Robotic Manipulation Course

Exercise 1

January 14, 2024

Preliminaries

Instruction to ROS and git are available on slides and ROS tutorials.

1 Before starting

You need to do this for every exercises. Follow the steps below before starting the assignment:

1. For the Gitlab repository, we created one **subgroup** for each one of the student. You can use the following pattern to access that:

   `https://version.aalto.fi/gitlab/robotic_manipulation_students_2024/<email_address_without_@aalto.fi>`

   for example if your email address is `daulet.baimukashev@aalto.fi` use:

   `https://version.aalto.fi/gitlab/robotic_manipulation_students_2024/daulet.baimukashev`

2. **Fork** every exercise (for example exercise 1 at https://version.aalto.fi/gitlab/robotic_manipulation_2024/exercise1) into **your subgroup**.

3. Now, you are ready to clone the forked repository (in your subgroup) and start modifying the code to solve the assignment.

4. **THIS IS EXTREMELY IMPORTANT!** Whenever you make changes in your ros workspace, you need to compile everything with the command

   `catkin_make`

2 Assignment

The goal is to periodically (25 Hz) read transformation from tf tree and publish the pose of frame *lumi_ee* with respect to the frame *base_link* to the topic */tf_pose* using *c++* template we provided to you via gitlab. We created a C++ template node `ros_intro.cpp` in the course gitlab group. Your goal is to modify that node to perform following steps:

- read the pose from tf tree
- publish it at a frequency of 25 Hz to a topic called */tf_pose*.

To test your code you need to launch the simulation:

```
roslaunch lumi_description show.launch
```

and than run your node with

```
rosrun exercise1 frame_publisher
```

and than check the topic you publish:

```
rostopic echo /tf_pose
```

Hint: You can now change the position of the robot using the sliders and check if the published information in the topic is reasonable. The default position of the end effector by default is: x: 0.5544, y: -3.328e-12, z: 0.508.

The same commands will be used by TAs for testing. Therefore, do not change the package/node names.

3 Report

No report is needed for this exercise.

4 Submission

The solution needs to be pushed to your repository in your own subgroup. Be sure to push your code before the assignment deadline. **Commits pushed after the deadline will be subjected to penalties according to the course's practicality**.

For GitLab users who simply want a quick overview of the commands to run in the existing project to push to GitLab, here they are.

- git add .
- git commit -m "Push existing project to GitLab"
- git push -u origin master

5 Deadline

Deadline for this assignment is 23rd of January at 23:59.