Introduction to Robotics Institut für Informatik, Freie Universität Berlin
WiSe 2021/22 Prof. Daniel Goehring, Stephan Sundermann, Khaled Alomari

Exercise 2 October 24, 2021

Submission online until **Tuesday, 02.11.2021, 11:55 a.m.**

We recommend using autominy inside a native Ubuntu 20.04 installation. If you want to try regardless inside a virtual machine, reducing the OpenGL level to 2.0 can fix some graphic glitches and increase performance significantly on some VMs using the following command:

```
SVGA_VGPU10=0 roslaunch autominy Simulated.launch
```

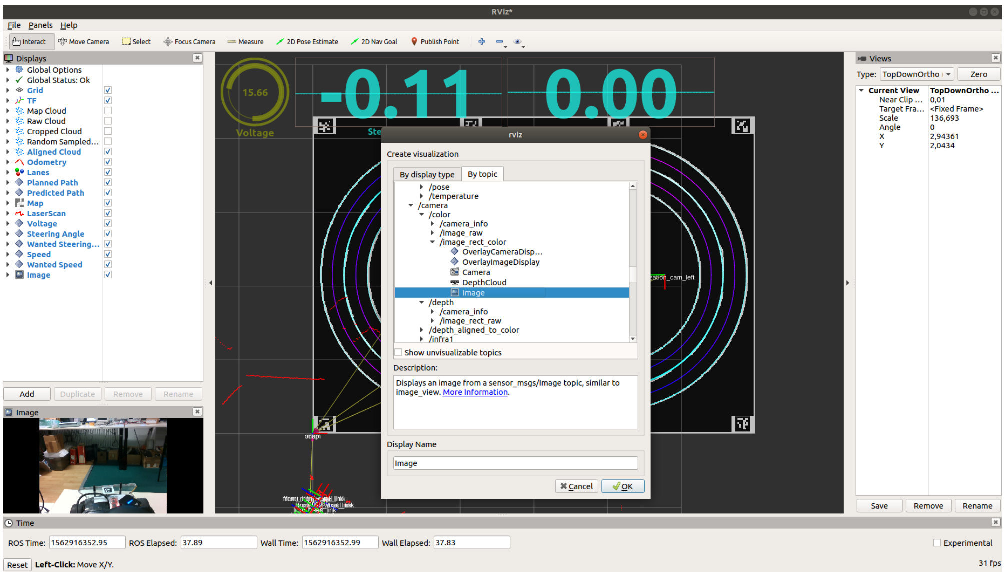
Assignment 2-1: RViz (4 Points)

Run rviz. Record a camera image of the robot model car, as the image is seen in the robot data visualization tool `rviz`.

Use the command `rostopic list`, you should see a list of topics, coming from the car now. Start `rviz` to visualize the car camera data as shown in the tutorial.

- Run `rviz` (type `rviz` in terminal)
- Click add button (bottom left)
- Click on `By topic` tab
- Select `/sensors/camera/color/image_raw_rect/Image`

Figure 1: Take a screenshot of the camera image in `rviz`. Include this image in your final pdf.



Displays an image from a sensor_msgs/Image topic, similar to image_view. More Information.

Display Name: Image

Type: TopDownOrtho

Current View: TopDownOrtho …
Near Clip …: 0,01
Target Fra…: <Fixed Frame>
Scale: 136,693
Angle: 0
X: 2,94361
Y: 2,0434

ROS Time: 1562916352.95 ROS Elapsed: 37.89 Wall Time: 1562916352.99 Wall Elapsed: 37.83

Left-Click: Move X/Y.

Assignment 2-2: Basic Subscriber / Publisher Node (6 Points)

The goal of this task is to write a two python nodes. The first node should should subscribe to the car's speed. The second node should let the car drive in a circle. This task is similar to the talker / chatter example from ROS and you might find some useful information there:

`http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29`

Create a new catkin package `assignment2_publisher_subscriber` in the `src` folder of the `catkin_ws_TEAMNAME` workspace on your machine. It should contain the catkin dependencies `rospy` and `autominy_msgs`. For creating a package, you can use the command: `catkin create pkg`. See `http://wiki.ros.org/ROS/Tutorials/catkin/CreatingPackage` for more information and the documentation from catkin tools at `https://catkin-tools.readthedocs.io/en/latest/verbs/catkin_create.html`.

In the terminal navigate to assignment2_publisher_subscriber/src directory. Create an empty python script file for the subscriber and publisher and mark it as executable with:

```
touch subscriber.py
chmod +x subscriber.py
touch publisher.py
chmod +x publisher.py
```

Write a simple node which subscribes to the topic: `/sensors/speed` The data type of this topic is `autominy_msgs/Speed` and it contains information about the speed from the motor. You can get the type on the current running ROS system with: `rostopic type /topic_name`.

For subscribing you need to define a callback function which is called, when a new message is received. Inside this callback function simply print the value of the speed message to the console.

Navigate to your workspace directory and compile the package with: `catkin build assignment2_publisher_subscriber`

For the second node publish to the topic `/actuators/steering_normalized` and publish an `autominy_msgs/NormalizedSteeringCommand` with a value of 1.0 to steer maximum to the left. In the same node add another publisher and publish to the topic `/actuators/speed`. This topic receives an `autominy_msgs/SpeedCommand` message for the speed motor. Using the publisher send a speed message to drive the car at low speed ($0.3m/s$). Publish these messages periodically.

Take a screenshot when the simulated car is driving and show that your node is printing the car's current speed. Commit the source code to your catkin_ws_TEAMNAME git repository and put a link to your source code in your final pdf.