Robot Programming

Project and Exam Rules

Giorgio Grisetti

Part of the material in these slides is taken from the Robotics 2 lectures given by G.Grisetti, W.Burgard, C.Stachniss, K.Arras, D. Tipaldi and M.Bennewitz

About Exam

- The evaluation is binary (pass/no pass)
- At each session we will establish some discussion dates. You will reserve a slot in these dates through a web system
- The exam consists in the presentation of the project

Project

- Done in C++
- Done on a private repo (gitlab/github) shared with the teachers of the course
  grisetti@diag.uniroma1.it
  digiammarino@diag.uniroma1.it
  giacomini@diag.uniroma1.it
  salem@diag.uniroma1.it

- **Each individual commit must be tagged.**
- A project can be done in small groups (max 3). The blocks of code written by each member should be tagged in the source as comments with the member initial (e.g. //gg: <your comment>)
- A project should be accompanied by a small report (2/3 pages) stating:
  - how to compile
  - How to run
  - How to test

Project Guidelines

- Written in C++
- Builds with
  - Cmake if on PC/no ROS
  - Catkin if on PC/ROS
  - Make if on AtMega
- Can be integrated in other subjects (e.g. Robotics/ AI/ and so on)

Project Examples

- **Own Robot[1 ppl]**: Build my own robot and integrate it in ROS – firmware made from scratch
- **2D Lidar Matcher[1 ppl]:** Implement a 2D scan matcher (see ICP example), working with 2d laser scans (integrated in ROS)
- **2D ICP Localizer [1 ppl]** Implement a 2D scan matching based localization (map is a set of 2d points)

  (integrated in ROS)
- **Normal based matcher [2 ppl]**
  - Extract the normals from a 2D scan, and produce a new data type (ROS)
  - Implement a node that constructs a 4D kd_tree with points + normals, and runs ICP with normals (math provided by us). Outputs a tf.

Project Examples

- **Obstacle Avoidance [no labiagi][1 ppl]**: Build a node that computes a repulsive field from the local laser scans and modulates the /cmd_vel so that the robot does not clash to the wall if instructed to do so.
- **Multi Robot Navigation [no labiagi][2 ppl]**: Build a node that monitors N robots navigating in the environment. (includes simulator setup)

  Allows to:
  - Show the map
  - show the position of each robot,
  - to relocalize them (via mouse)
  - Give goals

Project Examples

- **2D Simulator [1 ppl]**: Build a node that implements a 1 robot simulator. Reads the map from the map_server. Reads cmd_vel, produces */odom /tf and /base_scan*

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 author.

- If the above happens, your project will be changed, and you will come back at the next round.