Liège University
Academic year 2018-2019

Introduction to intelligent robotics

Milestone A1

AUBRY Julie
BOURDOUXHE Alexandre

Liège

31 mars 2019

1 Mapping

To perform the mapping, we use a grid of 300 over 300 cells. Each cell of the grid represents a square of ten over ten centimeters in the real word. If the home is bigger then 30 meters over 30 meters, we simply double the size of the map in the direction needed and copy the previous map in the center of the new one. A cell can contain three values

- 0 when the cell is unexplored
- 1 when the cell is explored and free
- 2 when the cell is blocked.

After every refresh of the map, we check if the map is completed. It is done by checking if an unexplored cell is nearby an explored and free cell. If the map is incomplete a reachable free cell nearby the unexplored cell is use as the next position to achieve.

2 Trajectory

A structure *trajectory_cell* is defined. It contains three variables x, y (the coordinates) and previous (the previous *trajectory_cell*) and a function traj returning an array containing all the previous *trajectory_cell*. The function *trajectory* returns an array of *trajectory_cell* defining the trajectory followed by the robot. The function is based on the following principle :

Algorithm 1 trajectory

```
Require: plan, youbot_position, target_position
  youbot_trajectory_cell.x = youbot_position.x
  youbot_trajectory_cell.y = youbot_position.y
  tab = array of trajectory_cell
  i = 1
  j = 2
  find = false
  plan_explored = matrix of size plan initialised at false
  tab(i) = youbot_trajectory_cell
  while ! find do
    current_cell = tab(i)
    for new_cell = left, right, top, down cells do
      new_cell.previous = current_cell
      if new_cell.x is close to target then
        find = true
      end if
      if plan(new_cell) == free and plan_explored(new_cell) == false and new_cell is reachable then
        tab(j) = new_cell
        j++
        plan_explored(new_cell) = true
      end if
    end for
    if i == j then
      break
    end if
    i++
  end while
  if find then
    return new_cell.traj().
  end if
  return empty array
```

3 Motion

We decide to separate motion and rotation as well as to move only in straight lines. We perform a mapping after every rotation and motion, and at the middle of a long motion (more than 3 meters of distance).

The rotation consists in choosing between 4 angles, depending on the direction of the movement. Indeed, as the robot currently performs horizontal or vertical movements only, it only needs to rotate in those directions. X and Y coordinates of the next position are analysed, and compared with the robot's coordinates to choose the correct angle. The rotation speed is equal to the angle difference, and decreases as the difference decreases.

The motion consists in computing the distance between the next position and the current position of the robot. As the movement is performed along the y or x axis, only one component is compared. The speed of the robot is set to the distance with the next position, and thus decreases as the distance decreases. Once again, the direction of the movement is chosen based on the X and Y coordinates of the robot and those of the next position.

Until the target is reached, the robot will loop between the states of motion and rotation, as it performs the determined trajectory.

The code can be found in the Matlab file joined to the report.

4 Video

The video of the robot exploring the map is available on YouTube. The link to the video is the following : Robot Video