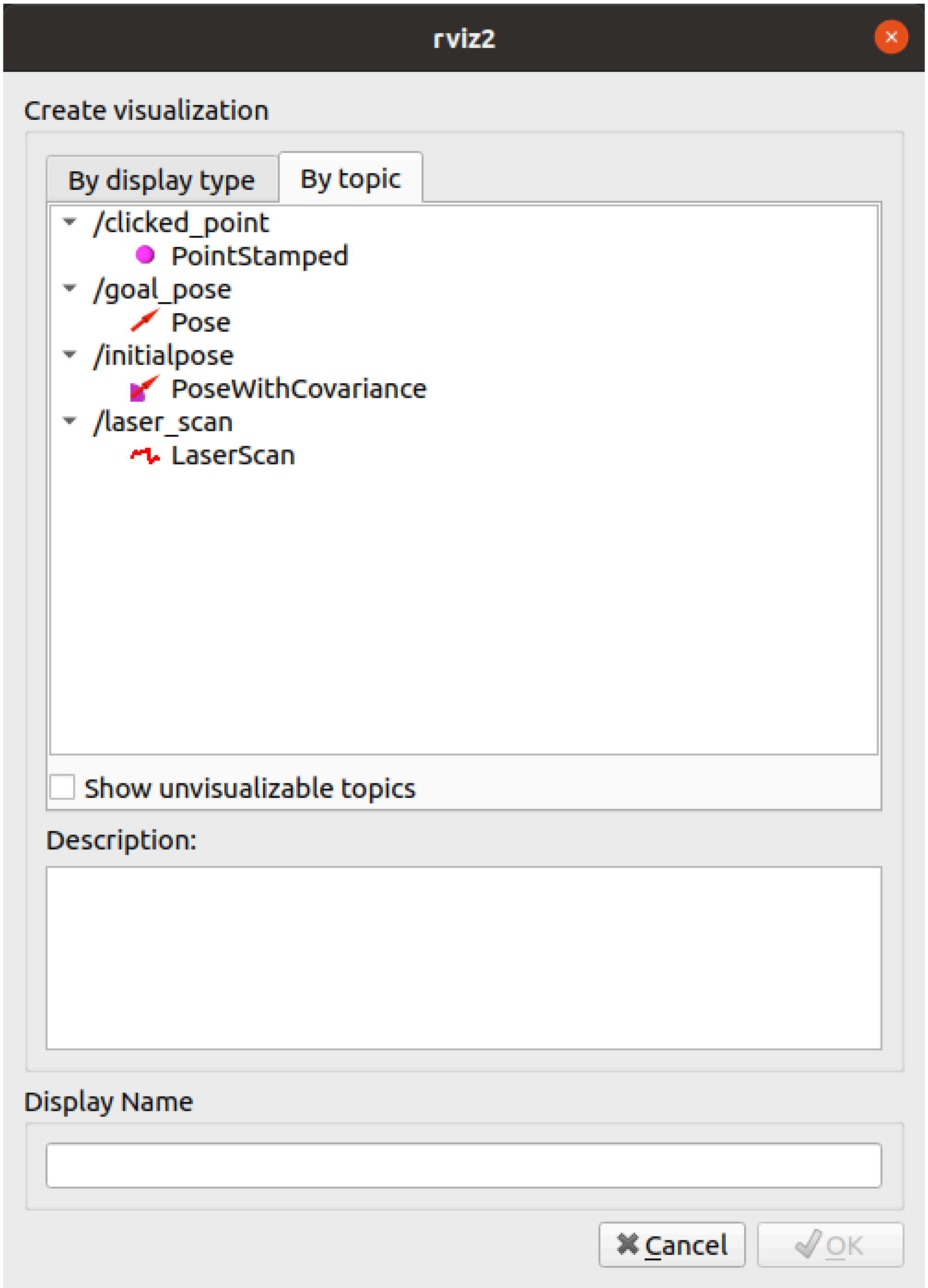
```
source /opt/ros/humble/setup.bash
rviz2
```

Then you need to configure the `fixed frame`:



Global Options	
Fixed Frame	vehicle_blue/lidar_link/gpu_lidar
Background Color	■ 48; 48; 48
Frame Rate	30

And then click in the button “Add” to include a display to visualize the lidar:



/clicked_point
- PointStamped

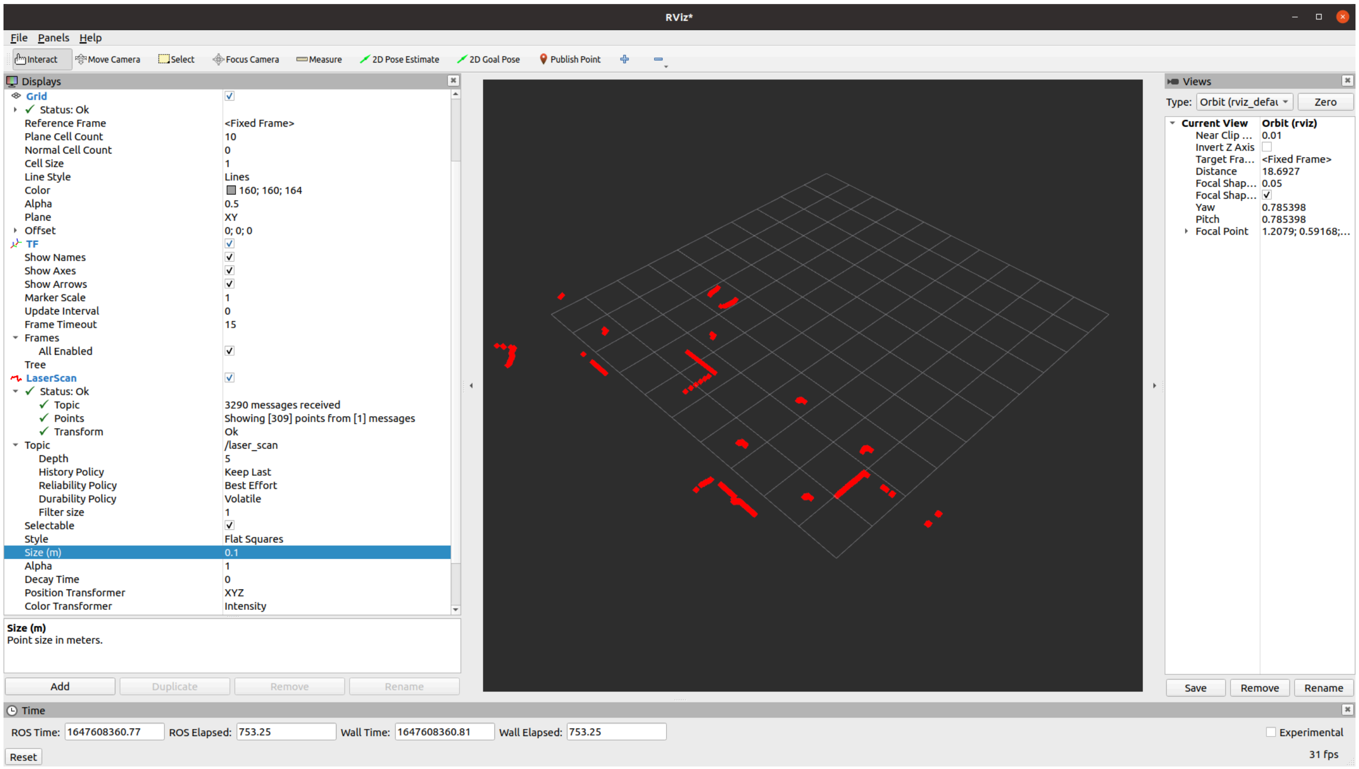
/goal_pose
- Pose

/initialpose
- PoseWithCovariance

/laser_scan
- LaserScan

□ Show unvisualizable topics

Now you should see the data from the lidar in Rviz2:



Grid	☑
✓ Status: Ok	
Reference Frame	<Fixed Frame>
Plane Cell Count	10
Normal Cell Count	0
Cell Size	1
Line Style	Lines
Color	▨ 160; 160; 164
Alpha	0.5
Plane	XY
Offset	0; 0; 0
TF	☑
Show Names	☑
Show Axes	☑
Show Arrows	☑
Marker Scale	1
Update Interval	0
Frame Timeout	15
Frames	
All Enabled	☑
Tree	
LaserScan	☑
✓ Status: Ok	
✓ Topic	3290 messages received
✓ Points	Showing [309] points from [1] messages
✓ Transform	Ok
Topic	/laser_scan
Depth	5
History Policy	Keep Last
Reliability Policy	Best Effort
Durability Policy	Volatile
Filter size	1
Selectable	☑
Style	Flat Squares
Size (m)	0.1
Alpha	1
Decay Time	0
Position Transformer	XYZ
Color Transformer	Intensity

Size (m)
Point size in meters.

Type: Orbit (rviz_defau

Current View	**Orbit (rviz)**
Near Clip …	0.01
Invert Z Axis	☐
Target Fra…	<Fixed Frame>
Distance	18.6927
Focal Shap…	0.05
Focal Shap…	☑
Yaw	0.785398
Pitch	0.785398
Focal Point	1.2079; 0.59168;…

ROS Time: 1647608360.77 ROS Elapsed: 753.25 Wall Time: 1647608360.81 Wall Elapsed: 753.25

☐ Experimental

31 fps

Summary

In this tutorial, you launched a robot simulation with Gazebo, launched bridges with actuators and sensors, visualized data from a sensor, and moved a diff drive robot.