Robot Programming

Project and Exam Rules

Giorgio Grisetti

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

- **Compile the form to request the project**
- Done on a private repo (gitlab/github) shared with **all** the teachers of the course

  grisetti@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 2). 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 (1 page) stating:
  - how to compile
  - How to run
  - How to test

Project Guidelines

- Written in C++
- Builds with catkin on noetic or melodic
- Can be integrated in other subjects (e.g. Robotics/ AI/ DL and so on)

Project Examples

2D Multi Robot ROS simulator [1 ppl]

Write a C++ configurable multi robot simlator supporting

- Grid maps
- Unicycle like mobile robots
- Laser Scanners

The system should read the configuration from a text file specifying the layout of the simulation environment:

- For each device : the frame_id, and the topics. If a device is mounted on another device, the position in w.r.t the parent device.
- For a lidar the number of beams and the max/min range
- For a robot the maximum velocities

The output should be integrated in ROS and work with the navigation stack.

Project Examples

2D Normal-Based Lidar Odometry [1ppl]

Write a ROS node that listens a sensor_msgs/LaserScan message.

Based on this, implement a Normal-Aware Lidar odometry that estimates the ego-motion of the robot

Instantiate a KD-Tree for each scan for the correspondece search and adjust the program so that it works in a scan-to-keyframe fashion. Similar to the ICP case, we provide the optimizeCorrespondence() routine.

The program should provide a consistent TF-tree (similar to the one developed in class)

Project Examples

ICP based localization [1ppl]

- Write a program that listens a Grid-Map, and extracts the obstacles (occupied cells), and populates a KD-tree with the world coordinates of these points.
- The program should output a transform between the map and odom, similar to what the localizer does.
- The program works by computing the transform between the current scan and the map, using ICP.
- It subscribes to a topic /initialpose to set the initial position from rviz
- Each time a new scan is received the program performs a new registration, starting from an intial guess. This initial guess is obtained by applying the odometry **displacement** to the estimate of the previous time instant.

Project Examples

Simple RVIZ [1ppl]

- 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 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.