Robot Programming

Project and Exam Rules (2024-25)

Giorgio Grisetti

About Exam

- The evaluation is binary (pass/no pass)
- It consists of
  - A multi-answer questionnaire
  - Homeworks OR Project

Exam: Questionnaire

- Questionnaire to be done **after** submitting the project/homework
- 10 Multi-answer questions
- Each question is worth 3 points
- The sum of points of each question is 0: the wrong answers are rewarded negatively
- There might be multiple correct answers, to get the maximum score you have to tick them all

- Too many wrong answers **invalidate** the project/homeworks (<0)
- The score to pass is 30% (+10 points)

Homeworks

- 3 homeworks, to be submitted through GIT/Classroom
- You have to pass 2 out of 3 of them
- **Each individual commit must be tagged**
- **Push on remote only when you are done**

  (in git the local commits are retained in the history)

Project Guidelines

- Written in C++
- Builds with ament on ROS2

- Can be integrated in other subjects (e.g. Robotics/ AI/ DL and so on)

Project Examples

Dynamic Planner

- The planner subscribes to the laser scan and computes a (local) distance map in the neighborhood of the robot each time a new scan is received
- This distance map is integrated in the global distance map of the planner, and the costs updated

  (distance map can be composed by taking the minimum)
- The planning algorithm uses A* with the cost of Dijkstra on the empty map as heuristic
- The output should be integrated in ROS and work with the navigation stack.

Project Examples

Simple RVIZ

- Write a program to control a mobile robot in ROS and display simple systems(rviz simple clone).
- The program is able to show
  - a map (received from the map server),
  - Laser scans
  - mobile bases (as a circle)

    All items are displayed according to their transform
  - particles of localization
- The program is able to issue
  - issue /initialpose message, to initialize the localizer
  - and /move_base/goal messages to set a planner destination

Project Examples

Inverse Kinematics of a robotic ARM

- Develop a ROS node that simulates a robotic arm
- The arm is controlled by joint positions AND it implements the inverse kinematics (using the pseudo-inverse method on a numeric jacobian)
- The system reads the Denavit-Hartenberg parameters from a configuration file

Project Evaluation

- During the discussion we will ask you to do small modifications of your work on the fly. Not succeeding in doing so means you did not understand what you wrote and raises questions on the project's authorship.
- If the above happens, your project will be changed, and you will come back at the next round.
- The project evaluation takes place in the month before the questionnaire date.