Indoor real-time navigation for robot vehicles

Final year project for the Embedded Systems Minor

The goal of the project is to implement a real-time navigation system for a robot vehicle and to setup a complete demonstrator based on an existing robot (Turlebot 3 Burger, figure 1) operating under the ROS framework (Robot Operating System).

The first step is to implement the navigation system, to simplify, on a laptop running Ubuntu / ROS as a first instance (it will be ported to the main robot embedded computer in a second step). To this purpose, an acquisition device based on a portable autonomous laser scanner (Lidar) will be used to generate a map of the indoor navigation environment (Polytech buildings). A typical use case to build a map is the following: we want to be able to move the laser scanner around and collect data with it all while watching the map get created/updated in real time on a remote base station.

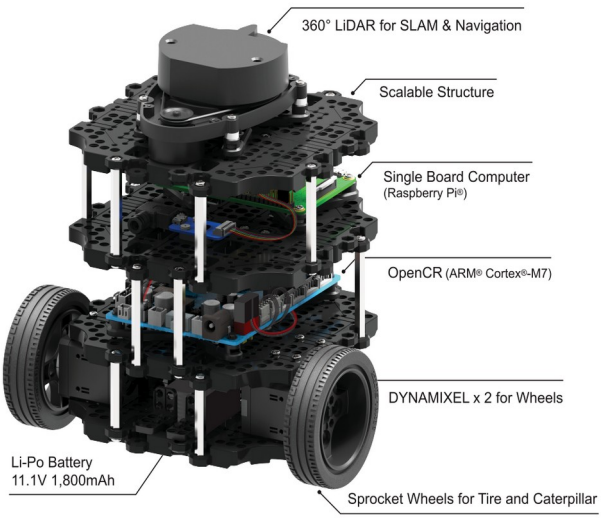

Figure 1. Turlebot 3 Burger.

The acquisition device is based on a Lidar sensor controlled by a mobile computer (laptop or robot embedded computer) that can both communicate through a serial / USB connection. The Lidar data is streamed to a remote PC base station (Ubuntu workstation) running SLAM and visualization softwares. The acquisition device and the remote base station are connected through the Polytech Wifi network and follow the ROS communication model (topics).
The SLAM algorithm can be based on Gmapping and a 3D visualization tool called RVIZ, both integrated in ROS. The PC workstation acts therefore as a base station for the system, remotely collecting Lidar samples and building the indoor navigation map in real time.

The project is made up of three parts that can be further decomposed in the following milestones:

1. SLAM on PC (workstation and laptop)
1.1 ROS setup and installation (Ubuntu 18.04 or 20.04 recommended)
1.2 ROS / RP Lidar demo
1.3 RP Lidar integration with Gmapping and SLAM demo from a laser scan bag file (workstation, laptop)
1.4 RP Lidar / Gmapping on laptop with SLAM on PC workstation using Wifi communication
1.5. Build a map of Polytech buildings (demonstration video on http://users.polytech.unice.fr/~bilavarn/)

2. Autonomous navigation with ROS
2.1 Navigation using Polytech maps (SLAM on workstation and Lidar on laptop using previous Polytech maps)
2.2 Webcam integration with video stream on RVIZ
2.3 Include real time object recognition (Yolo)

2.4 Add obstacle detection

3. Turtlebot navigation demo
3.1 Build Turlebot 3 Burger (teamwork)
3.2 Porting previous SLAM and navigation system on embedded computer (Raspberry Pi and / or Samsung Exynos)
3.2 Final demo showing Real Time autonomous navigation in Polytech buildings, with video
3.3 Final reporting, video demonstrators and GitLab

Organization, attendance and evaluation

1) The work is organized in a two-person team, each team works independently. Excessive similarity between two projects will suffer penalties in the final evaluation. Great work autonomy is an essential aspect of the project: each team must develop its own solution by himself, with minimum technical help from a professor. Two hours project meetings will take place every week essentially to check and supervise progresses in the project.
2) Final evaluation is based on the global project advancement (with respect to previous milestones), final completion, validation, quality of final solution/code/demonstration, and final report (all due to December 14). Final reporting will be limited to 5 pages maximum and structured as follows: Introduction / Specification / System overview, High-Level development / Hardware design / Software design, Tests and results, Conclusion (objective critique, possible improvements), References (articles, links, web), Appendices (code).
As a reminder, the overall coefficient for the project is 1 (4 ECTS).
3) Quality of your code will be taken into account in the final evaluation. Your code must be accessible on Polytech Github server (https://gitlab.polytech.unice.fr/) and clearly structured into directories and sub directories. Special attention will be paid to the naming of files, functions, variables and proving meaningful comments. A main README.md must be present and explain the structure of your code and how to run the full demo. It must also contain a copy of your final pdf report. See for example:
https://gitlab.polytech.unice.fr/bilavarn/SLAM_Lidar_Lite_V3_Cartographer

4) Presence to supervised project sessions is mandatory to check global project advances. Absence to supervised sessions will suffer penalties in the final evaluation.