University of Bremen

German Research Center for Artificial Intelligence

Worksheet 01: Ubuntu and ROS 2

Deadline: 02.11.2022
Robot Design Lab
03-IBGA-FI-RDL
WiSe 22-23

Universität Bremen
FB 3 – Mathematik und Informatik
Arbeitsgruppe Robotik
Prof. Dr. Dr. h.c. Frank Kirchner
Dr. rer. nat. Teena Hassan
M. Sc. Mihaela Popescu

---

1 Applications of Robotics 12%

Provide answers to these questions:

1. Name four application fields of robotics and describe for each of them a task that robots have to do. Which UN SDG goals are addressed by each of these applications? (6%)
2. Briefly explain three developments to which the Shakey robot project contributed. (3%)
3. Distinguish between teleoperation and full autonomy and give an example for each case. (2%)
4. Do you have a dream application for a robot that you would like to build yourselves in the future? Let us know! (1%)

2 Getting Started with Ubuntu 20%

1. Let's get familiar to the terminal window in Ubuntu. Open a new terminal window, try out the following commands with suitable parameters of your choice and describe briefly what each of them does. (10%)
   (a) `pwd`
   (b) `ls`
   (c) `cd`
   (d) `mkdir`
   (e) `touch`
   (f) `echo`
   (g) `cat`
   (h) `cp`
   (i) `mv`
   (j) `rm`

   Hint. Check out this reference https://ubuntu.com/tutorials/command-line-for-beginners#1-overview and the attached *Linux cheatsheet* document. In order to access the documentation of the commands, type `command --help` in the terminal window, for instance `ls --help`.
2. In a terminal window, create a new folder called `task2` and a new file called `robots.txt` in the folder. Write 3 types of robots in the text file using a command in the terminal window and add 3 robotic applications using the editor `nano`. Make a copy of the `robots.txt` file in the same directory called `robots_duplicate.txt`. For this task, use only commands in the terminal window. Submit a screenshot of the terminal window with the used commands and the folder `task2`. (6%)
3. What is root permission in Ubuntu and how can you execute a command with root permission? (2%)
4. Which commands can you use to verify: (2%)
   (a) How much memory is free?
   (b) The memory load of each running process?

3 Quiz on ROS2 15%

Provide answers to these questions:

1. What is the difference between **ros2 run** and **ros2 launch**? (2%)
2. What does the command **ros2 interface show std_msgs/String** do? (2%)
3. How can you verify that a ROS2 node is subscribing to a topic? (2%) [1]
4. Name two communication paradigms used in ROS2 to send and receive information between nodes. (2%)
5. Name three differences between ROS and ROS2. (3%)
6. Your robot has a camera that sends a video stream. Would you use a 'Service' or a 'Topic' to provide the video to a human that controls the robot with a joypad? Justify your answer. (2%)
7. Your robot is moving towards location G. As it is approaching location G, you want it to take a picture. Would you use a 'Service' or a 'Topic' to obtain this picture? Justify your answer (2%)

4 Write ROS2 Nodes to Operate on Numbers 45%

Note: This exercise involves programming using Python3. You can use the tutorial at this link to familiarise yourself with Python3: https://www.tutorialspoint.com/python3/index.htm

Read the simple ROS2 publisher and subscriber code provided at this link:
https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Writing-A-Simple-Py-Publisher-And-Subscriber.html
and answer the following questions:

1. What is the type of data that is published and what is the name of the topic on which it is published? (5%)
2. How many times per second does the publisher publish the data? How can you change this so that four messages are published per second? (5%)
3. Rewrite the publisher so that it publishes an integer. (10%)
4. Rewrite the subscriber so that it receives the radius of a circle of type integer, prints (i) the area of the circle if the radius is even or (ii) the circumference of the circle if the radius is odd. (20%)
5. Run the publisher and subscriber code. Provide screenshots of the outputs generated by the subscriber. (5%)

Please submit the following:

1. **Embed** relevant parts of your code in the PDF and **explain** briefly the changes that you made.
   Hint: Take a look at the LaTeX -template to see how to add code in the document.
2. **Attach** a zip file of the ROS 2 package containing your changed code.

5 Teleoperation of TurtleBot3 8%

Follow the instructions below to connect to and control your TurtleBot3 using the keyboard of your notebook.

1. Install Ubuntu 22.04 and ROS 2 Humble, either directly on your PC or in a VirtualBox. For more details, consult the installation guide provided in StudIP.

> **Important:**
> We highly recommend you install Linux and ROS2 directly on your PC (for example as Dual boot if you need Windows)! During testing we noticed that the VirtualBox sometimes loses the connection to the robot so it should only be used for testing, not for longer tasks like mapping or a competition!

[1] You can use ROS2-tools, you do NOT have to deliver C++/Python code to answer this question

2. Connect to the WiFi Network that we provide as follows:
   `SSID: RobotDesignLab1_5G` (for the Groups 01-06)
   `SSID: RobotDesignLab2_5G` (for the Groups 07-12)
   `Password: 12345678`

3. After the ROS2-installation, you need to configure the `.bashrc` file to have the same ROS_DOMAIN_ID[2] as your robot. Your robot's ROS_DOMAIN_ID should be 30+the number on the top of the laser scanner, so between 31 (for robot R1.1) and 42 (for robot R1.12). Open the file `.bashrc` with a text editor on your notebook and change or add the following lines:
   `source /opt/ros/humble/setup.bash`
   `export ROS_DOMAIN_ID=30` (change 30 to the number of your robot)
   `export LDS_MODEL='LDS-01'`
   `export TURTLEBOT3_MODEL='burger'`

4. Source the `.bashrc` file as follows:
   `source .bashrc`

5. Install the dependencies on your PC or Virtual Machine. If you use the Virtual Machine provided by us it should be already installed:
   `sudo apt install ros-humble-turtlebot3-teleop`

6. Connect the robot to the power supply or battery and turn it on by switching on the button. Make sure to watch the battery handling video and read the safety rules before using batteries.

7. Connect to the robot using `ssh` and the static IP of the robot: 192.168.1.30+robot number:
   `ssh ubuntu@192.168.1.30+no.of the robot`
   `Password: robotics`

8. Launch the ROS 2 nodes on the robot:
   `ros2 launch turtlebot3_bringup robot.launch.py`

9. In a new terminal window on your PC, start the node that allows you to control the robot:
   `ros2 run turtlebot3_teleop teleop_keyboard`

10. Move the robot around with keys and identify which keys perform which movement. **Fill up Table 1 and provide sequences of images showing your robot at different positions and orientations.**

Key	**Movement**
a	Steer to the left.
s	...
d	...
...	...

Tabelle 1: Template of a table showing robot steering commands.

11. After you finish teleoperation, close all nodes with `CTRL+C` and shut down the robot using the command:
    `sudo shutdown -h 0`

12. Wait until the green light is not flashing anymore and switch off the robot using the power button. This will prevent the accidental loss of data by cutting off the power when the robot control board is running. If you used a battery, remove it from the robot in order to be recharged.

13. If you use a virtual machine, turn off Ubuntu normally from the graphical interface or from the command line. Do NOT exit the virtual machine from the closing button on the upper right corner, as this might lead to the loss of all your installations and data in the virtual box image!

[2]It's not important now to know much about it, but if you are interested in the network functionality behind ROS 2, read the chapter about Domain-IDs here: https://docs.ros.org/en/humble/Concepts/About-Domain-ID.html

6 Feedback

Your feedback is very important to us. Please briefly answer the following questions:

1. How much time did you spend on doing this sheet per person? Anonymize your answer!
2. Was it too easy, easy, ok, hard, too hard?
3. Please tell us what you liked in this exercise sheet.
4. Did you face any difficulties? What should be improved?
5. Any other general remarks?

7 Submission Procedure

- Please use the LaTeX template provided in *StudIP/Wiki* to write your solutions. Upload the pdf file together with source code and other additional materials as a .zip file in Stud.IP under *Files/Submissions/ex01*.
- The naming style of your submission should follow the pattern **Gxx_0y_lastname1_lastname2_lastname3.zip**, where xx stands for the group number and y stands for the exercise sheet number.