Laboratory Guide

Introduction to Robotics

2024-2025 (P1 / S1)

Rita Cunha and José Gaspar
(mostly the same as Alberto Vale's guide from 2022)
DEEC/IST

TÉCNICO LISBOA

Pre-requisites and Installation

Pre-requisites

- **Personal laptop:**
  - Ubuntu 20.04 LTS / 18.04LTS or 16.04LTS (ROS1 is not supported in 22.04LTS)
  - Windows:
    1. Dual boot (with one of the Ubuntu versions mention above)
    2. Virtual machine with at least 4GB of RAM and 20GB of space (**VMware recommended**)*
    3. Using ROS 1 on Windows 10 is not recommended
    4. Using Docker and ROS 1 image not recommended (unless you are already familiar with it)

- **Laboratory computers (alternative):**
  - Ubuntu with ROS 1 pre-installed
  - Computers are connected to the deec-robots network (the same as all robots)
    - Username: xxxxx
    - Password: xxxxxxxxxxx

- Check https://si.tecnico.ulisboa.pt/software/vmware/ for free student license.
- VirtualBox can also be used, but not recommended.

Installing Linux and ROS 1

- Each ROS 1 version is linked to an Ubuntu LTS release

Operating System	ROS 1 Version	Python Version
Ubuntu 16.04LTS	ROS Kinetic	2
Ubuntu 18.04LTS	ROS Melodic	2
Ubuntu 20.04LTS **(recommended)**	ROS Noetic	3
Ubuntu 22.04LTS	Not Supported	---

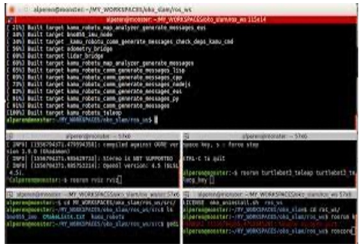
- Desktop-Full Install is **recommended**
- If you have trouble with the official ROS Noetic installation, follow the instructions in https://varhowto.com/install-ros-noetic-ubuntu-20-04

Recommendations

1. USE TERMINATOR

- On the laptop, run

```
sudo apt-get install terminator
```



Tips and Tricks

```
ctrl + shift + C (copy)
ctrl + shift + V (paste)
ctrl + shift + O (split the terminal horizontally)
ctrl + shift + E (split the terminal vertically)
ctrl + shift + Z (focus/unfocus a terminal window)
```

2. DEVELOP YOUR PROJECT IN PYTHON

Experience with GIT **[recommended]**

- If you have not used **GIT** yet, sooner or later you will need **it**!
- For a brief intro:
  https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F
- Create an account and backup your work here:
  GitHub: Where the world builds software https://github.com/

Introduction to ROS

ROS introduction and short courses

- "Introduction to the Robot Operating System (ROS)", from Rodrigo Ventura **[mandatory if this is your first contact with ROS]**
- Short course videos from Rodrigo Ventura and João Avelino **[complementary]**
  - Part 1: https://www.youtube.com/watch?v=3aVYUAj7sr4&t=1205s&ab_channel=RodrigoVentura
  - Part 2: https://www.youtube.com/watch?v=zqpKWHHlgOA&ab_channel=RodrigoVentura

Short Course on ROS Programming

2020 edition - Part 1

http://bit.ly/3lYxZrC

Rodrigo Ventura João Avelino

Institute for Systems and Robotics
Instituto Superior Técnico
Portugal
{rodrigo.ventura, javelino }@isr.tecnico.ulisboa.pt

TÉCNICO LISBOA
ISR
LARSyS
IEEE
IEEE Robotics & Automation Society
Portugal Chapter
EUROPEAN ROBOTICS WEEK
2:04:42
Institute for Systems and Robotics - Lisboa

Short Course on ROS Programming

2020 edition - Part 2

http://bit.ly/3lYxZrC

Rodrigo Ventura João Avelino

Institute for Systems and Robotics
Instituto Superior Técnico
Portugal
{rodrigo.ventura, javelino }@isr.tecnico.ulisboa.pt

TÉCNICO LISBOA
ISR
LARSyS
IEEE
IEEE Robotics & Automation Society
Portugal Chapter
EUROPEAN ROBOTICS WEEK
4:01:53
Institute for Systems and Robotics - Lisboa

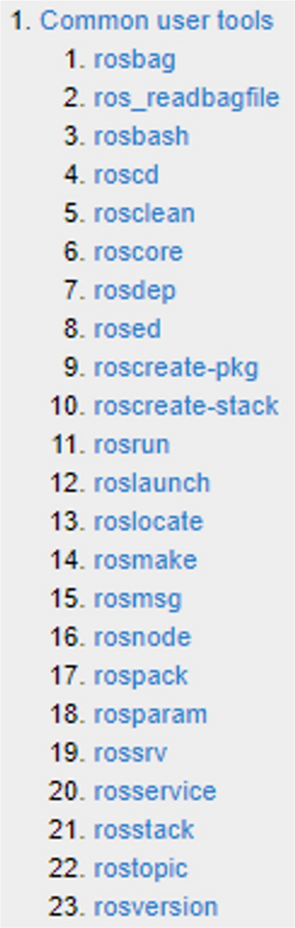
ROS Tutorials **[recommended]**

- Core ROS Tutorials: http://wiki.ros.org/ROS/Tutorials
  - Beginner Level **[all bullets recommended]**
  - Intermediate Level **[recommended: roslaunch tips, running ROS across multiple machines]**
- TF2 (TF deprecated): http://wiki.ros.org/tf2/Tutorials/
  more about TF2: https://articulatedrobotics.xyz/ready-for-ros-6-tf/
- Robot Model: http://wiki.ros.org/robot_model_tutorials
- Visualization: http://wiki.ros.org/visualization/Tutorials
- Navigation: http://wiki.ros.org/navigation/Tutorials

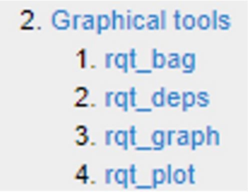
ROS relevant terms (**discover by yourself**)

- Package
- Node
- Topic
- Publish
- Subscribe
- Message
- Service
- Bag
- Parameters
- Launch files
- TF
- Rviz
- Gazebo

“Cheatsheet“ (ROS commands/params)



1. Common user tools
    1. rosbag
    2. ros_readbagfile
    3. rosbash
    4. roscd
    5. rosclean
    6. roscore
    7. rosdep
    8. rosed
    9. roscreate-pkg
    10. roscreate-stack
    11. rosrun
    12. roslaunch
    13. roslocate
    14. rosmake
    15. rosmsg
    16. rosnode
    17. rospack
    18. rosparam
    19. rossrv
    20. rosservice
    21. rosstack
    22. rostopic
    23. rosversion



2. Graphical tools
    1. rqt_bag
    2. rqt_deps
    3. rqt_graph
    4. rqt_plot

Useful for structure visualization of packages, nodes, topics and subscriptions

Hint:

Work with alias for the most used commands – **alias**

ROS Cheat-sheet

Relevant terms to search

Package	Bag
Node	Launch Files
Topic	Parameters
Publisher	TF
Subscriber	RVIZ
Message	Gazebo

Graphical user tools

```
rosrun rviz rviz

rosrun rqt_image_view rqt_image_view

rqt    (can be used to monitor)

rosrun rqt_tf_tree rqt_tf_tree

rosrun rqt_plot rqt_plot
```

Terminal user tools

```
roscore
rosrun <package_name> <node_name>
roslaunch <package_name> <launch_file>
roscd <package_name>

rostopic list
rostopic info <topic_name>
rostopic hz <topic_name>

rosnode list
rosnode info <node_name>

rosbag play <options>
rosbag record <options>

roscreate-pkg <options>

rosservice list
rosservice call <options>

rosmsg list
```

ROS code

- C/C++ **or** Python [**Python recommended if no proficiency with C/C++**]
  - For Python, install dependencies [**to run in your computer**]

    ```
    sudo apt-get install python3-rosdep
    sudo pip install -U rosdep
    sudo rosdep init
    ```

- MATLAB [**use ROS to record bags and read them in MATLAB**]

  Open and parse rosbag log file (since R2022a supports ROS Noetic):

  https://www.mathworks.com/help/ros/ref/rosbag.html

ROS Extra Notes

- Often forgotten – set the environment variables used by ROS

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

- Confirm the definition of the ROS packages path

```
echo $ROS_PACKAGE_PATH
```

- ROS/Linux editors: vim or nano [nano is recommended]
- Make use of the .bashrc file

```
source /opt/ros/noetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export EDITOR='nano -w'
```

- Copy a file from the robot to your laptop/Lab computer

IMPORTANT

```
scp <robot_username>@<robot_ip>:<path_to_file> <location_in_your_computer>
```

ROS Extra Notes [2]

- Explore the use of **`roslaunch`** and launch files
- Bag files may occupy too much space.

  Suggestion: record only the required topics and include compression

```
rosbag record -j <topics>
```

- If problems are detected, run

```
roswtf
```

- To install dependencies of a package, use rosdep

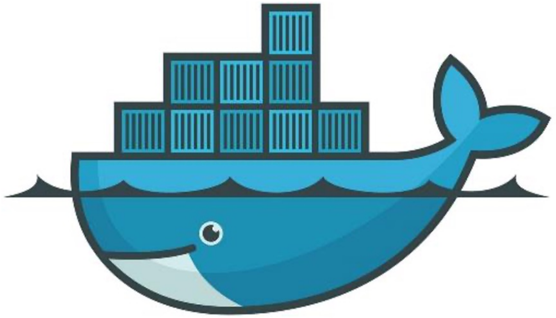
```
sudo apt-get install rosdep
rosdep init
rosdep install <package-name>
```

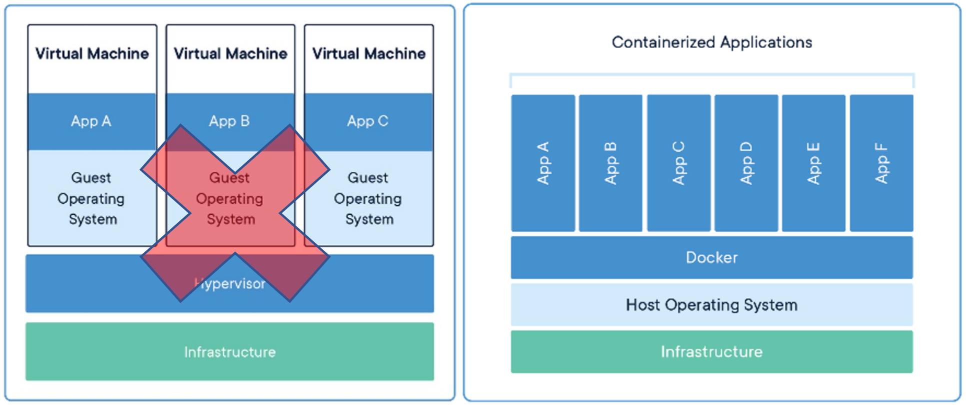
- ROS log files are located at

```
~/.ros/log
```

ROS with Dockers **[advanced]**

- Docker: *build self-sufficient, lightweight, and portable software containers that make creating, testing, and deploying applications easy and fast*
- **Docker image** + **Docker container**
- Docker images of ROS: https://hub.docker.com/_/ros
- For more information: https://www.docker.com/blog/top-questions-for-getting-started-with-docker/





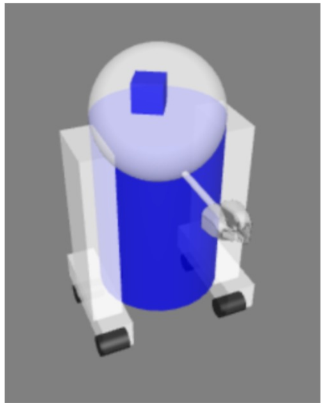
Gazebo

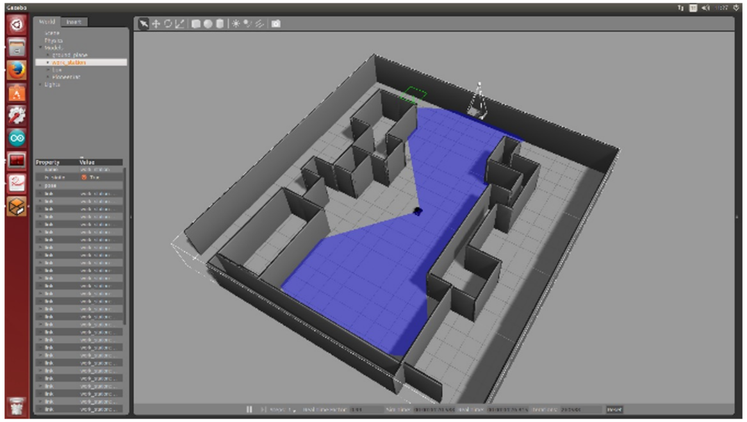
- Gazebo (simulate populations of robots in complex indoor and outdoor environments): http://gazebosim.org/tutorials?tut=ros_overview
- Install Gazebo: https://classic.gazebosim.org/tutorials?tut=install_ubuntu
- launch Gazebo GUI, client or server:

```
rosrun gazebo_ros gazebo
rosrun gazebo_ros gzclient
rosrun gazebo_ros gzserver
```

- Defaults are in a shell script:

```
source
<install_path>/share/gazebo/setup.sh
```





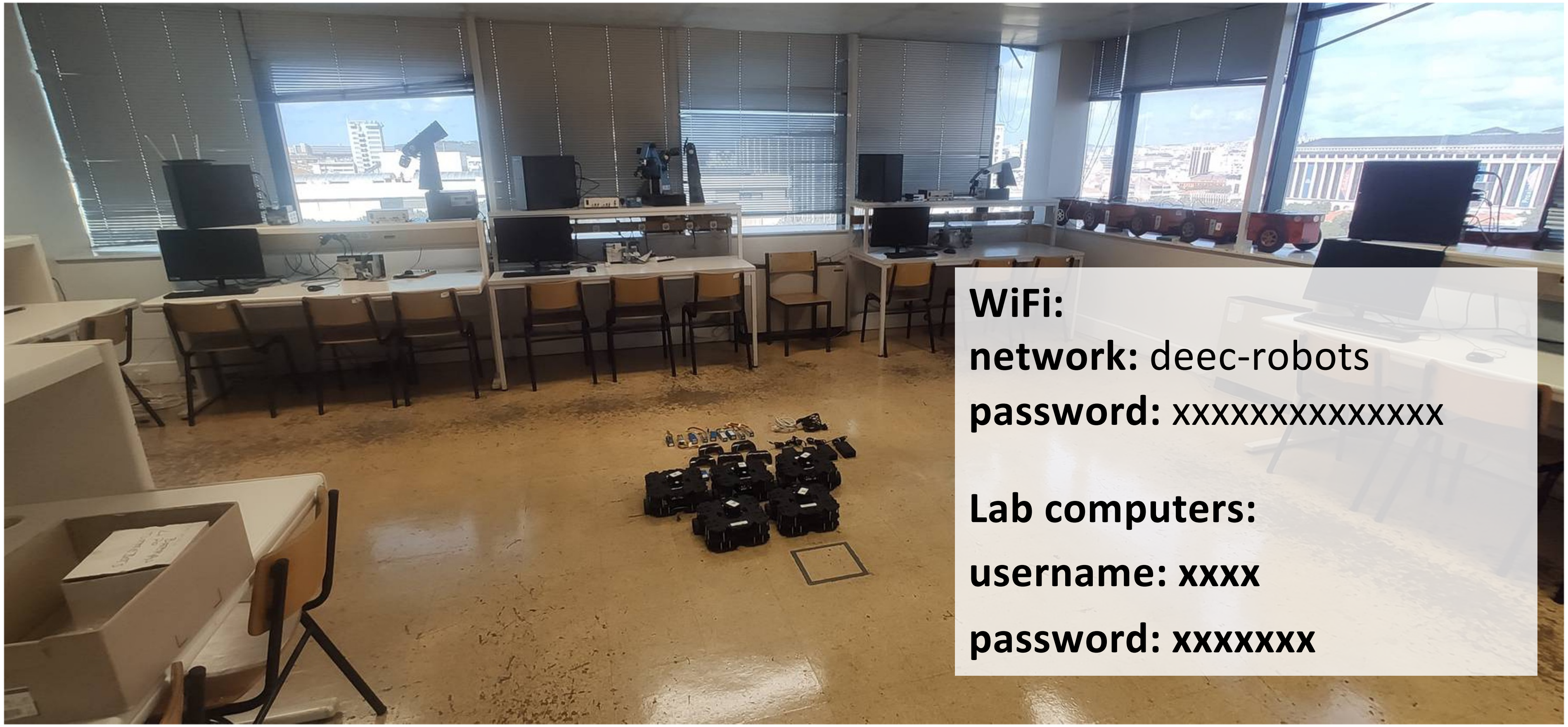
Gazebo tutorials **[recommended]**

- Gazebo Components (mainly SDF files, world, components and variables) https://classic.gazebosim.org/tutorials?tut=components&cat=get_started
- Capture screenshot https://classic.gazebosim.org/tutorials?tut=screenshot&cat=get_started
- Build a robot https://classic.gazebosim.org/tutorials?cat=build_robot
- Model Editor https://classic.gazebosim.org/tutorials?tut=model_editor&cat=model_editor_top
- Building a world https://classic.gazebosim.org/tutorials?tut=build_world&cat=build_world
- Using roslaunch to start Gazebo, world files and URDF models https://classic.gazebosim.org/tutorials?tut=ros_roslaunch&cat=connect_ros

- All tutorials are available here: https://classic.gazebosim.org/tutorials

The Lab and the Turtlebots

LSDC4, 5th floor, North Tower



WiFi:

network: deec-robots

password: xxxxxxxxxxxxxx

Lab computers:

username: xxxx

password: xxxxxxx

Lab desktops

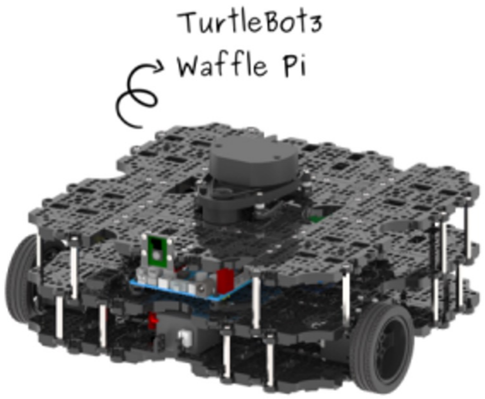
- Lab desktop computers shall be used **to test code** and/or **to record bag files** and **NOT for development**.
- Check the WiFi connection of the lab computers to the WiFi network (if not working properly, please contact Mr. Manuel Ribeiro).
- ROS_HOSTNAME AND ROS_IP have been added as an alias in the lab computers, they can also be exported by simply running EXPORT_TURTLE.
  These export commands are needed in each new terminal that will use ROS to communicate with the robot.

Turtlebot3 Waffle Pi

The Turtlebots are configured with a Raspberry Pi with **Ubuntu 18.04LTS** and **ROS 1 Melodic**

Two modes of operation:

1. **Simulator** (in your laptop)
2. **Real robot** (in the lab)



Resources:

- http://wiki.ros.org/turtlebot3_bringup
- https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/

Turtlebot3 – Simulator

- Install and launch Simulation Package (https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/)

```
cd ~/catkin_ws/src/

git clone -b noetic-devel https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git

cd ~/catkin_ws && catkin_make

source ~/catkin_ws/devel/setup.bash
```

(check ROS extra notes)

- ROS Gazebo and TurtleBot3 packages must be installed first.
  - To install Gazebo see slide 16 (Gazebo). To test if it is already installed, run `gazebo`
  - Install TurtleBot3 Packages

```
sudo apt-get install ros-noetic-turtlebot3-msgs

sudo apt-get install ros-noetic-turtlebot3
```

If having problems with shadows given your graphic card, disable **Scene-> shadows** in Gazebo GUI.

Turtlebot3 – Simulator, have fun with it!

- Launch Simulation World with waffle_pi (the first time takes a while)

```
export TURTLEBOT3_MODEL=waffle_pi
roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

- Do not close gazebo window and teleoperate the Turtlebot3:

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

- Run Rviz (explore rviz)

```
rosrun rviz rviz
```

and add topics (e.g. /odom, /scan)

save the configuration file to use next time

Turtlebot3 – Simulator, have fun with it!

- To watch **odom** data, add “Axes” and select “odom” as Reference Frame.
- To watch the **laser** data, add “LaserScan”, update “Size (m)” to 0.05 and in “Global Options”, write “base_scan” in the “Fixed Frame”.
- If the base_scan frame does not exist,

  start by checking existing tree of transforms between frames (TF Tree)

  ```
  rosrun rqt_tf_tree rqt_tf_tree
  ```
- Add required transform using

  ```
  rosrun tf2_ros static_transform_publisher 0 0 0 0 0 0 base_footprint base_scan
  ```

  or

  ```
  roslaunch turtlebot3_bringup turtlebot3_remote.launch
  ```
- Refresh TF Tree in rqt_tree

  More on TF2 later in the slides.

robot_localization

Source documentation:
http://docs.ros.org/en/noetic/api/robot_localization/html/index.html

- Installation

```
sudo apt-get install ros-noetic-robot-localization
```

- Run

```
roslaunch turtlebot3_gazebo turtlebot3_world.launch

roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

- Download these files: “turtlebot3_localization_sim.launch” and “turtle_localization_sim.rviz”
  https://drive.google.com/file/d/1r-MhG1eeL_UwCm4o0tzNNzrNiOSx5i-_/view?usp=share_link
  https://drive.google.com/file/d/1fwHbZ_xELZxwQFVYXq0JAGlX2oAFJ4yS/view?usp=share_link

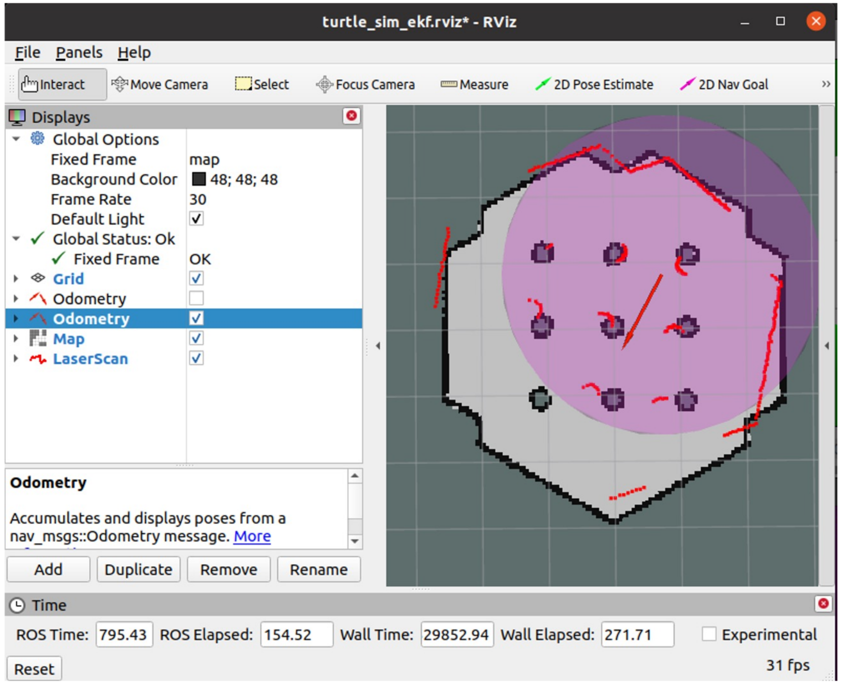
Feel free to edit the launch file and to create your own rviz configuration.

robot_localization (cont.)

- Run

  ```
  roslaunch <pkg_name>
  turtlebot3_localization_sim.launch
  ```

- In Rviz open config and choose “turtlebot3_ turtlebot3_localization_sim.rviz”
- Teleoperate the robot and play with rviz
- See the results
- Odom and IMU are included, but laser is not included in ekf_localization



Global Options
Fixed Frame map
Background Color ■ 48; 48; 48
Frame Rate 30
Default Light ☑
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ☑
Odometry □
Odometry ☑
Map ☑
LaserScan ☑

Odometry

Accumulates and displays poses from a nav_msgs::Odometry message. More

ROS Time: 795.43 ROS Elapsed: 154.52 Wall Time: 29852.94 Wall Elapsed: 271.71 □ Experimental

31 fps

- The launch file includes a command to run rviz. Try to change it to include the configuration file.
- Understanding TF/TF2 is mandatory

TF2 [mandatory]

- To identify the current transformations run:

```
rosrun tf2_tools view_frames.py
```

```
evince frames.pdf
```

or

```
rostopic echo /tf
```

or

use Rviz (configure fixed frame and add all the axes)

or

```
rosrun rqt_tf_tree rqt_tf_tree
```

gmapping

Source documentation: http://wiki.ros.org/gmapping

- Installation

```
sudo apt-get install ros-noetic-gmapping
```

- run

```
roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=gmapping
```

- Save the map to file

```
rosrun map_server map_saver -f my_map
```

- Watch the map

```
rosrun map_server map_server my_map.yaml
```

`rosrun rviz rviz` and add map and change fixed frame to map

- Experiment the parameters and values of the gmapping

amcl

Source documentation: http://wiki.ros.org/amcl

- Installation

```
sudo apt-get install ros-noetic-navigation
```

- Based on https://emanual.robotis.com/docs/en/platform/turtlebot3/nav_simulation/

```
roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=$HOME/map.yaml
```

(instead, you can use the map that you compiled before)

- Take the opportunity to learn more by opening and understanding the 3 launch files

  `amcl.launch`, `move_base.launch` and `turtlebot3_navigation.launch`

  in `/opt/ros/noetic/share/turtlebot3_navigation/launch`

- "amcl" estimates the pose of the robot, "move_base" moves the robot autonomously to reach a goal.
  At this point, we are only concerned with using "amcl".

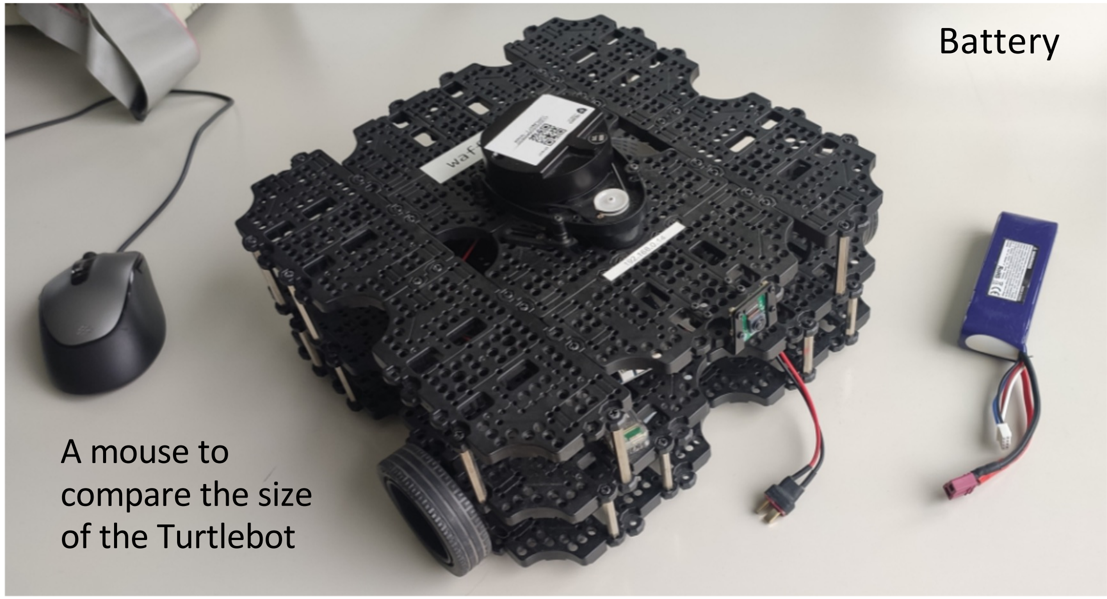
- Write your own launch file to run what is needed for amcl only.
- Experiment with the parameters.

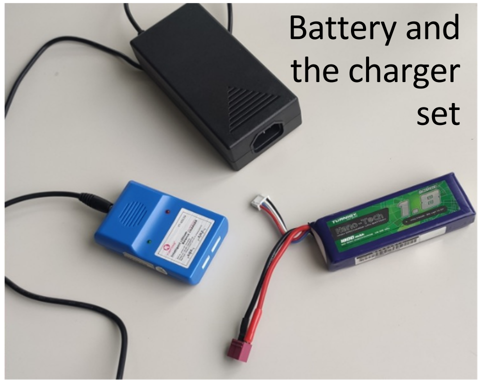
Turtlebot3 – real robot

There are 5 Turtlebots3 Waffle Pi with a numbered label from 1 to 5

Batteries, chargers and gamepads for remote operation are also available at the lab.

After using the equipment return it to the right place.





Turtlebot3 – real robot (cont.)

- Package **turtlebot3_bringup** (see http://wiki.ros.org/turtlebot3_bringup)
  - Subscribed Topics: **cmd_vel** (geometry_msgs/Twist)
  - Subscribed Topics: **reset** (std_msgs/Empty)
  - Published Topics: **odom** (nav_msgs/Odometry)
  - Published Topics: **tf** (tf2_msgs/tfMessage)
- Package **hls_lfcd_lds_driver** (see http://wiki.ros.org/hls_lfcd_lds_driver)
  - Published Topics: **scan** (sensor_msgsTutorials) http://wiki.ros.org/sensor_msgs/Tutorials)
- Monitor (to help!)
  - `rqt`

Turblebot3 – real robot [Laptop Setup]

6. Install the Turtlebot3 packages - run in the laptop once **[skip for lab computers]**.

```
sudo apt-get install ros-noetic-dynamixel-sdk
sudo apt-get install ros-noetic-turtlebot3-msgs
sudo apt-get install ros-noetic-turtlebot3
sudo apt-get install ros-noetic-teleop-twist-keyboard
```

7. Additional configurations. Run in the Laptop/Lab computer (or append to you ~/.bashrc file)

```
export TURTLEBOT3_MODEL=waffle_pi
export TURTLEBOT3_NAME=waffle4                                  [TurtleBot3 363636 on the stick]
export TURTLEBOT3_IP=192.168.28.[11…15]                         [TurtleBot IP]
export TURTLEBOT3_NUMBER=[11…15]                                [Last numbers of the TurtleBot3 IP]
export ROS_MASTER_URI=http://192.168.28.[11…15]:11311           [TurtleBot3 IP]
export ROS_HOSTNAME=192.168.[27/28].XXX                         [lab computer / laptop IP]
export ROS_IP=192.168.[27/28].XXX                               [lab computer / laptop IP]
```

If `ipconfig` is not available, use `ip address` or install net-tools to run `ifconfig`:

```
apt update
apt install net-tools
```

Turblebot3 – real robot [Connecting]

1. On a new terminal of the Laptop/Lab computer, and SSH into the RPi of the **Turtlebot**

```
ssh user@192.168.28.[11…15]
```

2. Start a roscore instance inside the RPi of the **Turtlebot**

```
roscore
```

3. Open another terminal in the Laptop/Lab computer and SSH into the RPi again

```
ssh user@192.168.28.[11…15]
```

4. Sync the robot time **[not mandatory]**

```
sudo apt-get install ntpdate
sudo ntpdate ntp.ubuntu.com
```

5. Launch the robot drivers

```
roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

CREDENTIALS:
Username: xxxx
Password : xxxx
Network: deec-robots
Password: xxxxxxxxxxxxxx
IP range: 192.168.28.[11…15]

Turblebot3 – real robot [Communications]

8. Test the communications: can the computer see the robot’s topics?

```
rostopic list
```

9. Interesting topics to query: “/odom”, “/scan”

```
rostopic echo "/odom"
rostopic info "/odom"
rostopic hz "/odom"
```

10. Test motion commands

```
rostopic pub /cmd_vel geometry_msgs/Twist '[0.0, 0.0, 0.0]' '[0.0, 0.0, 0.5]' (or)
rostopic pub -r 1 /cmd_vel geometry_msgs/Twist '[0.1, 0.0, 0.0]' '[0.0, 0.0, 0.0]'
```

11. To record topics in rosbags

```
rosbag record -a (for all topics)
rosbag record /<topic name> (for specific topics)
```

* If path unspecified, bags get recorded in the folder where the command is run. See http://wiki.ros.org/rosbag/Commandline#record

Turblebot3 – real robot [Operation]

12. Control the robot. On the laptop/Lab computer, run

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

12. Copy a file from the robot to your laptop/Lab computer. On the laptop/Lab computer, run

```
scp <robot_username>@<robot_ip>:<path_to_file> <location_in_your_computer>
```

IMPORTANT

Turtlebot3 – real robot (cont.)

- Now apply in the real robot the “robot_localization”, “gmapping” and “amcl” tested in the simulation!

- **Remember:** one of the strengths of ROS is the simplicity to jump from simulation to real robots.

ROS EXTRA RESOURCES - SUMMARY

- Official ROS website:
  https://www.ros.org/
- ROS Wiki:
  http://wiki.ros.org/
- Core ROS Tutorials:
  http://wiki.ros.org/ROS/Tutorials
  - Beginner Level **[all bullets recommended]**
  - Intermediate Level **[roslaunch tips]**
- TF2 Tutorials:
  http://wiki.ros.org/tf2/Tutorials/
  https://articulatedrobotics.xyz/ready-for-ros-6-tf/
- Robot Model:
  http://wiki.ros.org/robot_model_tutorials
- Visualization:
  http://wiki.ros.org/visualization/Tutorials
- Navigation:
  http://wiki.ros.org/navigation/Tutorials
- Aruco detector:
  https://wiki.ros.org/aruco_detect
- Camera Calibrator:
  http://wiki.ros.org/camera_calibration
- MATLAB [**use ROS to record bags and read them in MATLAB**]
  Open and parse rosbag log file (since R2022a supports ROS Noetic): https://www.mathworks.com/help/ros/ref/rosbag.html