Wednesday, 10 May 2023

Line-following Competition 1
Cognitive Robotics 2022–2023

Lab session

1 Getting started

This is the first of three sessions in which we will create a robot that can quickly and accurately follow a white line on the ground. Your mBot is equipped with a downward-facing line detection sensor that can sense whether the ground below it is light or dark. Your performance will be measured on a test track in Snellius 404, but you will practice on an improvised track that you can make yourself. You will again write a lab report on this topic, so make sure you keep detailed notes during these three weeks!

2 Prerequisites

You will need the Arduino IDE and the libraries to interface with your mBot. You have installed and tested this before, so this should not be a problem. You can use the document on Brightspace as a reference for relevant functions and methods.

You can find a suitable location inside the Snellius building to start developing and testing. Get a dark gray garbage bag and some white tape from the TAs, which you can use to make your own test track. Make sure you use a double layer of tape, otherwise it might not be white enough for the line-following sensors.

3 This week's goals

The *end goal* is to create the fastest and most accurate line-following robot. But, like most things in this course, the *process* and its detailed description is what really counts. The goals below are described to guide you along your process, and are necessary but not sufficient parts of your lab report.

3.1 Getting your mBot ready

Compile a simple program to your mBot using the `MeDCMotor` and `MeLineFollower` objects. Examine the values that they accept and return when placed on a dark and light floor.

3.2 Following a straight line

What would be necessary to follow a straight line? Think about the state space of the robot. What measurements can be used to determine the robot's position? Can you use PID control? Why or why not? Why and when may it not be a good idea to send full power to both motors?

3.3 Following a curve

Investigate curves with different angles and radii. Do you handle sharp curves differently from others? How sharp can your curves be? What do you think will happen when your robot encounters a 90-degree angle?

3.4 Getting lost

Part of the “cognitive” part of robotics is to reason about how to solve problems in a smart (and maybe human-like) way. What do you do when both sensors lose the track? Do you move backwards? Can you think of a way to use *memory* to keep track of or estimate the robot’s location?