INTRODUÇÃO À ROBÓTICA / INTRODUCTION TO ROBOTICS

2024/2025

Mini-Project 1

Hand-out: 12 September 2024

Objective

The objective of this mini-project 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 `robot_localization`[2] to implement and test robot self-localization to estimate its position and orientation as it progresses along a path, using **Extended Kalman Filter (EKF-)based localization**. A dataset[3] in the. form of rosbag is provided

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

[2] http://wiki.ros.org/robot_localization and https://docs.ros.org/en/melodic/api/robot_localization/html/index.html

[3] https://github.com/irob-labs-ist/turtlebot3_datasets

with odometry and imu measurements along with the ground-truth poses and a map of a given environment. Implement an EKF that uses the odometry and imu data at the maximum available frequency and ground truth data down-sampled to a frequency of 1 Hz. Use the map provided in the dataset for visualization only.

2. Use ROS `gmapping` (ROS package `gmapping`[4]), with odometry and laser scanner readings **from the real robot** to map the area (e.g., in LSDC4) and store the obtained map, to be used for self-localization in 3.
3. Use ROS `amcl`[5] with the map from 2., odometry and laser scanner readings **from the data set** to test robot self-localization to estimate its position and orientation as it progresses along a path, using **Monte Carlo Localization (MCL)**. Repeat the same exercise but with readings **from the real robot.** 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:

- map provided with dataset and map estimated from the real robot using gmapping - in RVIZ;
- robot paths (estimated by the robot and estimated from the dataset) for MCL and EKF-based localization, respectively – in RVIZ;
- [for the dataset only] ground-truth path– in RVIZ;
- [for the dataset only] error (estimated position minus ground truth position) along the path with associated uncertainty (computed from the estimate covariance matrix) 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 and fluctuations in the uncertainty of the estimates may occur.

[4] http://wiki.ros.org/gmapping
[5] http://wiki.ros.org/amcl