Lab 1: Reactive Behaviors and State Machines

CSCI 3302: Introduction to Robotics

Report due 1/28/25 @ 11:59pm

The goals of this lab are to understand

- Setting up a world in Webots
- Getting data from sensors on the e-puck robot
- Position and velocity control
- Using state machines to implement a basic controller

You need:

- A functional Python 3.7 or 3.8 development environment
- Webots robot simulation software (https://cyberbotics.com/)

Overview

A very simple way to program a robot is to directly tie its sensor input to wheel motion. Examples include obstacle avoidance, for example "turn right if you see an obstacle in front", light following or avoidance, or line following. One type of these robots is a "Braitenberg Vehicle". It would be quite limiting, however, if robots would exhibit always the same behavior when presented with stimuli. To achieve more complex behaviors, robots need to switch between different operational modes based on the context they're in. In the first part of this lab, you will play with different standard behaviors. You will then create a simple finite state machine and switch between different behaviors to accomplish a complex task.

Instructions

Each group must develop their own software implementation and turn in a lab report. **You are encouraged to engage with your lab partners for collaborative problem-solving**, but are expected to understand and be able to explain everything in your write-up. If your group does not finish the implementation by the end of the class, you may continue this lab on your own time as a complete group.

Updated 9/9/2020 to add Hints/Clarifications/Extended Due Date

Part 1: Introduction to Webots

1-A) Download Webots version R2023b https://cyberbotics.com/#download

1-B) Follow the first four tutorials on the Webots website to familiarize yourself with the software interface and components: https://cyberbotics.com/doc/guide/tutorials

1. First Simulation: https://cyberbotics.com/doc/guide/tutorial-1-your-first-simulation-in-webots
    a. Create a project directory and follow instructions on creating an arena. You will need to submit a screenshot of this.
2. Modifying the Environment: https://cyberbotics.com/doc/guide/tutorial-2-modification-of-the-environment (Optional)
3. Modifying the Appearance of Objects: https://cyberbotics.com/doc/guide/tutorial-3-appearance (Optional)
4. Creating a Robot Controller: https://cyberbotics.com/doc/guide/tutorial-4-more-about-controllers

Part 2: World Creation and Controller Programming

Open lab1.wbt world in webots.

Implement a state machine-based controller that recreates the following behavior:

The robot follows the wall on its left side, adjusting to maintain a consistent distance from the wall while moving straight. If the robot encounters an obstacle in front, the robot makes a right turn to avoid the obstacle before continuing to follow the wall. Once the robot detects the light source from left, front, and right light sensors (ls0, ls1, ls2, ls5, ls6, ls7) it will make a 180 degree turn and follow the wall on its right side. When the robot detects the light source again, it stops.

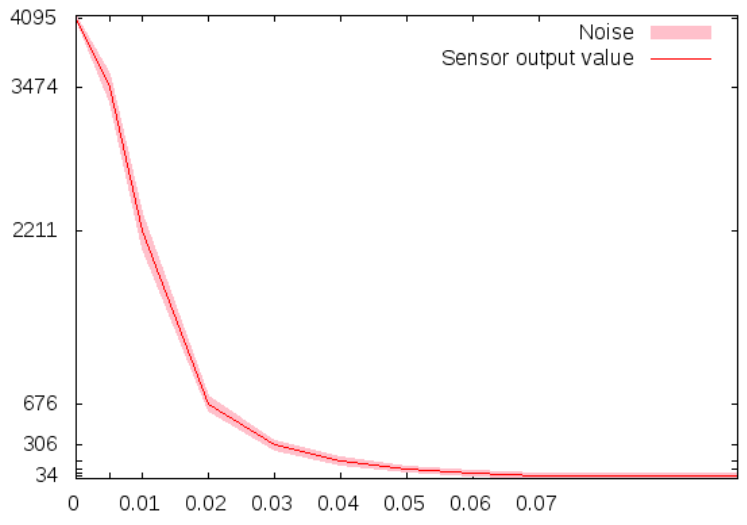
HINT: Planning your answer for Part 3 - #3 before writing code will make this easier.

HINT: You can manually time out how long the robot should actuate its wheels to turn around, or you can use the distance sensors behind it to tell when you've finished turning.

HINT: The distance sensor doesn't return a distance value, it returns an intensity measurement: you will need to convert the returned value to a distance, which will only be approximate given the low accuracy of this sensor.

You can use the lookup table below to map distance sensor values to distances:

Light Intensity



Meters

HINT: The distance sensor on the e-Puck is very noisy! You can right-click the robot and click "Show Robot Window" to view sensor readings. You can then place the robot near obstacles and see what the sensor values are to use as thresholds in your code.

Part 3: Lab Report

Create a report that includes/answers the following:

1. The names of everyone in your group

2. Screenshots of your world from Tutorial 1 and 4.

3. A drawing of your state machine. Make sure all the states and transitions are labeled and that it is faithful to your implementation.

4. Were you able to get your controller to complete the task? If not, which parts failed? Why?

Updated 9/9/2020 to add Hints/Clarifications/Extended Due Date

5. A statement indicating whether you have worked with Python before, and if so, describe your experience.

6. How much time did you spend programming **Part 2** of this lab?

Please submit a zip file containing your Lab Report in PDF form, your Webots world file (.wbt), and Controller file (.py) on Canvas. We only need one submission per group.