ROBOTICS LAB (RL) & FIELD AND SERVICE ROBOTICS (FSR)
Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione

Instructor (RL): Jonathan Cacace, Ph.D. [jonathan.cacace@unina.it]
Instructor (FSR): Fabio Ruggiero, Ph.D. [fabio.ruggiero@unina.it]

[Updated 09/06/2021]

INSTRUCTIONS FOR THE TECHNICAL PROJECT

GENERAL RULES

- The assigned topic must be solved individually.
- Students attending only Robotics Lab (RL) should follow only the parts with the RL tag. Students attending only Field and Service Robotics should follow only the parts with the FSR tag. Students attending both classes must follow both tags.
- A **detailed English report** (max. 25 pages each part, RL and FSR) must come along the produced **code** and a **video** showing the achieved results.
    - Students taking both RL and FSR must produce a single report including the two contributions (max. 50 pages).
    - The source code, the PDF of the report, and the video must be upload on a private repository on *github*.
        - When the student is ready to "deliver" the project, the instructors must be added as collaborators.
        - The README.md of the repository must contain a comprehensive description about how to download, compile, install, and run the solution. In this section, the student must also add any reference to external dependencies needed to compile/run the project solution.
    - The link to github repository must be sent to the instructors' emails. **The final oral examination day will be agreed with each instructor starting from one week later the submission of the technical report.**
    - The report should stress out the implemented choices and the motivations behind them. In particular:
        - **RL:** The student should highlight the external tools included in the project, the logic workflow of the algorithm, and the achieved results.
        - **FSR:** The student should include the mathematical formulation and all the plots of interest to the resolution of the project.
    - The report will undergo a plagiarism scan check.
- The projects are classified into three categories: *i)* wheeled robots; *ii)* aerial robots; *iii)* legged robots. For those students not attending FSR, additional topics may be considered (*i.e.*, industrial robotics) and should be asked to the RL instructor directly.
- The technical projects are assigned by the instructors based upon the students' surnames.
    - It is possible to ask the instructors, via email, for a change of the project **category** by and not later than the end of the course. The **request must be motivated** (*e.g.*, personal preference towards the wheeled robots instead of the quadrotors, or vice versa; preference to get an argument close to a thesis topic; ….). The instructors will then assign another project under the general criterion of balancing out all the arguments. Not all the requests will be accepted.
    - Students attending both RL and FSR should send one single email to both instructors.

- There are no restrictions on the programming languages and simulation environments. Therefore, a student may use Matlab, Simulink, C++, Python, physics engines, and so on. The ROS middleware must be used for those students taking both RL and FSR. There is instead a restriction on the employed libraries/commands: it is forbidden to use libraries/commands automatically solving the specific problems indicated within the description of the assigned technical project. As an example, if the project requires the development of an RRT method for planning purposes, the student must implement the whole algorithm without using a library/command implementing a method to solve it.
- The instructors are always at disposition to clarify each doubt.

TECHNICAL PROJECTS ASSIGNMENT

- Surnames A-C: Project 1
- Surnames D-GI: Project 2
- Surnames GL-N: Project 3
- Surnames O-Z: Project 4

TECHNICAL PROJECTS DESCRIPTION – PROJECT 1

Robot (RL/FSR): Differential-drive robot
Characteristics (RL/FSR): Perform a state-of-the-art research, or commercial research, to find out a differential drive robot from whose datasheet it is possible to extract the necessary kinematic and dynamic parameters. It is possible to use already existing imported models.
Scenario (RL-FSR): The goal of this project is to develop a control system for a wheeled mobile robot used in a logistic environment. The environment is composed by 3 rooms: a warehouse and two destination rooms. In this scenario, the robot must transport an object from the warehouse to a destination room. The coordinates of each room are known. During the whole task, the robot must be able to localize itself into the map without using precise information provided by the simulator (*i.e.*, odometry and additional sensors must be used to solve localization and/or simultaneous localization and mapping problem).
Mission (RL): Before starting, the robot must create a map of the environment using SLAM techniques. When the robot reaches the warehouse, it must detect and read a QR-code or an AR-Marker to know where to move the transported object. An example of the working environment is depicted in Fig. P1.

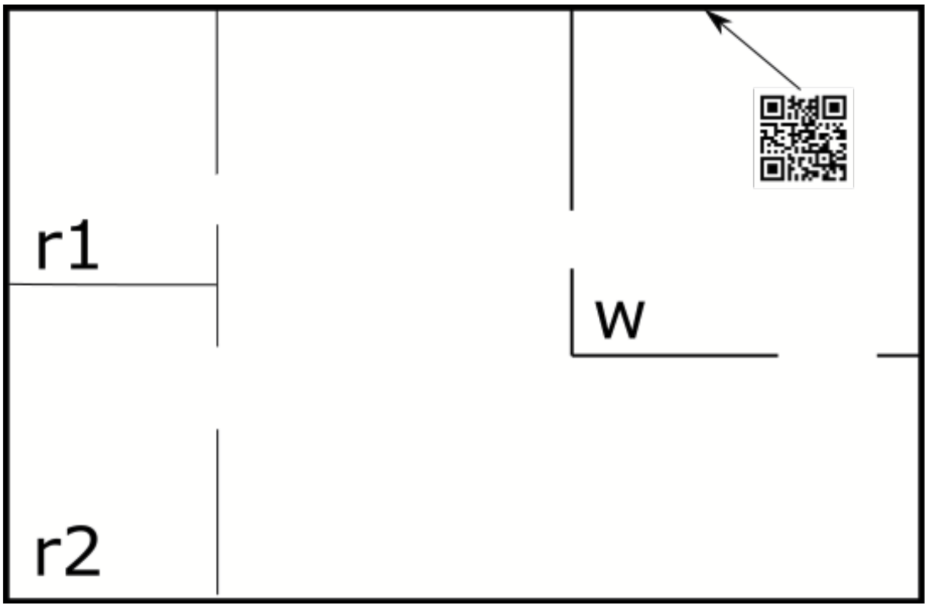

Fig. P1. The warehouse and the destination rooms have one or more doors.

Starting from a random position, move the robot into the warehouse *W*.
Search and read for the QR-code: this contains either the name or the ID of the room where the transported object should be brought.
Then, move the robot to the target room.
The resulting path should be non-trivial.
Note: During the exam, the content of the QR-code/AR-marker can be changed by the instructor to test the robustness of the proposed solution.
Planning (FSR): RRT or bi-directional RRT, along with the A* algorithm, must be employed to find out the collision-free path. A suitable time law must be implemented for each tract of the path, taking into account the maximum wheel velocity for the chosen robot.
Control (FSR): Free to be chosen among the tracking and/or the regulation controllers illustrated during the course. Please, put attention on the fact that the feedback is given by the estimated pose extracted by the odometry/localization technique. You do not have to feed back to the controller the pose retrieved from the simulator! The controller sample time is 10 ms.

TECHNICAL PROJECTS DESCRIPTION – PROJECT 2

Robot (RL/FSR): Differential-drive robot
Characteristics (RL/FSR): Perform a state-of-the-art research, or commercial research, to find out a differential drive robot from whose datasheet it is possible to extract the necessary kinematic and dynamic parameters. It is possible to use already existing imported models.
Scenario (RL/FSR): The goal of this project is to develop a control system for a wheeled mobile robot used in an exploration task. The environment is composed by 4 main locations: a starting location and three other locations to be explored. In this scenario, the robot must travel towards the locations to be explored to find a given object. The coordinates of the locations are known. During the whole task, the robot must be able to localize itself into the map without using precise information provided by the simulator (*i.e.*, odometry and additional sensors must be used to solve localization and/or simultaneous localization and mapping problem).
Mission (RL): The map of the environment is unknown. However, after the end of the task, the map of the inspected locations must be generated. Within each location, a QR-code/AR-marker is placed in an unknown position. The coordinates of the area to be explored are instead known. At the beginning, the user inserts an id among the ones present in the QR-codes within the exploring locations. The robot must navigate and explore all the locations until the id requested by the user has been found or all the locations have been explored. An example of the working environment is depicted in Fig. P2.

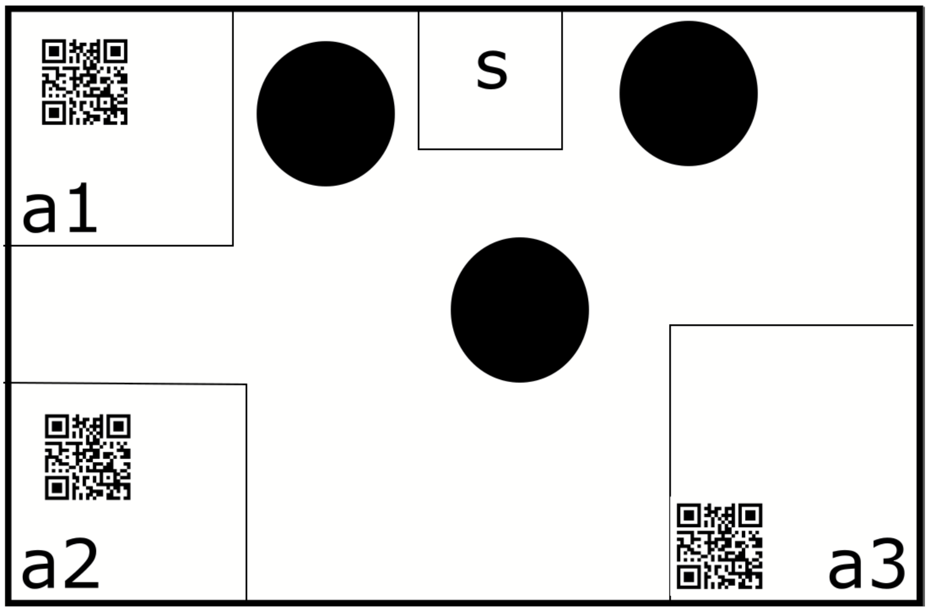

Fig. P2. Each area to be explored is a room with a door. Black circles are obstacles.

Starting from an initial position *S*, the robot moves towards one of the locations to be explored (*a1*, *a2*, or *a3*). The robot searches the QR-code inside the area. If the code contains the desired id specified by the user, the robot comes back to the starting position. Otherwise, the robot moves to another area that must be explored. If all the areas have been explored, the robot comes back to *S*. The resulting path should be non-trivial.
Note: During the exam, the content of the QR-code can be changed by the instructor to test the robustness of the proposed solution.
Planning (FSR): Planning must be carried out with an online version of the artificial potential method. The force field is seen as acceleration vector. Use random solutions to avoid possible local minima.

Control (FSR): Free to be chosen among the tracking and/or the regulation controllers illustrated during the course. Please, put attention on the fact that the feedback is given by the estimated pose given by the odometry/localization technique. You do not have to feed back to the controller the pose retrieved from the simulator! The controller sample time is 10 ms.

TECHNICAL PROJECTS DESCRIPTION – PROJECT 3

Robot (RL/FSR): Quadrotor.
Characteristics (RL/FSR): Perform a state-of-the-art research, or commercial research, to find out a quadrotor from whose datasheet it is possible to extract the necessary kinematic and dynamic parameters. It is possible to use already existing imported models. It is possible to use the RotorS library.
Scenario (RL/FSR): The goal of this project is to develop a control system for an aerial robot employed in a navigation task. This robot must navigate over a set of predefined waypoints avoiding obstacles. Obstacles can be classified into two types: solid obstacles and empty objects. In the first case, the obstacle must be avoided flying around it. In the second case, the robot must pass through it (you can consider it as a window). The obstacles must be placed at different altitudes!
During the whole task, the robot must be able to localize itself into the map using precise information provided by the simulator.
Mission (RL): The map of the environment in unknown. The robot must navigate all the waypoints passing through the empty obstacles. To recognize this kind of obstacles, a QR-code or an AR-marker must be placed on its corner allowing the robot to know where the center of the empty space is, as you can see from Fig. P3-1.

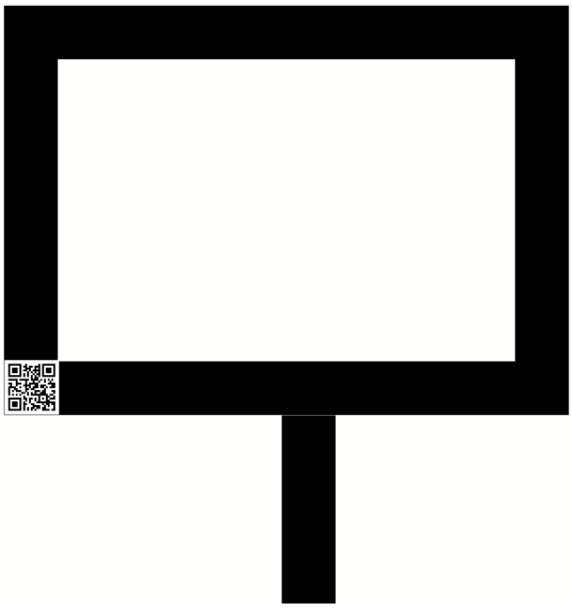

Fig. P3-1. Example of an empty obstacle with the QR-code on the corner.

Note that the location of the code, with respect to the center of the obstacle is known.
An example of the working environment is depicted in Fig. P3-2.
Starting from an initial position *S*, the robot takes off.
The robot must navigate the environment to reach the position *F*, while passing through the empty obstacles. The other obstacles must be avoided.
The resulting path should be non-trivial.
Planning (FSR): Planning must be carried out with an online version of the artificial potential method. The force field is seen as acceleration vector. Use random solutions to avoid possible local minima.
Control (FSR): Free to be chosen among the controllers illustrated during the course. The controller must also deal with uncertainty about the knowledge of the dynamic parameter of the robots. In detail, the mass in the controller is underestimated by 10%. Besides, an external force of 1N is continuously acting along the x-direction of the world inertia frame. The controller sample time is 1 ms. Notice

that the controller can be turned off during take-off and landing phases: in these cases, planning and feedforward techniques can be employed.

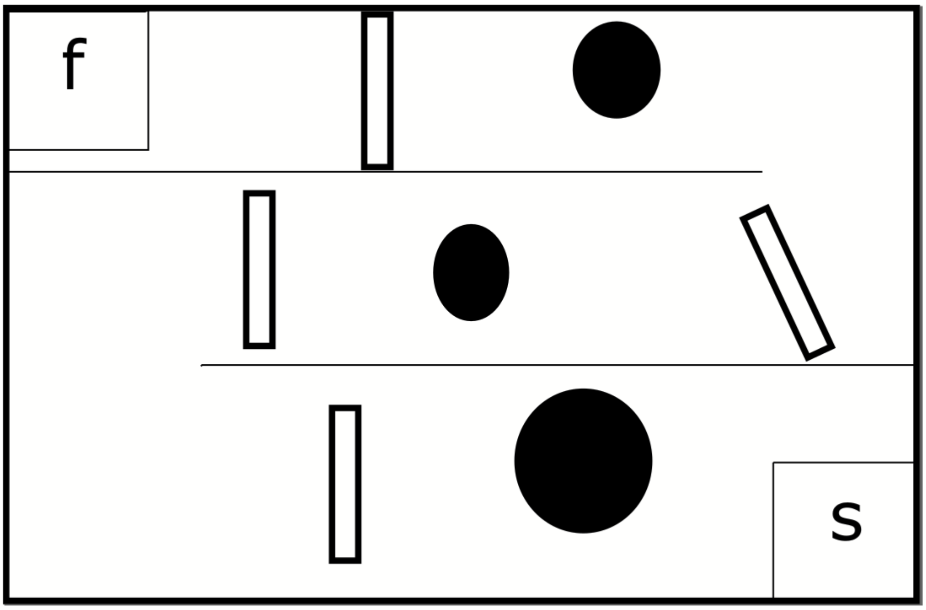

Fig. P3-2. The space is dived in corridors. Black circles are solid obstacles. White rectangles are empty obstacles.

TECHNICAL PROJECTS DESCRIPTION – PROJECT 4

Robot (RL/FSR): Quadruped.
Characteristics (RL/FSR):
The quadruped is the DogBot (https://github.com/ReactRobotics/DogBotV4).
For kinematics and dynamics, it is possible to use KDL and/or iDynTree (https://robotology.github.io/idyntree/master/index.html) libraries.
For gait planning it can be used the Towr library (https://github.com/ethz-adrl/towr). The suggested gait to be employed is the *trot*.
To solve the optimization problem within the controller, the C++ AlgLib library can be used (https://www.alglib.net/optimization/quadraticprogramming.php).
Scenario (RL/FSR): The goal of this project is to develop a control system for a legged robot robot used in an exploration scenario. The environment is composed by 4 main locations: a starting location and three other locations to be explored. In this scenario, the robot must travel towards the locations to be explored to find a given object. The coordinates of the locations are known. During the whole task, the robot must be able to localize into the map using directly information provided by the simulator.
Mission (RL): The map of the environment is unknown. However, after the end of the task, the map of the inspected locations must be generated. Within each location, a QR-code is placed in an unknown position. The coordinates of the area to be explored are instead known. At the beginning, the user inserts an id among the ones present in the QR-codes within the exploring locations. The robot must navigate and explore all the locations until the id requested by the user has been found or all the locations have been explored. An example of the working environment is depicted in Fig. P4.

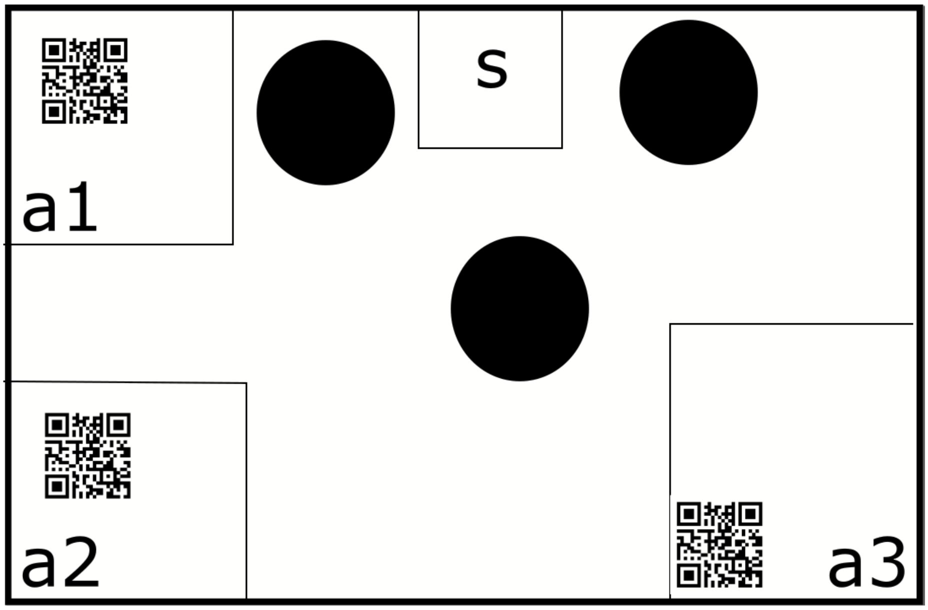

Fig. P4. Each area to be explored is a room with a door. Black circles are obstacles.

Starting from an initial position *S*, the robot moves towards one of the locations to be explored (*a1*, *a2*, or *a3*). The robot searches the QR-code inside the area. If the code contains the desired id specified by the user, the robot comes back to the starting position. Otherwise, the robot moves to another area that must be explored. If all the areas have been explored, the robot comes back to *S*. The resulting path should be non-trivial.
Note: During the exam, the content of the QR-code can be changed by the instructor to test the robustness of the proposed solution.

Planning (FSR): Planning must be carried out with an online version of the artificial potential method. The force field is seen as acceleration vector. Use random solutions to avoid possible local minima.
Control (FSR): Use the design based on the optimization control problem illustrated during the course. The controller must also deal with uncertainty about the knowledge of the dynamic parameter of the robots. In detail, the mass in the controller is underestimated by 10%. Besides, an external force of 1N is continuously acting along the *x*-direction of the world inertia frame. The controller sample time is 1 ms.

For any technical issue related to this fourth project, please contact the Assistant, Viviana Morlando [viviana.morlando@unina.it].