

CSC752: Autonomous Robotic Systems, Fall 2020

Assignment 2

This assignment is worth 20 points and it is due on or before September 09, 2020. You will create your first ROS package. You will implement a program that subscribes to a laser scan message from the Husky robot and processes this data. We suggest to refer to the ROS best practices available at <http://wiki.ros.org/BestPractices>. There is a lot of help with regard to conventions, units etc. They also provide a (C++) template you can use as a starting point.

1. Create the package `husky_behavior_controller` from scratch. Use the command `catkin_create_pkg` to create a new package with the dependencies `roscpp` (or `rospy` if you prefer to program in Python) and `sensor_msgs`. Refer to the tutorial "Creating a ROS Package" if you are stuck: <http://wiki.ros.org/ROS/Tutorials/CreatingPackage>.
2. Check the `CMakeLists.txt` and `package.xml` files.
3. Create a subscriber to the topic `/scan`.
4. Add a parameter file with topic name and queue size for the subscriber of the topic `/scan`.
5. Create a callback method for that subscriber which delivers the smallest distance measurement from the vector ranges in the message of the laser scanner to the terminal. The message type is defined as: http://docs.ros.org/kinetic/api/sensor_msgs/html/msg/LaserScan.html. You might want to launch the empty Husky world with `roslaunch husky_gazebo husky_empty_world.launch` and then add an object into Gazebo to get a distance measure.
6. Add your launch file from Assignment 1 to this package and modify it:
 - Run the `husky_behavior_controller` node.
 - Load the parameter file
7. Pass the argument `laser_enabled` from your launch file to the `husky_empty_world.launch` file with value `true`.
8. Run RViz and visualize the laser scan. Add RViz to your launch file. You need to change the *Fixed Frame* (under Global Options) and set it to `odom`. Also, you need to adapt the size of the laser scan points. You can save your RViz configuration as the default configuration for the future.
9. Submit the following files by committing them to the SVN into your users directory:
Source code of your package.