Mestrado em
Engenharia Informática e de Computadores
MSc in
Information and Computer Engineering

INTRODUÇÃO À ROBÓTICA / INTRODUCTION TO ROBOTICS

2019/2020

Mini-Project 1

Hand-out: 23 September 2019 **Due:** 1 November 2019

Objective

The objective of this homework is to provide the course students with the opportunity to get familiarized with the practical aspects of mobile robot localization, using Bayesian filters to handle the associated uncertainty in perception and action effects. For this purpose, students must get acquainted with some of the available ROS packages for mapping and self-localization (particularly the navigation stack), learning how to use them, and being able to explain formally their operation principle. Students must also get used to saving all the relevant data to be reported using `rosbags`.

Procedure

The work will be implemented in a TurtleBot3 Waffle Pi[1] mobile robot. The robot has an onboard laser scanner, used to acquire the environment map and to self-localize, and a Raspberry Pi processor where all ROS drivers are running. The algorithms implemented in the mini-project will run on an external computer that communicates with the onboard ROS master using WiFi. A dataset with a map, laser scan and odometry readings, taken from the real robot in a given scenario, as well as the ground truth for the actual robot path (obtained from a Motion Capture System) will also be provided to be used in some parts of the work.

The main steps to be followed to achieve the objectives of the project are:

1. Use ROS `ekf-localization`[2] with the map, odometry and laser scanner readings **from the dataset** to implement and test robot self-localization by estimating the traversed path, using **Extended Kalman Filter (EKF-)based localization**.
2. Use ROS `gmapping` (ROS package `gmapping`[3]), odometry and laser scanner readings **from the real robot** to map the area (e.g., in LSDC4) where the robot will have to self-localize later, and store the obtained map.

1 http://www.robotis.us/turtlebot-3-waffle-pi/

2 https://github.com/guilhermelawless/ekf_localization/tree/master - modified from the original ROS package at https://github.com/vislab-tecnico-lisboa/ekf_localization

3. Use ROS `amcl`[4] with the map from 2., odometry and laser scanner readings **from the real robot** to test robot self-localization by estimating the traversed path, using **Monte Carlo Localization (MCL)**. Drive your robot manually (by teleoperation) through some path, along which the robot should be able to self-localize.

Expected results

The following list represents the minimal set of results to be reported:

- maps (provided with dataset and estimated from the real robot) - in RVIZ;
- robot paths (estimated by the robot and estimated from the dataset) for MCL and EKF-based localization, respectively – in RVIZ;
- ground-truth path (for the dataset only) – in RVIZ;
- error vs time (mean value and $\pm 2\sigma$ bounds) of estimated path with respect to ground-truth path (from the dataset) for EKF-based localization.

The groups are strongly encouraged to explore and modify relevant parameters of the two methods, e.g., measurement noise and process noise models, number of particles), so as to be able to present a diverse set of results and justify the differences among them as a function of the parameters used.

The tele-operated paths should be such that the localization problem is not solved trivially everywhere along them. This choice will be subject to evaluation. Forcing robot-lost situations to illustrate kidnapping recovery will be rewarded in the evaluation of the mini-project.

Reporting

The mini-project will be evaluated taking into account the quality of reporting on the work done, the results presented in class and in the written report, and the ability to explain them formally. The written report should be no longer than 10 pages A4, one column, 12pt, 1.5pt spacing. All reports shall be structured in Sections as follows:

1. Introduction – *summarizing the work done*
2. Background – *brief introduction of main equations and algorithms involved in the used methods, to establish notation, citing references used*
3. Implementation – *core section, where the implementation of the applied methods is described with some detail*
4. Results – *addressing mainly the questions raised in the homework text*
5. Conclusions – *what went well; what went less well, explaining why*

3 http://wiki.ros.org/gmapping

4 http://wiki.ros.org/amcl

6. References