Université catholique de Louvain – Ecole Polytechnique de Louvain

LELME2732: Robotics — 2023 edition
Problem-based learning: instructions

Louis Devillez, François Heremans and Renaud Ronsse

The problem situation is directly inspired from the "Eurobot" contest in which the students in mechatronics are taking part.

The objective of this project is to implement different modules, among which a trajectory planning method and a low-level controller. This is designed for a mobile robot with a differential-drive structure, in a simulation environment and on a real robot.

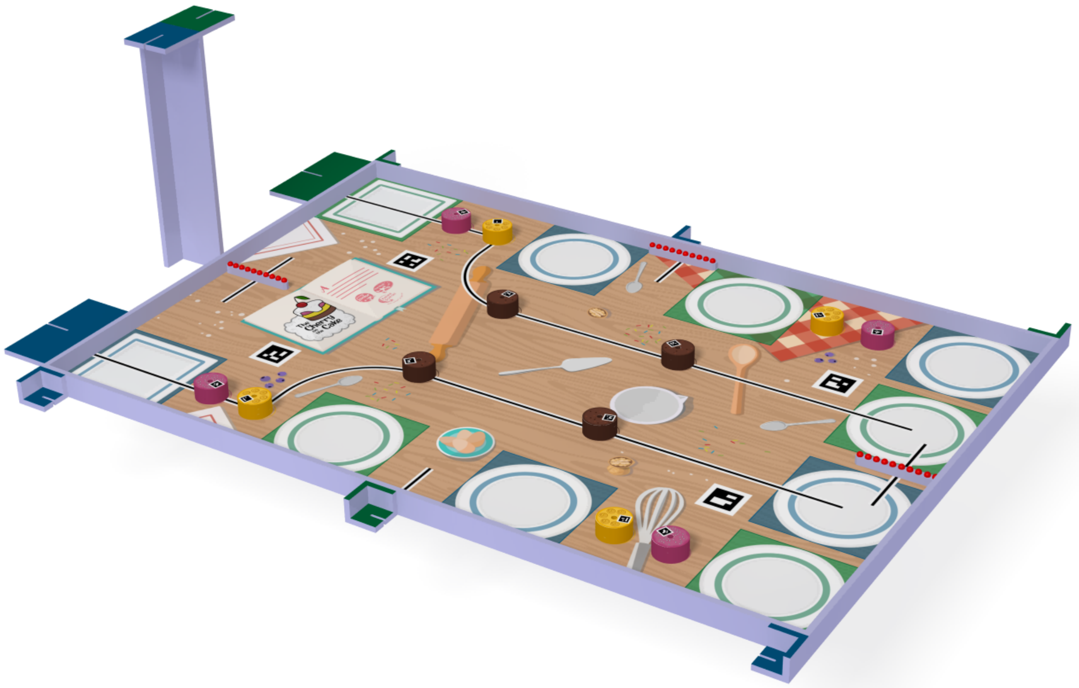

Fig. 1: Table representing the robots environment, created in the simulation framework.

To tackle this problem, you have to comply with the following constraints:

- Groups must be composed of maximum 5 students. Students in mechatronics are encouraged to make the same groups as for their project.
- Each group has to post a comprehensive report of **maximum 10 pages** (5 recto-verso) on Moodle for the 26th of March, 18:00. Annexes are allowed, but should not be mandatory to understand what is written in the report.
- The report format must be PDF and can embed some multimedia materials (e.g. some movies created with the simulator).

- Collaboration between groups is moderately allowed but **not** for the writing (you cannot copy/paste between reports) and **has to be declared** in the written report (e.g.: "For this section, we collaborated with the group of . . . ").

- The project will count for 40% of the final mark. Each member of the group will receive the group mark, except if it explicitly appeared that the workload was not fairly distributed.

- The structure of the report and its content are free, but keep in mind that this is a robotics course and not a programming-oriented one. Your report cannot be a manual explaining your functions and variables. Instead of code, favor functional blocks. . . The purpose of the report is to highlight the principles explored during the course and to explain how you adapt them to the current problem.

- When it is possible, try to explain your algorithms with an example derived from the problem you are working on. For instance, the principle of the pathfinder should be explained from an example on the project map (see Fig. 1), rather than just with general principles.

- As opposed to real experiment, the simulation offers to access and modify different parameters (e.g. robot position, level of noise ...). You are encouraged to use these data to quantify and highlight the quality of your modules (odometry compared to real robot position, opponents detection compared to their actual position. . .). Note that its is also possible to perform similar comparisons on the real robot, for instance with the real wheel speed compared to the target one.

- In your report, you need to motivate every choice. There are no bad choices as long as they are correctly motivated. For mechatronics students, choosing a solution because it is closer to your actual robot is a perfectly valid motivation (but you have to state it).

- Your report can be written in English or in French, but you can only choose one language for the whole report (except for some technical terms).

- Each figure in your report which is not made by yourself must be referenced. Otherwise, your final grade will be decreased. If you adapt a picture from another source, you can write in the legend *'with the courtesy of + reference'* in English or *'adapté librement de + référence'* in French.

- At the end of each simulation match, the teams score is computed. However, a well designed controller is more valuable than a badly designed one getting a higher score. Your grade will not be proportional to your team score.

Problem statement

We just provide here a quick overview of the problem statement. Most of the instructions are available at the following address (you are expected to read carefully this document): `https://git.immc.ucl.ac.be/LELME2732.GrX/eurobot_2023_students` (where X should be replaced by your group number). To be able to read this document, the access must first be granted, see **Git** section.

In this project, you are provided with a simulation environment (created with Robotran, `http://www.robotran.be`) representing mobile robots with either two differentially-driven standard wheels or three swedish wheels, and moving on a table of dimension 3 m x 2 m. The simulated mobile robots are models of a real robot called *Minibot*. The rendering of this table is strongly inspired from the one being used in the Eurobot contest (`http://www.eurobot.org/`), see Fig. 1.

The main objective is to grab and bring back targets to your team basis. Therefore, a good path-planning method and wheels low-level controllers are important parts of this project.

To achieve this, you have a limited time during which robots motion is allowed: $90s$. After this amount of time, a moving robot will get a high penalty for its team. The same also happens for a robot not located in its initial basis when starting the game.

Finally, obstacles and opponents avoidance is an important part of this project. Each robot touching an opponent will get a penalty. You are required to design algorithms able to avoid getting penalties.

The main developments are carried out in simulation. However, you are also requested to test the algorithms developed on a real robotic platform: the *Minibot*. Students taking part to the *Eurobot* competition will test it on their own Minibot. Other students will test it with the teaching staff, also on a Minibot platform. You do not have to reproduce all the features developed in simulation, but you need to select some points you would like to compare between simulation and the real robot (speed regulation, beacon localization, odometry, path following...). Results might be different between simulation and reality. It is important to discuss the similarities/differences between simulation and reality and to develop assumptions explaining the possible differences.

Support

The reference book of Siegwart et al. [1] is the main reference for this project. The associated lectures — taught between S1 and S4, are also of high relevance.

On top of that, five structured "tutoring" sessions will be organized during the following weeks, on Tuesday morning (10:45-12:45):

- 21/2 (S3): This session aims at providing a demo of the simulation environment. At the end, help will be provided to students facing difficulties to install it. This NOT an install party: you should already have started the installation on your own.
- 28/2 (S4): You are expected to have implemented the low-level control method, i.e. to be able to control the speed of the left and right wheels despite perturbations. You are also

expected to have selected at least your planning and obstacle avoidance methods, and started implementing them.

- 07/03 (S5): You are expected to have finished implementing the planning and obstacle avoidance methods, and to have started implementing position tracking (odometry and/or triangulation) and opponents detection. By the end of S5 you are expected to deliver an intermediate report with an overview of the algorithm.
- 14/3 (S6): You are expected to have finished implementing the position tracking (odometry and/or triangulation) and opponents detection methods.
- 21/3 (S7): You are expected to reach the end of the project and to have only minor issues to be solved. The transfer to the real robot *Minibot* might be tested during this session.

One week after this last session, i.e. Tuesday the 28^{th} of March (to be confirmed), a demo party will be organized virtually to showcase the performance of your robots. The submission of the final report, instead, is expected **by the end of week 7 (S7, 26/03)**.

Working on a joint project requiring a lot of programming can be hard to achieve within a group of more than two-three persons. Therefore, we strongly advise you to distribute the workload among the group members, taking into account that many parts can be done in parallel.

Here are some pieces of advice:

- It is possible to remove the simulated noise impacting the injected motor voltage or the reading of the wheel speeds... When some students are working on reducing these noise effects, others can already work on other modules, with no noise. For instance, the initial position of the robot is not perfectly known and requires a calibration. It is possible to remove this uncertainty, so that some students work on the following steps, while others program a correct robot calibration.
- A good path-planning method is a huge part of the project, which can be designed without having a perfect robot control at the beginning.
- Some students can work with the beacons to detect the opponents, while others control the robot on a map with no opponent.
- Global strategy (i.e. sending target positions to the path-planner) is the last programming step, but you can already start thinking about it before.

For this particular project, the supporting staff is composed of Louis Devillez, François Heremans and Renaud Ronsse.

Technical features

Git

The project is hosted on Gitlab, a web-based repository manager based on Git. Here is a short description of Git. It is a very famous code-management software, especially useful to develop programming projects in a team. However, some interesting features make it also attractive for people working alone. Thanks to Git, you can see what are all the modifications you made on the code, you can repair errors, easily merge codes coming from different contributors... Using Git for this project has also the advantage that the supporting staff has a direct access to your code, making it easier for them to help if needed. A one-hour tutorial (in French) can be found at this link:
`http://openclassrooms.com/courses/gerez-vos-codes-source-avec-git`
Similar documents exist in English and other languages.

To get access to the project, you first need to create an account on the *iMMC Gitlab server*. To do so, you need to go to `https://git.immc.ucl.ac.be/` and to connect with your *UCLouvain* account and password (the same one you use to read UCL mails, to connect to the UCL Moodle...). An account will then be automatically created. Next step is to ask to François Heremans (mail to `francois.heremans@uclouvain.be`) to add you to your group project on this Gitlab server. Please send only one mail for the whole group, with the group number and the first and last names of all the group members.

Then check the project install instructions in the *doc* folder of your group to launch the simulator.

Code and report submission

You have to submit both your controller code and your report (with possibly additional multimedia materials). Everything should be placed in a *zip* file and uploaded on the Moodle website. For your code, you will only provide the folder *grX* (*X* being replaced by the number of your group) containing your controller.

References

[1] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. MIT Press, second edition, 2011.