Fall 2021 RIS Lab I project sheet

You will be using some of the uuv_simulator packages for this project. Run the commands below to install the packages.

Melodic:

- sudo apt install ros-melodic-uuv-simulator

Noetic:

- roscd
- cd ../src
- git clone https://github.com/my-name-is-D/uuv_simulator.git
- cd ..
- catkin build

You will then likely run into some more errors about missing packages when you try to run the launch files. You can solve these by simple finding the name of the package in the error message, finding it's repository on github and cloning it, similar to how you cloned the uuv_simulator packages.

Please then complete the tasks below and submit your solutions by end of 21.11.2021. Please write the % contribution of each group member next to each exercise.

1. Launch uuv_gazebo/rexrov_default.launch and inspect the nodes, topics, services it launches and the message/service types they use to communicate.
   a. Write a short report describing these nodes, topics, services and messages including screenshots of rqt or the terminal output from which you collected this information. **(15 pts)**
   b. Include the commands you used to retrieve this information and write down what these commands do. **(10 pts)**
2. Controlling the robot using teleop_twist_keyboard:
   a. Write a launch file that launches the rexrov in uuv_gazebo/rexrov_default.launch into the world spawned by uuv_gazebo_worlds/empty_underwater_world.launch and the teleop_twist_keyboard node in such a way that you can use the teleop node to control the simulated ROV. Please include comments in this launch file to explain what each line is doing. **(15 pts)**
   b. Log the topic 'teleop_twist_keyboard/cmd_vel' using rosbag, move the vehicle around with your keyboard while rosbag is recording. Stop teleop_twist_keyboard and control the robot by replaying the generated rosbag. Create a plot of the traced trajectory of the ROV, and attach a screen shot of rqt_graph while the bag is playing **(5 pts)**
3. In this exercise you need to create a node that outputs forces and torques as input to control the AUV in uuv_gazebo/rexrov_wrench_control.launch based on this input. Please download this launch file from the page linked

into catkin_ws/src/uuv_simulator/uuv_gazebo/launch/rexrov_demos and launch it into the world spawned by uuv_gazebo_worlds/empty_underwater_world.launch

   a. Write a node similar to teleop_twist_keyboard that publishes forces and torques to control the AUV. Your node must publish to the topic /rexrov/thruster_manager/input **(15pts)**
   b. Write a launch file similar to the one in exercise 2 that launches the AUV and your node. **(5 pts)**

4. Creating and controlling your own robot.
   a. Create a urdf file that describes a simple ROV of your own design using the default geometric shapes (or design your own robot in Blender) **(10 pts)**
   b. Decide on the placement of thrusters for this ROV and give your reasoning. **(5 pts)**
   c. Write your own node that takes in forces and torques as input and publishes thrust commands for your ROV. The conversion from forces/torques to thrust commands has to be called as a service. **(10 pts)**
   d. write a launch file that **(10 pts)**
      i. load this robot into uuv_gazebo_worlds/empty_underwater_world.launch in gazebo
      ii. starts the node you created in exercise 3 to publish force/torque data
      iii. starts the node you created in part c that publishes thrust commands to your vehicle