



1 Introduction

Welcome! In this first mp you will get to play with the Gazebo simulator and use Rviz, two tools which will be pivotal in future mps.

Learning objectives

- Get introduced to ROS and Gazebo

System requirements

- Ubuntu 20.04
- ROS Noetic
- ros-kinetic-ackermann-msgs

2 Development Instructions

In this part, you will learn how to start and use the Gazebo simulator, use *roslaunch* to run different scripts, and configure and utilize *Rviz*.

2.1 Short Gazebo Tutorial

The code for MP0 will come together inside the ROS workspace, which will also be used in later MPs. In this MP, you will work with vehicle models in Gazebo simulator using ROS.

To run the simulator, open a new terminal tab, and run the following commands:

```
source /opt/ros/noetic/setup . bash
cd ~/Desktop/CodeACar22/
catkin_make
```

There should be no error during the execution of the command and when finished, you should see two folders *devel* and *build* been generated.

The next step is to run command:

```
source ./devel/setup . bash
```

After all the previous setup steps are finished, you can start the simulator by running command

```
roslaunch mp0 track.launch
```

You should be able to see two windows opened as shown in Figure 1 and 4.

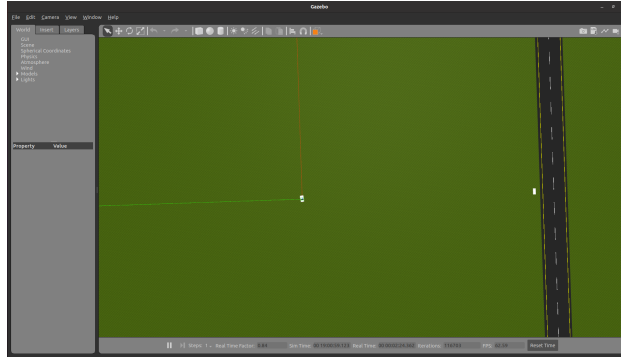


Figure 1: The Gazebo window. The vehicle is stopped before the pedestrian.

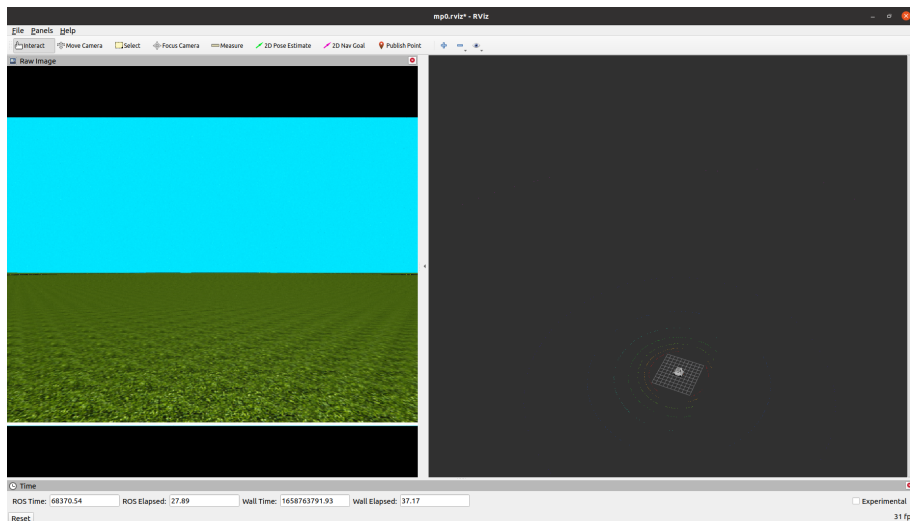


Figure 2: The RIVZ window corresponds to the Gazebo window in Figure 1.

Each launch file you select, such as *track.launch*, will load a different world into Gazebo. To load other worlds, you can hold *Ctrl* and *c* in the terminal tab where you ran the *roslaunch* command to close the gazebo window and type any of the following commands:

```
roslaunch mp0 eceb.launch  
roslaunch mp0 highbay.launch  
roslaunch mp0 road.launch
```

Gazebo Window The window shown in Figure 1 is the Gazebo simulator window. You should be able to see the vehicle and pedestrian in the simulator window.

Rviz Window The window shown in Figure 4 is the Rviz window. Rviz is a ROS graphical interface that allows you to visualize ROS topics. In this MP, we will mainly use Rviz to visualize LIDAR data from the vehicle. In this MP, the Rviz window is divided into 4 different sections.

The window on the right will show a visualization of the point cloud data from the LIDAR.

The window on the top left (Raw Image) is showing the video streaming from the camera on the vehicle.

2.2 Running Manual Control

To interact with the vehicle in Gazebo, you utilize the *rospy* library and write scripts. We have developed a script for you to take manual control of the vehicle.

In a new terminal enter the following commands:

First in a new terminal, you need to load up the paths to many of the scripts ROS requires.

```
source /opt/ros/noetic/setup . bash
source ~/Desktop/CodeACar22/devel/setup . bash
```

Now run:

```
roslaunch mp0 keyboard .py
```

A small Tkinter window will appear on the screen. To control the vehicle, make sure it is in focused application on your screen. You can use either the Gazebo or Rviz window to follow the vehicle while you move it around the world.

To move the vehicle, you can use *WASD* or the arrow keys. If you press *R*, it will reset the vehicle position.

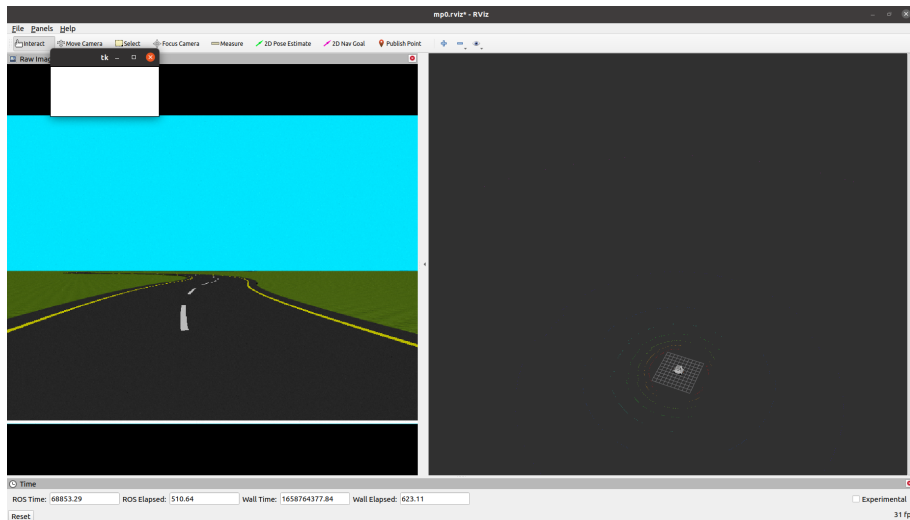


Figure 3: Setup using Rviz.

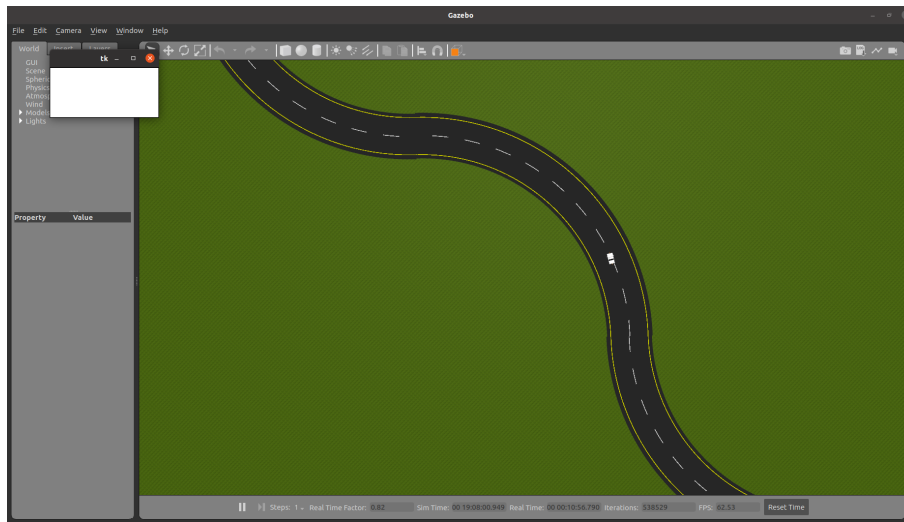


Figure 4: Setup using Gazebo.