ROB - Advanced robotics

Adam Fabo

Brno University of Technology, Faculty of Information Technology
Božetěchova 1/2. 612 66 Brno - Královo Pole
xfaboa00@fit.vutbr.cz

April 4, 2023

Today's lab will be focused on advanced ROS topics. You will learn:

- Recording and replaying ROS messages
- Visualizing recorded messages
- Showing data from RGB and depth camera
- Running robotic simulations

Recording ROS messages

Recording messages is useful when data collecting process is tedious and takes a long time.

- Messages can be recorded using command line utility rosbag
- Record messages using: `rosbag record -a -O <file_name>.bag`
- Replay messages using: `rosbag play <file_name>.bag`

Record Trilobot messages

Record movement of Robot Trilobot.

- Turn on Triboot and put it on the ground in starting location
- Navigate to **/catkin_ws/src/trilobot_arrows/src/** and run `python3 arrows_control_solved.py`
- Record all ROS messages by running command: `rosbag record -a -O Trilo.bag`
- Put the Pygame window with arrow control in focus and drive trilobot for 10-20 seconds
- End recording of the messages by pressing Ctlr-C in terminal with rosbag process running
- Close Pygame window

Replay Trilobot messages

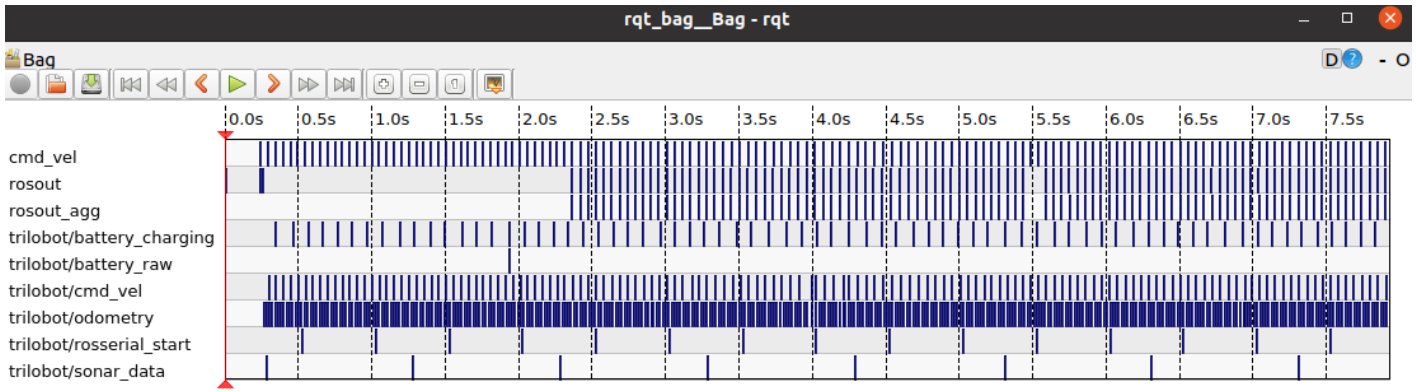
Replay recorded movement.

- Put Trilobot on the place where you started recording
- Replay recorded message by running command `rosbag play Trilo.bag`
- Observe Trilobot replaying previous movement

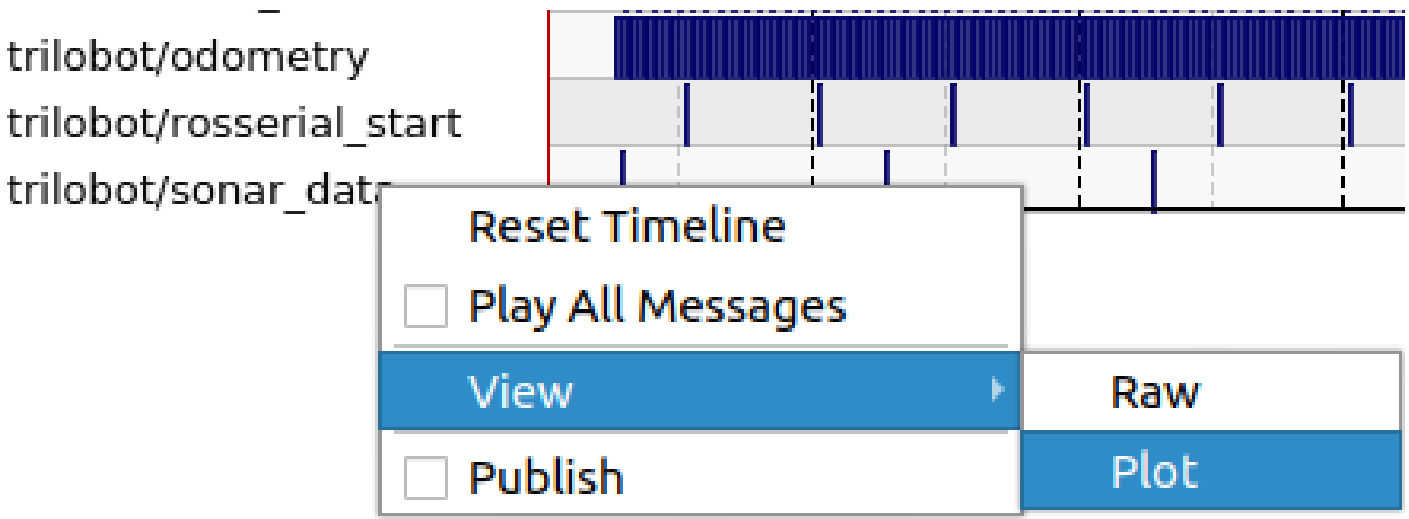
Visualizing recorded messages

Tool rqt_bag offers GUI for visualization of recorded messages.

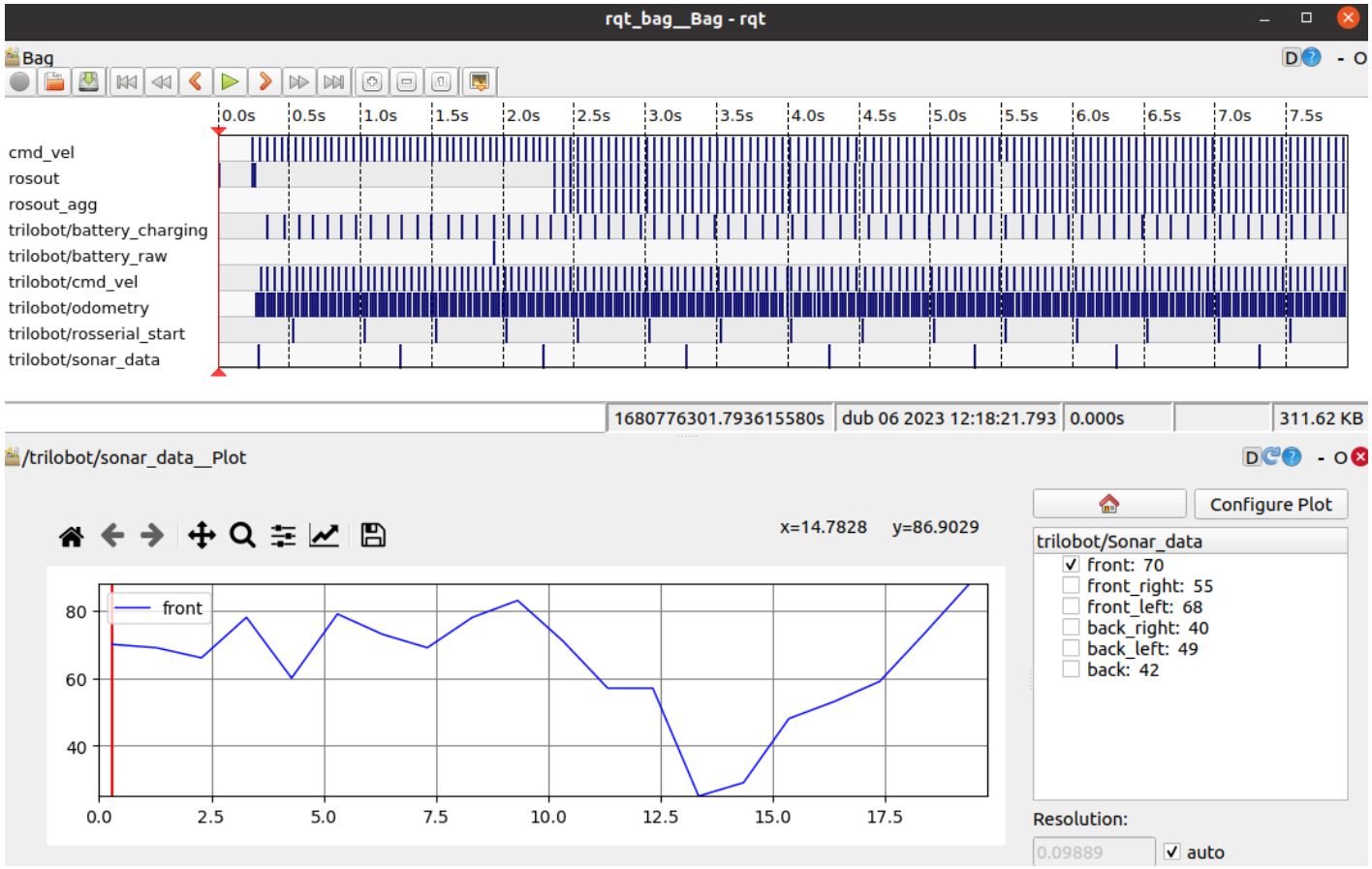
- Recorded messages can be visualized by running _bag <file_name>.bag



- It is possible to plot graph of recorded values
- Right click at topic and selecting View->Plot

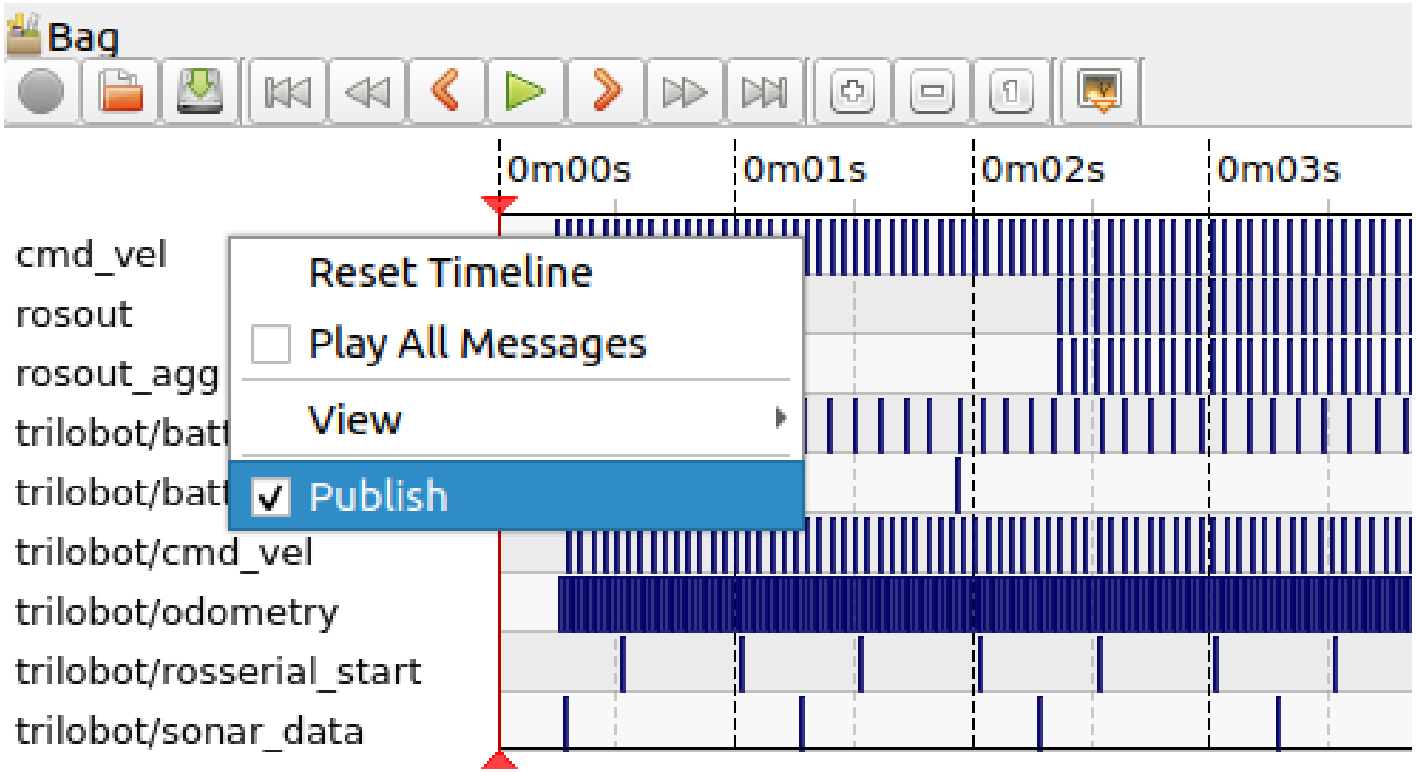


Plotting 2D graphs



Replaying recorded messages

- It is possible to replay recorded messages
- Right click at topic and checking Publish checkbox
- Messages can be replayed by clicking at play button



Visualizing recorded messages

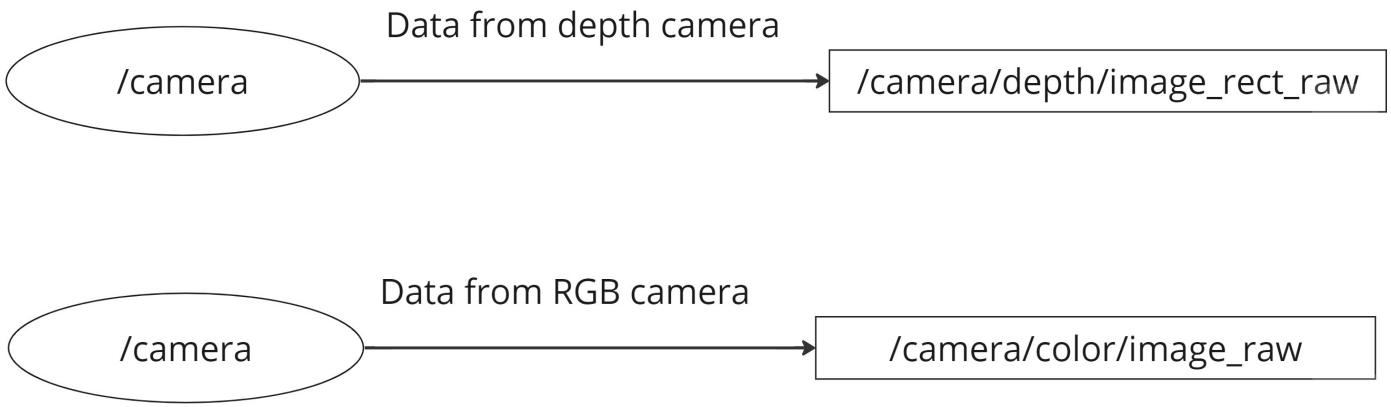
Visualize the gathered data.

- Run command `rqt_bag Trilob.bag`. New window should open up.
- In this window take a look at recorded data and see how messages are sent when controlling Tilobot. Focus on sonar_data and cmd_vel.
- Plot the sonar data into a 2D graph by right-clicking on sonar data topic, and selecting View->Plot. Arrange windows so that graph is visible.
- Replay movement of the Trilobot. Set Trilobot into its starting location. Right click at topic cmd_vel and tick checkbox publish. Replay the messages by pressing spacebar, or play button.

Depth camera

- Robot Trilobot has a camera that behaves like normal camera and like depth camera
- Depth camera measures distance to obstacles and nearest objects

- Data from camera are sent as ROS messages
- Camera published 2 types of messages - one for RGB camera, other for depth camera



Get the data from camera

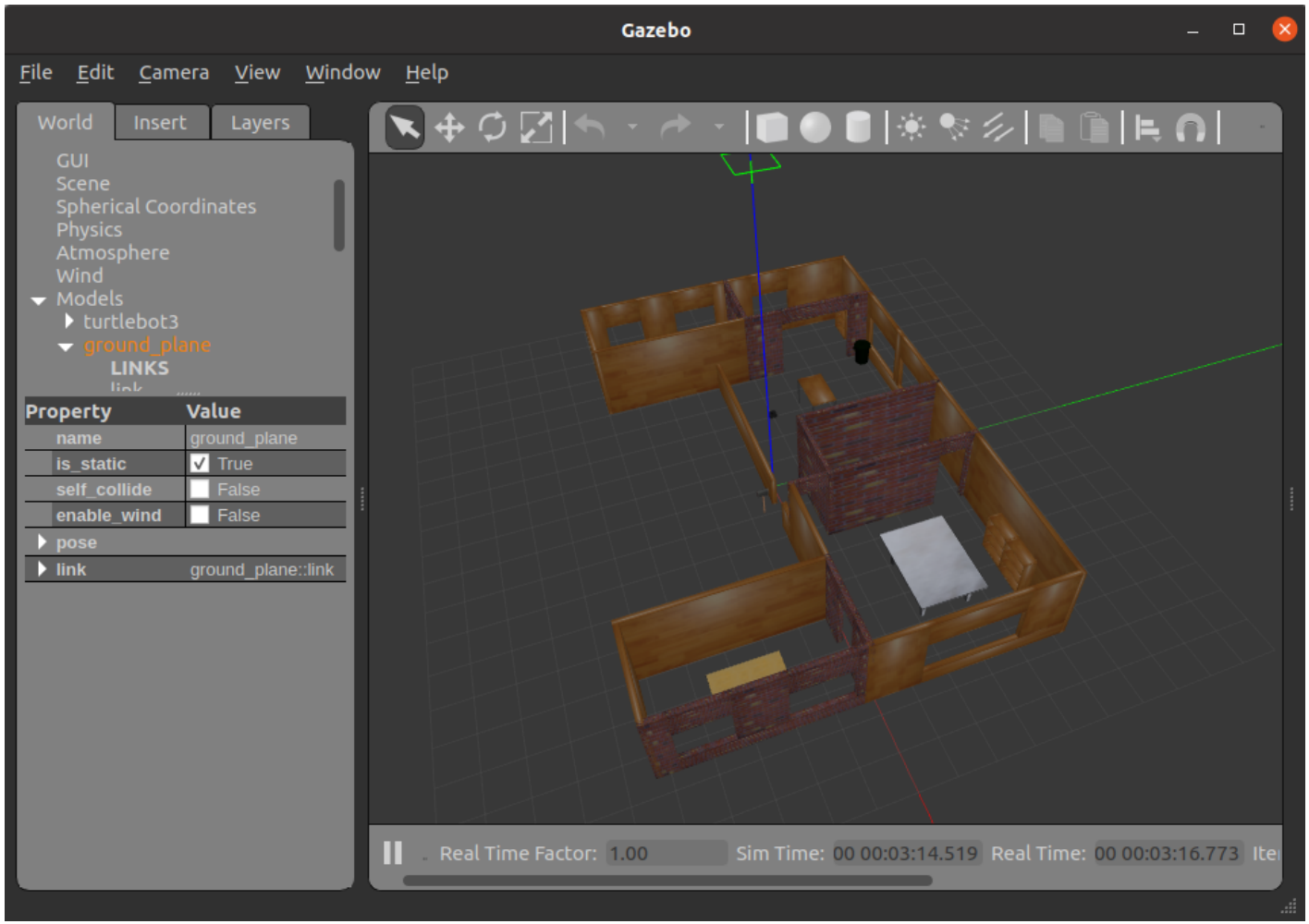
Visualize data from normal and depth camera. Move around room with robot and look at the camera output.

- Run command `rosrun image_view image_view image:=/camera/color/image_raw` to open new window with camera feed.
- Navigate to **/catkin_ws/src/trilobot_arrows/src/** and run `python3 arrows_control_solved.py`
- Navigate robot around the room and look at the camera output
- Close the window with camera visualization and run visualization of depth camera by `rosrun image_view image_view image:=/camera/depth/image_rect_raw`
- Once again, control robot using keyboard and look at the camera output

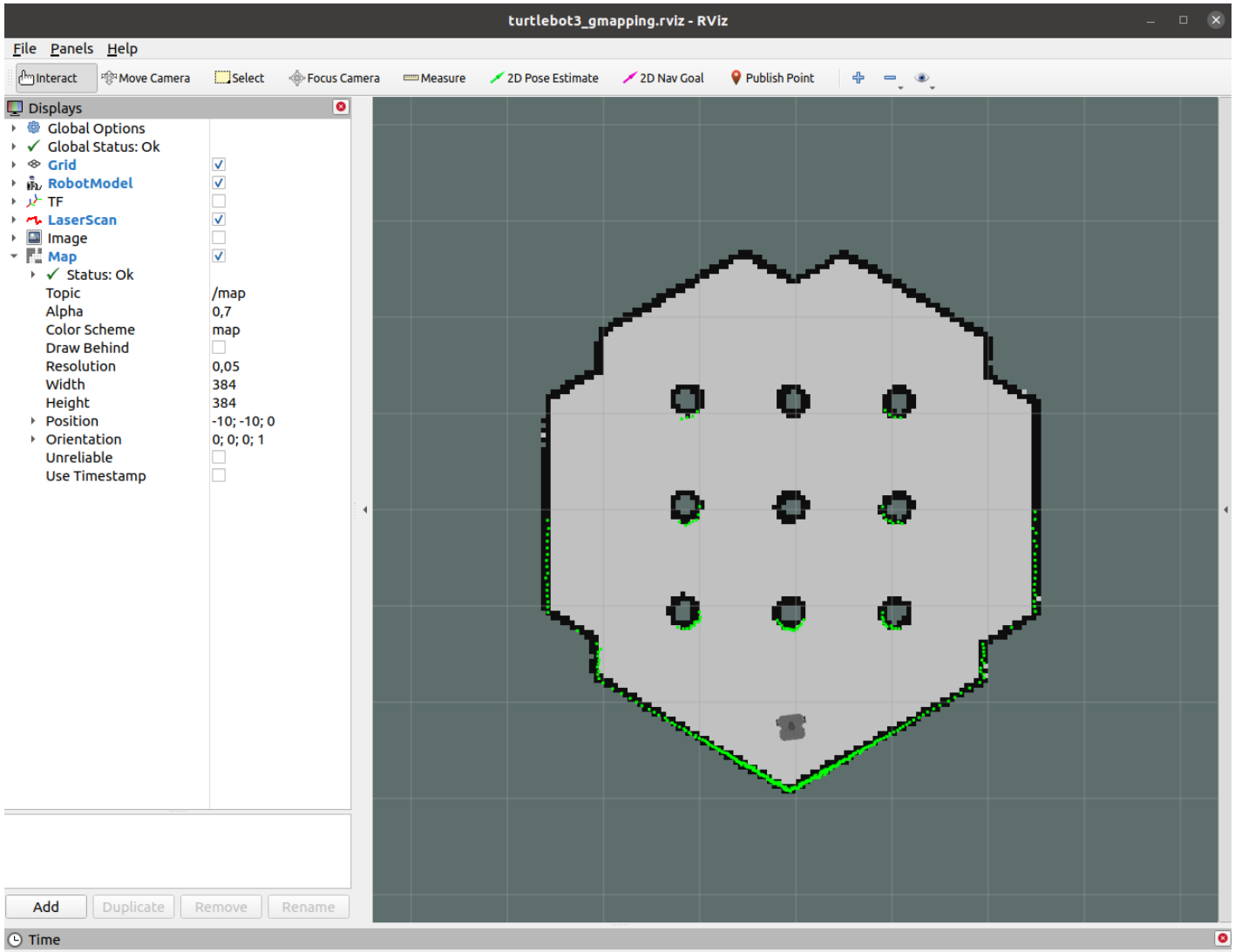
Simulations

- Simulations are huge part of robotics
- They allow for developing robotic applications without building real robot
- Simualation SW used in this lab is Gazebo

- Gazebo is open source 3D robotic software
- It is integrated with ROS
- In this laboratory package Turtlebot3 will be used



- Rviz (ROS-Visualizer) is ROS 3D robot visualizer
- Can visualize data from LIDARs, RGB camera, depth camera ...



Test if your simulation environment is working.

- Close everything from previous exercises and turn off the Trilobot
- Open new Gazebo window with simulation by running command `roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch`
- Run teleop that allows for keyboard control of the robot using: `roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch`
- Control the robot from this terminal OR open Pygame window

Use gazebo simulation for Simultaneous Localization and Mapping. Create a map of the simulated environment.

- Keep everything running as it is
- RUN command `roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=gmapping`
- Control the robot with keyboard and map the whole simulated space
- After whole simulation is mapped, save the map by running `rosrun map_server map_saver -f ~/map`

Use map created in previous exercise for Navigation.

- Close Rviz window
- Run command `roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=$HOME/map.yaml`
- Control robot with keyboard and take a look at the algorithm working
- Change different starting locations by clicking at 2D pose estimate and once again control the robot
- Close Pygame window
- Use navigation goal button to set different locations and observe the robot moving

Thank You For Your Attention !