CDA 4621 / CDA 6626

Grad Total: 110 points + Extra Credit

Undergrad Total: 100 points + Extra Credit

Spring 2024

Lab 3

Motion Planning

Total: 100 points

Due Date: 3-7-2024 by 11:59pm

The assignment is organized according to the following sections: (A) Objective, (B) Requirements, (C) Task Description, (D) Task Evaluation, and (E) Lab Submission.

A. Objective

This lab will teach you how to do motion planning for a robot to reach a goal while avoiding obstacles. This lab will teach you about: (1) Camera, and (2) Object Detection.

A.1 Camera

Camera will be used to recognize colored landmarks.

A.2 Object Detection

Objects are detected based on specific color. For this lab, the robot needs to recognize a single yellow colored object.

B. Requirements

Programming: Python

B.1 Physical Robot

See Appendix B.1.1 Robobulls-2023 or Appendix B.1.2 Yahboom 2024.

B.2 Robot Simulator

See Appendix B.2.1 Webots "RosBot".

Github Server: https://github.com/biorobaw/FAIRIS-Lite (see "README.md")

C. Task Description

The lab consists of the following tasks:

- Task 1: Motion To Goal
- Task 2: Bug Zero Algorithm
- Statistic Task (Required only Grads, Extra Credit Undergrads)
- Extra Credit Task: Tangent Bug Algorithm

C.2 Task 1: Motion to Goal

The robot needs to reach a yellow cylinder (goal) and stop when 0.5 meters from it, see Figure 1. There are no obstacles between the robot and the goal. Robot should be tested to start from different locations and orientations. Since the goal may not be visible when facing a different direction, the robot will need turn until the goal is in view and then move towards it. You may apply a PID controller, where the input function can be the distance and orientation of the yellow cylinder (blob), and the output can be the individual linear wheel velocities. You may use the camera to recognize the camera and its orientation and the distance sensors to compute distance. Task should run for 30 sec.

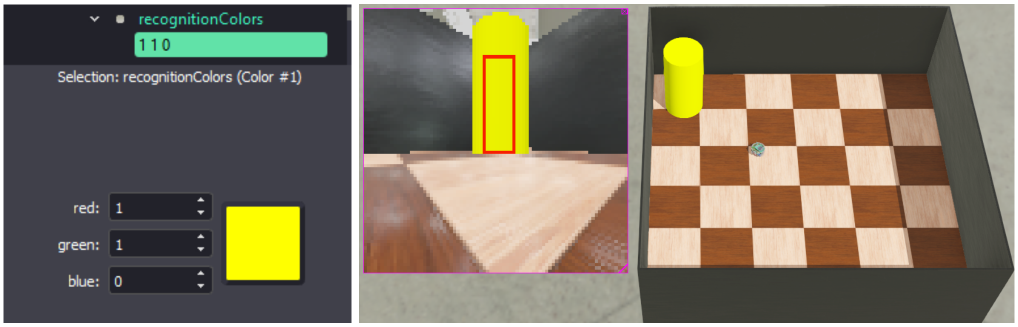

Figure 1. Camera color object recognition in Webots for motion to goal.

C.2 Task 2: Bug Zero Algorithm

The robot should implement the Bug Zero algorithm, using the worlds shown in Figure 2. Design a single controller for both worlds. The goal is represented by the yellow-colored cylinder from Fig 1. The robot should start from "S" and move towards the goal "G" while avoiding obstacles. Robot should navigate around obstacles no further than 0.5 meters away from them. You may use the camera to recognize the camera and its orientation and the distance sensors to compute distance. Task should be completed in less than 3 minutes. Your algorithm should perform correctly for either left or right turns.

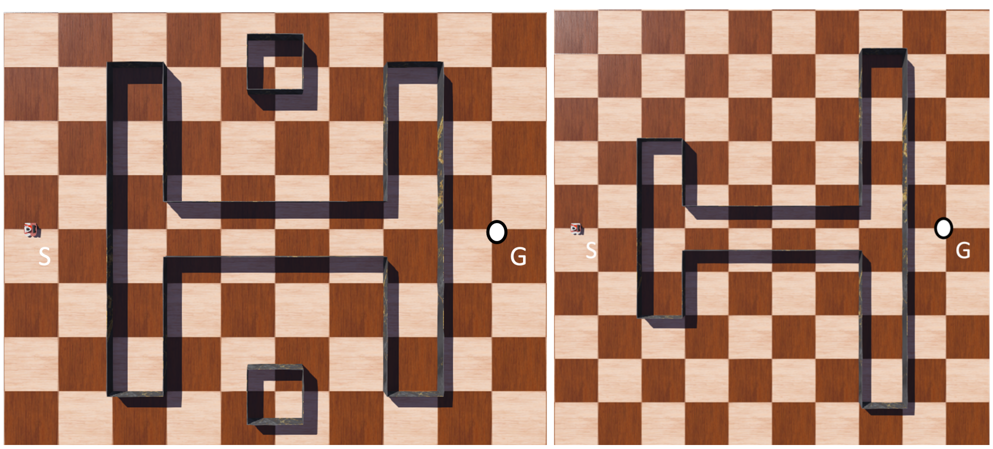

Figure 2. Different object configurations, with robot starting and 'S' and goal at 'G'.

C.3 Statistic Task: Bug Zero Algorithm

Perform statistics on the following:

- Run both tasks with 3 different speed combinations while trying to minimize overall travel time.
- Analyze and graph min, max and variance of velocities, orientations, and distances to goal and obstacles.
- Discuss your results including any navigation errors.

C.4 Extra Credit Task: Tangent Bug Algorithm

The robot should follow the Tangent Bug algorithm, using the worlds shown in Figure 2. Design a single controller for both worlds. The goal is represented by the yellow-colored cylinder. The robot should start from "S" and move towards the goal "G" while avoiding obstacles. Wall following should be no further away than 0.5 meters from the wall. You may use the camera to recognize the camera and its orientation and the distance sensors to compute distance. Task should be completed in less than 3 minutes.

D. Task Evaluation

Task evaluation: (1) program code, (2) report, including a link to a video showing each different task, and (3) task presentation videos of robot navigation. Note that tasks are needed in future labs.

D.1 Task Presentation (90 points)

The following section shows the rubric for the different tasks:

- Task 1 (45 points)
  - Robot reaches the cylinder from any starting position and orientation (20 points)
  - Robot finds the cylinder when the cylinder is not seen in the camera (20 points)
  - Robot stops 0.5 meters or less from the cylinder (5 points)
  - Robot hits the cylinder (-5 points)
- Task 2 (45 points)
  - Robot completes correct bug zero algorithm stopping within 0.5 meters of the cylinder (20 points)
  - Robot moves correctly around walls (10 points)
  - Robot switches correctly from motion to goal to wall following (5 points)
  - Robot switches correctly from wall following to motion to goal (5 points)
  - Robot performs consistent left or right turns (5 points)
  - Robot hits the cylinder (-5 points)
  - Robot travels a hardcoded path defined in advanced (-10 points)
- Statistics Task (10 points) - Required for grad students. Extra credit for undergrad students.
  - Statistics are correctly computed, analyzed, and graphed (10 points)
- Extra Credit Task: (45 points task + 5 points report)
  - Robot completes correct tangent bug algorithm stopping within 0.5 meters of the cylinder (20 points)
  - Correct endpoints and interval of continuity are computed (5 points)
  - Correct heuristic distance to the goal is computed (5 points)
  - Robot switches correctly from motion to goal to wall following (5 points)
  - Robot switches correctly from wall following to motion to goal (5 points)
  - Robot hits the cylinder (-5 points)
  - Robot travels a hardcoded path defined in advanced (-10 points)

D.2 Task Report (10 Points)

The report should include the following (points will be taken off if anything is missing):

- Mathematical computations for all kinematics. Show how you calculated the speeds of the left and right servos given the input parameters for each task. Also, show how you decide whether the movement is possible or not.
- Conclusions where you analyze any issues you encountered when running the tasks and how these could be improved. Conclusions need to show an insight of what the group has learnt (if anything) during the project. Phrases such as “everything worked as expected” or “I enjoyed the project” will not count as conclusions.
- Video uploaded to Canvas showing the robot executing the different tasks. You should include in the video a description, written or voice, of each task. You can have a single or multiple videos. Note that videos will be critical in task evaluations.
- Deductions:
  - Handwritten text or images (-5 points)

    - Missing video link requires in person presentation (-20 points)
    - The submission format is not correct (see assignment details) (-10 points)
    - Failure to answer TA's email within 48 hours of the initial email (-50 points)
    - Additional deductions may apply depending on submission

Videos

Video needs to be clear and audible and must show the robot performing the correct path and showing the program outputs printed to the console, as outlined in the assignment. TA may ask you additional questions to gauge your understanding via Canvas Message or MS Teams. Failure to reply may result in point deduction.

- Task 1 Video
    - Record a single video of the robot performing the motion to goal task.
- Task 2 Video
    - Record a single video of the robot performing Bug Zero algorithm.
- Extra Credit Video
    - Record a single video of the robot performing the Tangent Bug algorithm.

E. Lab Submission

Each student needs to submit the programs and report through Canvas under the correct assignment. Submissions should have multiple file uploads containing the following:

- The "zip" file should be named "yourname_studentidnumber_labnumber.zip" and should contain the Python file containing your controller. Controllers should be named as "Lab3_Task*X*.py", where *X* is the task number. The zip file should contain any supporting python files needed to run your code. For example, if you created functions to perform actions and these are kept in a file, this file needs to be included. At the top of each python file include a comment with the relative path from FAIRIS_Lite that this file needs to be placed in. Example #WebotsSim/libraries/my_functions.py.
- Videos should be contained in a separate zip folder with the name "yourname_studentidnumber_labnumber_videos.zip" and added as a separate file upload to Canvas.
- The report should be in PDF and should be uploaded to Canvas as a separate file.
- All zip files must be in .zip format. We will not accept RAR, 7z, tar, or other.