

2. Assignment, Introduction to Robotics WS18/19

1. (3 Points)

Record a camera image of the robot model car, as the image is seen in the robot data visualization tool “rviz”

Go to the tutorial for the model car:

<https://github.com/AutoModelCar/AutoModelCarWiki/wiki>

To communicate with the car, it mainly boils down to the following steps, where you have to change some environment variables for ROS:

<https://github.com/AutoModelCar/AutoModelCarWiki/wiki/MultipleMachines>

Perform the following steps in your console:

- export ROS_MASTER_URI=http:// ip of the target machine, followed by port number :11311
- export ROS_IP= ip of the local machine
- export ROS_HOSTNAME= ip of the local machine

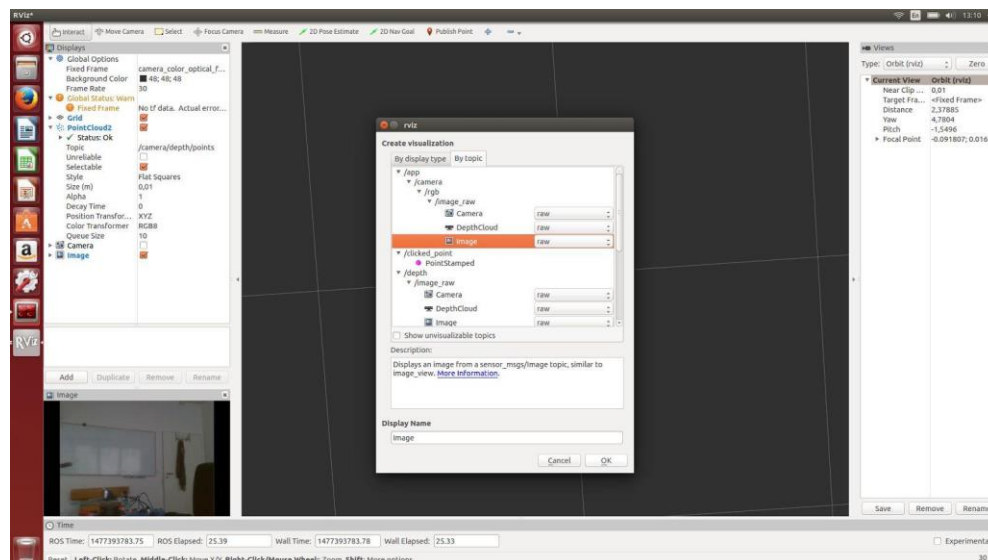
Use the command “rostopic list”, you should see a list of topics, coming from the car now.

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 app/camera/rgb/image_raw/Image

Take a screen shot of the camera image in rviz. Include this image in your final pdf.

(See example below)



2. (7 Points)

If not already installed, install the package ***python-catkin-tools***.

Create a new catkin package “assignment1_publisher_subscriber” in the `src` folder of the `catkin_ws_user` workspace on your machine. It should contain the catkin dependencies `rospy` and `std_msgs`. For creating a package, you can use the command:

catkin create pkg

See http://catkin-tools.readthedocs.io/en/latest/verbs/catkin_create.html for more information.

In the terminal navigate to the created `assignment1_publisher_subscriber/src` directory. Create an empty python script file and mark it as executable with:

touch publisher_subscriber.py

chmod +x publisher_subscriber.py

Write a simple node which subscribes to the topic: **/yaw**

The data type of this topic is `std_msgs/Float32`. 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 function publish a new message with the topic name “/assignment1_publisher_subscriber” and of type `std_msgs/String` which contains some words like ‘I heard: ’ concatenated by the currently received yaw value.

Navigate to the `catkin_ws_user` directory and compile the package with:

catkin build assignment1_publisher_subscriber

In connection with the model car, run your created node on your machine and print your published messages inside the remote terminal (SSH connection) with:

rostopic echo assignment1_publisher_subscriber

If you turn the model car in horizontal axis you should see that the yaw angle is changing.

Commit the source code to your catkin_ws_user git repository. Put a screenshot of the rostopic echo output and a link to your source code in your final pdf.

