Robotic Manipulation

Exercise Practicalities

Jens Lundell Vladimir Petrik

Exercise session overview

- Exercise sessions every Wednesday 08:15-10:00 and Thursday 10:15-12:00 via Zoom.
- Voluntary presence.
- Teacher assistants (TAs) present to help you.

Requirements

- A Linux system (we provide and support Ubuntu 18.04 with installed ROS on that)
- ROS Melodic (we support this version) `http://wiki.ros.org/melodic/Installation/Ubuntu`
- catkin tools `https://catkin-tools.readthedocs.io/en/latest/installing.html`
- MoveIt! for Melodic `https://moveit.ros.org/install/`
- The MuJoCo simulator *mujoco200 linux* `https://www.roboti.us`.
- The MuJoCo license found in MyCourses under "For Aalto users".
- You can find a virtual box image on MyCourses under "Assignments".

Communication

- The preferred means of communication is the course slack channel (separate email for registration has been sent to all course participants) and not email.
- When you sign up to the slack workspace use your **aalto username** as the **Nick name** as we will link this to your gitlab repository.
- If you have specific problems with your code do not send it over email or slack. Instead, tell us that you have some problems and push your latest commits to the gitlab repository and we will pull it from there and start investigating.
- Usually more than one student have similar problems and thus we advocate asking questions in the exercise slack channels to enable students to help each other out.
- TAs and the lecturer will, per default, not answer emails or slack messages during the weekends.

Exercises

- In total six problems:

Exercises

- In total six problems:
  1. Introduction to ROS.

Exercises

- In total six problems:
  1. Introduction to ROS.
  2. Planning algorithms benchmark in MoveIt

Exercises

- In total six problems:
  1. Introduction to ROS.
  2. Planning algorithms benchmark in MoveIt
  3. Simple pick and place with MoveIt

Exercises

- In total six problems:
  1. Introduction to ROS.
  2. Planning algorithms benchmark in MoveIt
  3. Simple pick and place with MoveIt
  4. Force control

Exercises

- In total six problems:
  1. Introduction to ROS.
  2. Planning algorithms benchmark in MoveIt
  3. Simple pick and place with MoveIt
  4. Force control
  5. Two-finger grasp planning

Exercises

- In total six problems:
  1. Introduction to ROS.
  2. Planning algorithms benchmark in MoveIt
  3. Simple pick and place with MoveIt
  4. Force control
  5. Two-finger grasp planning
  6. Dual-arm (hybrid control)

Tentative exercise schedule

The exercises are introduced on the following exercise sessions

- 12th of January intro to exercise 1
- 26th of January intro to exercise 2
- 9th of February intro to exercise 3
- 23rd of February intro to exercise 4
- 9th of March intro to exercise 5.
- 23rd of March intro to exercise 6

Exercise Deadlines are stated separately in each assignment PDF.

Submissions

- Each solution include at least source code written in C++ that solves the problem and for most of the exercises you also need to submit a report written in English.
- The report (saved as PDF) should answer the questions posed in the assignment which can, for example, be the following
  1. Which planner is fastest and why? Plot the running times and answer the questions by comparing the algorithms.

Submissions

- Each solution include at least source code written in C++ that solves the problem and for most of the exercises you also need to submit a report written in English.
- The report (saved as PDF) should answer the questions posed in the assignment which can, for example, be the following
  1. Which planner is fastest and why? Plot the running times and answer the questions by comparing the algorithms.
- If the exercise requires you to submit both a report and code you need to submit both. Otherwise, your solution will be rejected and you will be awarded with 0 points.

Submissions

- Each solution include at least source code written in C++ that solves the problem and for most of the exercises you also need to submit a report written in English.
- The report (saved as PDF) should answer the questions posed in the assignment which can, for example, be the following
  1. Which planner is fastest and why? Plot the running times and answer the questions by comparing the algorithms.
- If the exercise requires you to submit both a report and code you need to submit both. Otherwise, your solution will be rejected and you will be awarded with 0 points.
- Each student forks the exercise into their own gitlab group, solves the exercise there, and finally upload everything to the respective repository before deadline

Grading

- If the exercise requires both code and a report the report accounts for 50% and the code for 50% of all points awarded for that exercise. Otherwise, one or the other accounts for 100% of the points.
- The code is graded based on correctness (0-100%).
- The report is graded based on:
  - Correctness (0-100%),
  - How well it is written (satisfactory, good, excellent).
  - We will grade both language and structure of the report. For example, the report should be easy to read and coherent, and you need to refer to all figures, tables etc. Think of every single report as a part of your future MSc. thesis.
- TAs will push, to your gitlab repository, the grade and feedback for the given exercise.

Exercise rules

- Exercises are handed in and done **individually**
- You are allowed to discuss the problems but not share solutions.
- No copying of exercises (neither code nor report). If we notice plagiarism it is reported and consequences follow.
- No late submissions are accepted.