ROB314 – Session 2 - Exo1

Theory

- ROS package structure
- Integration and programming
- ROS C++ client library (roscpp)
- ROS subscribers and publishers
- ROS parameter server
- RViz visualization

Exercise

In this exercise, you will create your first ROS package. The package should, in the end, be able to subscribe to a laser scan message from the Husky robot and process the incoming data. This node will be the basis for the next exercises.

Make sure to look at the ROS template for reference https://github.com/leggedrobotics/ros_best_practices It will help you a lot for the implementation, as it has a similar node to what you have to do in this exercise!

1. Download the Zip archive containing prepared files of the package `husky_highlevel_controller` at this adress : https://perso.ensta-paris.fr/~battesti/download/husky_highlevel_controller.zip

2. Install the package in your catkin workspace :
   unzip the folder in catkin_ws/src
   ```
   cd ~/catkin_ws
   catkin_make
   source ~/catkin_ws/devel/setup.bash
   ```

3. Inspect the `CMakelists.txt` and `package.xml` files.
4. Inspect the source code.
   For the moment, the node doesn't do anything.

5. Create a subscriber to the `/scan` topic, this topic contains the laser scan message from the Husky robot. Find inspiration in the *ros_best_pratices* package, but keep the subscriber's callback empty.

6. Add parameters with topic name and queue size for the subscriber of the topic /scan.

7. Create a callback method for that subscriber which computes and outputs, to the terminal, the smallest distance measurement from the vector *ranges* in the message of the laser scanner.
   Inspect the message type here :
   http://docs.ros.org/kinetic/api/sensor_msgs/html/msg/LaserScan.html.
   Tips: To filter some "NaN" value, you can use the function std::isnormal()

8. Add your launch file from “Session 1 - Exercise 3” and additionally add:
   - running the `husky_highlevel_controller` node.
   - loading the two parameters with *param* tag

9. Pass the argument `laser_enabled` from your launch file to the `husky_empty_world.launch` file with value `true`.

10. Show the laser scan in RViz with the plugins “LaserScan” (click *add* at the bottom left of the window) .

11. Add RViz to your launch file.

12. Make sure to set *odom* as the *Fixed Frame* (under Global Options) and adapt the size of the laser scan points. You can save your current RViz configuration as the default configuration by pressing ctrl+s.