COMP 581: Introduction to Robotics

Fall 2020
Instructor: Prof. Ron Alterovitz
Teaching Assistant: Victor Murta <scarlet@cs.unc.edu>

Lab 2: Wall Following

Deadline

February 26, 2020 at the beginning of class. For the late policy, see the course syllabus. **Before the due time, each team member should individually submit your team's code in a zip file on Sakai**; see specific instructions in the Code section below**.**

The Objective

The primary purpose of this lab is to implement forward kinematics and wall following for a mobile robot.

Robot Design Requirements

Your robot must be built only using pieces from your Lego Mindstorms kit. Your should design your robot such that the dark gray center button may be pressed without moving the robot and such that pushing the button does not disrupt any of the sensors. Your robot should remain in one piece at all times, its diameter with respect to the ground should not exceed 40 cm, and its diameter shall not change substantially during the task. All measurements will be made with respect to the *measuring point*: a distinct point of your choosing that is on the **frontmost surface** of your robot. More specifically, your robot will be placed on the ground behind a starting line drawn on the ground, and your chosen measuring point must be a point on your robot directly above the starting line, and no part of your robot is permitted to be in front of the starting line. Your robot should be designed to accomplish the task autonomously without any human intervention (except for pushing the dark gray center button as required).

The Task

The task in this lab is to be completed in a single run, without picking up your robot.

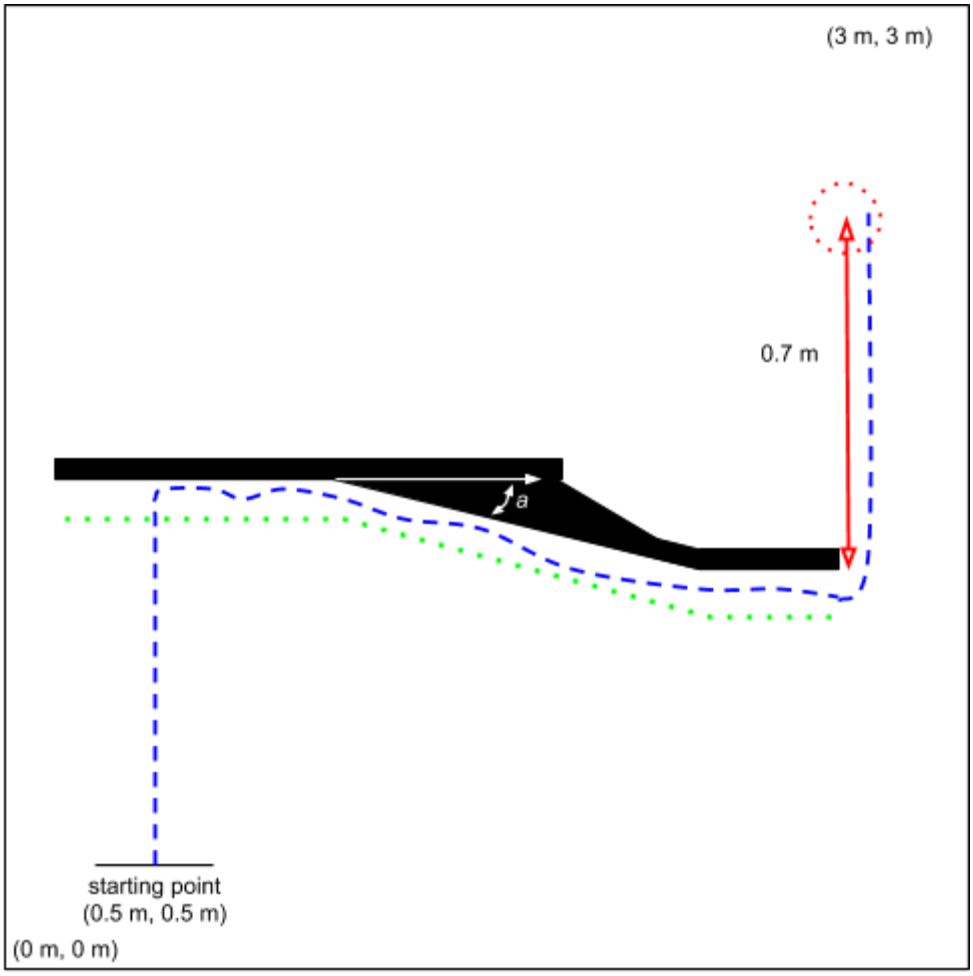
Your robot will be placed behind a starting line such that its measuring point will be on a specific *starting point* on the starting line (see figure below). You will then push the dark gray center button. Your robot should then move straight forward (in the positive *y* direction the ground). Straight ahead, at a distance of somewhere between 60 and 120 cm, will be a wall that is perpendicular to the robot's forward motion. Once your robot is less than 30 centimeters from the wall, your robot should turn right and follow the wall. Once starting to follow the wall, your

robot measuring point should never be more than 30 cm from the wall at all times while following the wall.

The wall may be curved and may bend up to 45 degrees. The wall will be at least 17 cm high, of unknown width, and at most 5 cm thick near its ends (although the wall may have greater thickness at other points).

The robot should follow the wall until the wall ends. When the wall ends, your robot should turn left and move in the positive *y* direction (the same direction the robot was heading at the beginning) for 0.7 m. Your robot must then stop as accurately as possible, 0.7 m past the wall.

All of the robot motion must be completed within a 3 meter square workspace without leaving the workspace. The starting point is at coordinate (0.5m, 0.5m). The task (from the time the button is pushed until the robot stops moving) must be completed in under 90 seconds.



In the figure above, we illustrate an example of a possible environment that complies with the rules. We note that the environment in your actual run may be different. The robot starts at the starting point. The wall is shown in black. The angle of any bend (e.g., angle *a* in this

environment) is unknown in advance, but it is at most 45 degrees with respect to horizontal. *Note that there may be more than one bend*. The dotted green line represents 30 cm distance to the wall. The blue dashed line shows a possible robot trajectory that would receive full credit if completed in the allotted time and assuming the robots stops inside the dotted red circle.

Points:

Points for tasks:
Attempt: 40 points. You will receive these points simply by having EV3 MicroPython successfully installed, showing up to the room on the lab due time, submitting your code on Sakai on time, having a robot that conforms to the design requirements, and having your robot attempt the course.
Detect the wall: 10 points. You will receive these points if your robot stops or attempts to turn right when it approaches the wall.
Turn at the wall: 5 points. You will receive these points if your robot successfully turns right and starts following the wall.
Following the wall: 20 points. You will receive 20 points if your robot is never more than the allowable maximum distance from the wall. This requirement must be satisfied 60 cm after the robot is supposed to turn right until the end of the wall. Partial credit will be awarded.
Turn at end of wall: You will receive 5 points for stopping or successfully turning left at the end of the wall.
Accuracy of reaching goal: 20 points. Points will be based on the accuracy of reaching the robot's target location when it stops at the end of the task. Full points will be awarded if the robot stops inside a circle of radius 15cm around where it ideally should end. Partial credit will be given based on distance to this circle.

Penalties:
There will be a 10 point penalty if, after crossing the start line, your robot ever exits the 3m by 3m workspace.
There will be a 10 point penalty if more than 90 seconds elapse from when the dark gray center button is pushed at the start location until the robot stops.
If the robot is still moving after 2 minutes, then the instructor can stop the run and no additional points can be earned.

Code

To complete this lab you will use a subset of the EV3 MicroPython API:
https://le-www-live-s.legocdn.com/sc/media/files/ev3-micropython/ev3micropythonv100-71d3f28c59a1e766e92a59ff8500818e.pdf
Important: You may only use the following modules from EV3 MicroPython API (covered in chapters 3-6 of the API document above):
ev3brick
ev3devices
parameters

tools
You may *not* import, use, or, copy the “robotics” module, or any of its included classes or functions, in your program (covered in chapter 7 of the API document above).
You should feel free to use any module from MicroPython that is available by default on your EV3 as described on the MicroPython API:
http://docs.micropython.org/en/latest/index.html

A robot relying on a module not allowed as stated above will be disqualified. Also, any robot using Wi-Fi or Bluetooth communications in any way will be disqualified.

You must submit on Sakai a zip file of your Python source code files. The zip file should include all the Python source code files you wrote (which should have a .py file extension). In the comments at the top of the Python source file containing your main .py file, please include the names and PID’s of all team members.

You should feel free to discuss concepts in the course with other students in natural language (e.g. English). You should not share any programming code with others and must write all the robot’s programming code yourself. If you access any sources other than the textbook or documents on Sakai, you must cite them in comments at the top of your code and send an e-mail to the instructor with the citation. You must fully understand your code and be able to reproduce the algorithmic approach without references if asked. The Honor Code is in effect for these policies.

Good luck!