Universidade de Aveiro
Departamento de Eletrónica, Telecomunicações e Informática

Robótica Móvel e Inteligente / Intelligent Mobile Robotics
(Academic year of 2021-2022)

Assignment 2

Robotic challenge solver using the CiberRato simulation environment

December, 2021

1 Objectives

In this assignment each group should develop a robotic agent to command a simulated mobile robot in order to implement a set of robotic tasks, involving different navigation skills.

The list of tasks is the following:

1. Localization: The agent needs to navigate and localize itself in an unknown maze, using the movement model, the distance to obstacles and the compass. GPS is not available. Motors, distance sensors and compass are noisy. You can consult the `C4-config.xml` file to get the noise parameters. Collisions with walls will be penalized.
2. Mapping: The agent needs to explore an unknown maze in order to **completely map** its navigable cells. At the same time, the agent needs to localize target spots placed in the maze. The number of target spots may change between mazes. After completing the mapping task, the agent should return to the starting spot. Collisions with walls will be penalized.
3. Planning: The agent needs to compute a closed path with minimal cost that allows to visit all target spots, starting and ending in the starting spot.

2 The CiberRato environment

The CiberRato simulation environment will be used to assess the agent developed to overcome the different tasks. The simulated robot (see figure 1) is equipped with 2 motors (left and right) and 3 leds (visiting, returning and finish). In terms of sensors, it includes a compass, four obstacle sensors, a ground sensor, and a collision sensor. The simulated robot navigates in a delimited rectangular arena that can be seen as a bi-dimensional array of fixed size cells. Each cell is a square with side length equal to twice the diameter of the robot. The maximum size of the arena is 7-cells tall and 14-cells wide. Cells may be referenced by their position in the labyrinth where cell (0,0) is located at the bottom left corner and cell (13,6) is located at the top right corner.

A maze is defined by putting thin walls (width = 0.2 robot diameters) between adjacent cells. The limits of the outer walls are placed exactly at the limit of the arena. The target cells are detectable by the ground sensor.

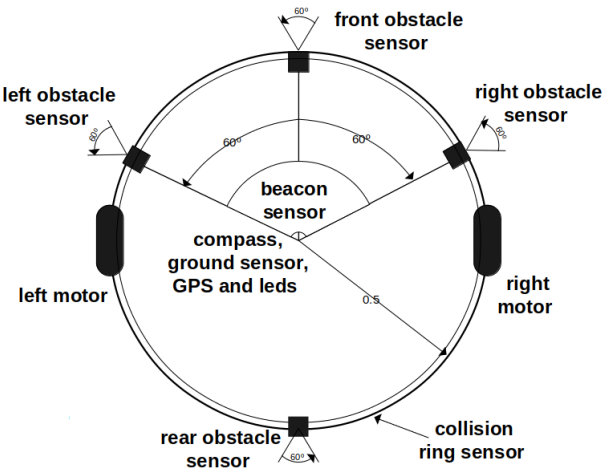

Figure 1: The simulated robot.

3 Movement model

Consider that the robot's pose is given by (x, y, θ), where x and y define the robot position and θ specifies the robot orientation. When the command sent to the simulator at step t is `DriveMotors(`in_t^l, in_t^r`)`, then the following equations determine the new robot pose.

An IIR filter is applied to each of the powers provided by the agent (in_t^l and in_t^r) that models the inertial characteristics of the motors and generates the effective powers that will be applied to the motors, corresponding to

$$out_t = \frac{in_i + out_{t-1}}{2} * \mathcal{N}(1, \sigma^2) \tag{1}$$

where out_t is the power applied at time t, out_{t-1} the power applied at time $t-1$, and $\mathcal{N}(1, \sigma^2)$ Gaussian noise with mean 1 and standard deviation σ.

Then, the movement is splitted in a translation of the robot position, considering its current orientation, followed by a the change of the orientation of the robot. For the translation one has

$$x_t = x_{t-1} + lin * cos(\theta_{t-1}) \tag{2}$$

$$y_t = y_{t-1} + lin * sin(\theta_{t-1}) \tag{3}$$

$$lin = \frac{out_t^l + out_t^r}{2} \tag{4}$$

and for the rotation

$$\theta_t = \theta_{t-1} + rot \tag{5}$$

$$rot = \frac{out_t^r - out_t^l}{D} \tag{6}$$

where D is the robot diameter (1 in the CiberRato environment). This provides the new robot pose (x_t, y_t, θ_t) at the next step, in case no collisions occur. If the new pose involves a collision, the simulator only applies the rotational component.

4 Scoring scripts

Each execution of your agent should create:

a map file showing robot's internal view of the map

a path file that specifies how to go from the starting position, visit all targets and return to the starting position.

These files should have the same base filename and extensions `.map`, for the map file, and `.path`, for the path file (see `run.sh` for an example on how to create these files).

To check if the map file is correct, you can use the the `mapping_score.awk` script, executing:

awk -f mapping_score.awk planning.out ***myrob.map***

in the `simulator` directory.

The map file is a text file that contains 27 lines, each with 55 characters. Where the center of the file always represents the starting position, marked as `0`. Below you can find an example of a map file (line numbers are not part of the file contents):

```
1
2
3
4
5
6
7
8
9
10
11                      - - - - - - - - - - - - - -
12                     |XXXXXXX|XXXXXXXXX|XXXXXXXXX|
13                      - - X - X - - X X X - - - X
14                     |XX0XXXXXX|XXXXX|X|XXXXXXXXX|
15                      X - - - X X - - X - X - - X
16                     |XXX|XXX|XXX|XXX|1XX|XXX|XXX|
17                      - X X - X - X X - X - X - X
18                       |X|XXXXX|XXX|X| |XXXXX|XXX|
19                      - X - - - X X X - - X X - X
20                     |XXX|XXXX2XX|X|X|XXXXX|XXXXX|
21                      X - X - - X X X X - X X - X
22                     |X| |X|XXXXXXX|XXX|XXXXXXXXX|
23                      X - X X - - X - - X - - - X
24                     |XXXXXXXXXXXXXXXXXXXXXXXXXXX|
25                      - - - - - - - - - - - - - -
26
27
```

To check if the path file is correct, you can use the the `planning_score.awk` script, executing:

awk -f planning_score.awk planning.out ***myrob.path***

in the `simulator` directory.

The path file is a text file that contains, in each line, the x and y coordinates of the center of the cells that are visited by the path. Coordinates are relative to the starting position and are measured in robot diameters. The starting and ending coordinates should always be `0 0`. Below you can find an example of a path file (line numbers are not part of the file contents):

```
1 0 0
2 2 0
```

```
4 0
6 0
6 2
8 2
10 2
12 2
14 2
14 0
14 -2 #1
16 -2
16 -4
18 -4
18 -6
18 -8
16 -8
16 -10
14 -10
12 -10
10 -10
10 -8
8 -8
8 -6
6 -6 #2
4 -6
2 -6
2 -8
2 -10
0 -10
-2 -10
-2 -8
-2 -6
0 -6
0 -4
0 -2
-2 -2
-2 0
0 0
```

5 Evaluation

The evaluation of this assignment will be composed of 3 components: an execution test of the agent, a report and a presentation.

- The agents will be tested and graded by teachers, in batch mode, in their own computers. Thus, the format specified for their execution is mandatory. The agent must generate a file representing the map of the maze, generate a file indicating the best path computed and finish its run in the starting spot.
- The report must follow the Springer LNCS paper template and must contain the following sections:
  - the *Introduction*, describing the general approach followed to tackle the different tasks and summarizing the results achieved;

  - a section entitled *Localization approach*, describing the approach taken for localization;
  - a section entitled *Results*, presenting how the evaluation of the agent was done and the results achieved;
  - the *Conclusion*, drawing the main conclusions and pointing out future directions;
  - the *References*, listing the bibliographic documents used.
- Each group will make a presentation of its work, consisting of a 10-minutes oral presentation (including discussion time), based on PDF slides, made available to the evaluation board beforehand.

6 Deliverables and deadline

- Source code of the developed agent. The source code should be contained in a folder called **agent** that includes:
  - all the source code of the agent;
  - an executable version of the script **run.sh** that starts the agent for C4 and supports the options presented in the following example:
    ./run.sh -c 4 -p 0 -r myagent -h 127.0.0.1 -f fname
    where `fname` is the name of both the mapping and path files.
  - an executable version of the script **build.sh** that allows to build (compile, ...) your agent.
- Report (in PDF format, according to Springer LNCS paper template).
- Presentation (in PDF format).

Source code, report and presentation must be submitted in the Moodle's course page. The following dates apply:

- Source code and report: January 23$^{\text{th}}$, 2022, by 23:59.
- Presentation: January 27$^{\text{th}}$, 2022, by 13:00.

7 Bibliography

- "Principles of Robot Motion: Theory, Algorithms, and Implementations", Howie Choset et al., MIT Press, Boston, 2005.
- "Introduction to Autonomous Mobile Robots", Second Edition, Roland Siegwart et al., MIT Press, 2011.
- "Artificial Intelligence: A Modern Approach", 3rd edition, Russel and Norvig, Pearson, 2009.