COMP 581: Introduction to Robotics

Fall 2020
Instructor: Prof. Ron Alterovitz
Teaching Assistant: Sandeep Kumar <sandeep@cs.unc.edu>

Assignment 6: Bug Race

Deadline

November 11, 2020 at the beginning of class. For the late policy, see the course syllabus. **Before the due time, please individually submit your controller code in a single file on Sakai**; see specific instructions in the Code section below**.**

The Objective

The primary purpose of this lab is to program the Pioneer mobile robot in the Webots simulator to move as quickly as possible from a start location to a known goal location in an unknown environment while avoiding static obstacles.

The Setup

This lab will be completed using the Webots simulator. You must implement a controller that enables the Pioneer robot to achieve the required task. Given a Webots World file that includes the Pioneer robot and an environment, your controller should make the robot move to the goal location while avoiding obstacles in the environment. Your robot should be designed to accomplish the task autonomously without any human intervention.

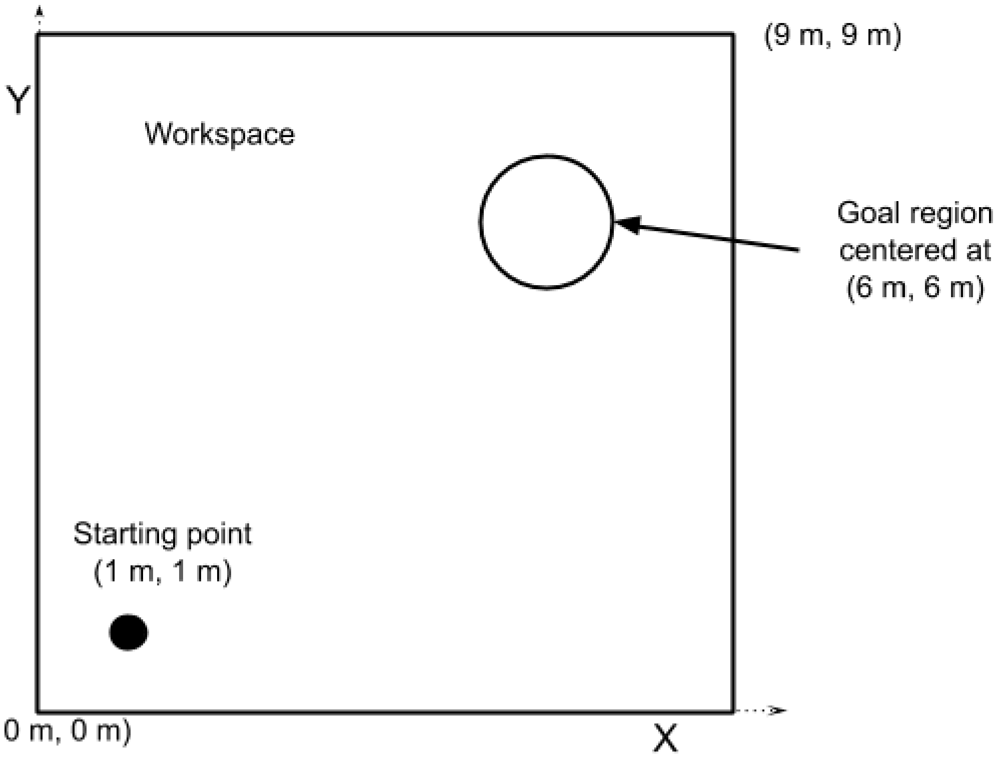
To help you develop your controller, we provide you with the Pioneer robot in a few sample environments (see Sakai for the .wbt world files). You should design your controller such that it is able to complete the same task even in a different environment. To test your controller, we will use different test environments (although following the same basic structure as the sample environments) to run your controller, and you will be evaluated on the basis of the performance of your robot in those environments. All measurements will be made with respect to the **frontmost center point** of the Pioneer Robot.

The Task

The Pioneer Robot will be placed at the *starting point* (1 m, 1 m) facing the positive y-direction. Note that the coordinates in Webots are different and you should consider the one defined in the figure below for the task. The *goal point* is located at (6 m, 6 m) and the *goal region* is defined as the region that is within 50 cm of the goal point (This is denoted by the blue patch in the world files). (Note that the goal point will be free of an obstacle but that some of this goal region may be occupied by an obstacle.) All robot motion must be completed within the 9 m x 9 m workspace.

The objective for your robot is to autonomously move from the starting point to the goal region as fast as possible while staying in the workspace. Upon running the simulation using the "*Run in real time*" button in Webots, your robot should begin the task. Upon reaching the goal region, the robot should stop.

In the workspace there will be between 2 and 6 obstacles. Your robot will not know the shapes or locations of the obstacles before starting the task. Each obstacle will be made of ultrasonic reflective walls that are at least at the height of the sonar sensors of the Pioneer robot. Each obstacle will be a closed shape composed of straight and curved segments and may contain corners. It is not necessarily a convex shape, and no angle on the surface of the obstacle will be greater than 120 degrees. No gap on the ground between two obstacles will be less than 100 cm wide.



This lab is to be completed in a single run and all robot motion must be completed within the specified workspace. The task must be completed in under 3 minutes.

Points

Points Awarded:

- Attempt: 20 points. You will receive these points simply by having a valid controller associated with the Pioneer robot and submitting your code on Sakai on time, and having your robot attempt the course. We should be able to use the controller which you provided to make the robot move forward.
- Moving around obstacles: 40 points. If your robot stops in the goal region, you will receive full credit for moving around obstacles. If your robot does not stop in the goal region, some points will be awarded if your robot follows the wall of at least one obstacle it encounters, based on the number of edges followed.
- Goal accuracy: 20 points. Points will be awarded based on proximity to the goal point when your robot stops. If your robot does not stop autonomously before the time limit, no goal accuracy points will be awarded.
- Time: 20 points. Points will be awarded based on time elapsed from when the "*Run in real time*" button is pressed at the beginning until the robot stops. Faster completion will be awarded more points. If your robot does not stop in the goal region, no points for time will be awarded.

Penalties:

- There will be a 10 point penalty if, after starting, your robot ever exits the 9m by 9m workspace (or collides with the outer brick walls).
- Exceed time limit: If your robot does not stop by 3 minutes, then your run will be terminated and no further points after 3 minutes will be earned.

Code

Important: You may only use the following modules from Webots:
Robot
Motor
DistanceSensor
PositionSensor
GPS

You may *not* import, use, or, copy other modules from Webots, or any of its included classes or functions, in your program. You can still use other basic modules to handle computations like Numpy for Python or stdio.h, stdlib.h, and math.h for C. If you would like to use any other library, please contact TA Sandeep Kumar via email and we will consider adding it to the list.

Please follow the overall structure which Webots provides for the controller when you create a controller using the Wizards menu. Look up the Pioneer documentation on Webots website for the names of the sensors, the robot would also have a GPS sensor with the name 'gps' which you can use. Further details on how to use the GPS sensor can also be found in Webots documentation.

A robot relying on a module not allowed as stated above will be disqualified.
You must submit on Sakai a single Python or C file for your controller (which should have either a .py or .c file extension). In the comments at the top of the file, please include your name and PID.

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 email 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!