ROB314 – Session 1 – Ex. 3

Husky and Gazebo

Theory

- ROS architecture
- ROS master, nodes, and topics
- Console commands
- Catkin workspace and build system
- Launch-files

Exercise

Get to know ROS by inspecting the simulation of a Husky robot.

1. Preparation:
```
sudo apt update
sudo apt install ros-melodic-position-controllers ros-melodic-effort-controllers
ros-melodic-joint-state-controller
```

2. Test Gazebo:
```
gazebo
```

   To stop the program: Ctrl + C in the terminal. It can take a while...

3. Setup the Husky simulation
```
sudo apt-get install ros-melodic-husky-simulator
```

4. Launch a simulation, for example :
```
roslaunch husky_gazebo husky_playpen.launch
```
It will take some time to start, as the simulator will need to download resources from the Gazebo servers.

   You can move the visualization with the left mouse button or shift + left mouse button.

   And inspect the created nodes and their topics using
```
rosnode list
rostopic list
rostopic echo /a_nice_topic (for example /imu/data, /navsat/fix, etc.)
rostopic hz /a_nice_topic
rqt_graph
```
   For more information take a look at the slides or:
   http://wiki.ros.org/rostopic
   http://wiki.ros.org/rosnode

5. **Have a look at the file** `husky_playpen.launch`. To find this file, you can use the command `roscd husky_gazebo`.

   Have a look to others launch files included in `husky_playpen.launch`

6. **Command a desired velocity** to the robot from the terminal (`rostopic pub [TOPIC]`)

   Hint 1: You can use this topic */husky_velocity_controller/cmd_vel*

   Hint 2: If the robot stops again after sending the velocity command, specify the rate of the publisher. Check out `rostopic pub –help`.

7. Create a **catkin workspace** *catkin_ws* as described in the lesson.
8. Use **teleop_twist_keyboard** package to control your robot using the keyboard. Find it online (github) and compile it from source!
   Use `git clone` to clone the repository to the folder ~/catkin_ws/src.
   Compile with the dedicate ros tool…
   For a short git overview see:http://rogerdudler.github.io/git-guide/files/git_cheat_sheet.pdf

   Don't forget to do a:
   ```
   source ~/catkin_ws/devel/setup.bash
   ```

9. Create a new directory "launch" in *~/catkin_ws/src/teleop_twist_keyboard/*.
   Inside, write a launch file, called for example *husky_robocup.launch* with the following content :
   ○ With **ROS melodic**, husky simulation with a different world:
      Include *husky_empty_world.launch* file and change the *world_name* argument, e.g.
      `/usr/share/gazebo-9/worlds/robocup14_spl_field.world`

   ○ With **ROS noetic**, husky simulation with a different world:
      Include *empty_world.launch* file and change the *world_name* argument, e.g.
      `/usr/share/gazebo-11/worlds/robocup14_spl_field.world`

   ○ `teleop_twist_keyboard.py` node