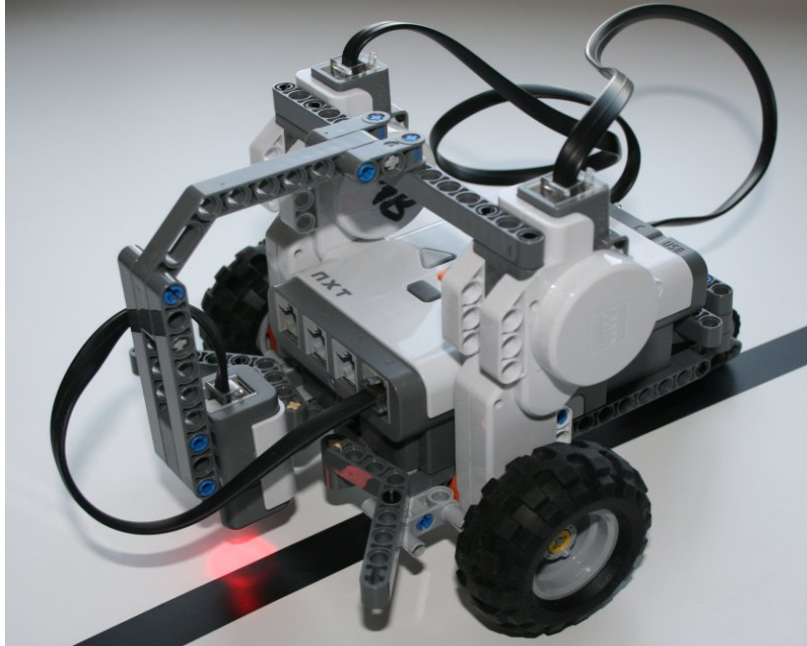


Fuzzy Line Following NXT Robot

Stefan Bracher

One of the tutorials within the Lego Mindstorms NXT Software shows how to program a robot that follows a line using Boolean Logic. In the following, an improved version, based on Fuzzy-Logic, is presented. Although a little bit more complex, the program is still implemented using NXT-G Code.



Original "Boolean Logic"-Version by Lego

The original version by Lego uses the reflected light to distinguish between black (the line) and white (the floor). Whenever the light sensor is over the black line, the left motor is activated, what makes the robot turn right. When the sensor is on the white surface, the right motor is activated and the robot turns left. In this way, the robot follows the line. - Or to be correct, it follows the boundary between black and white.

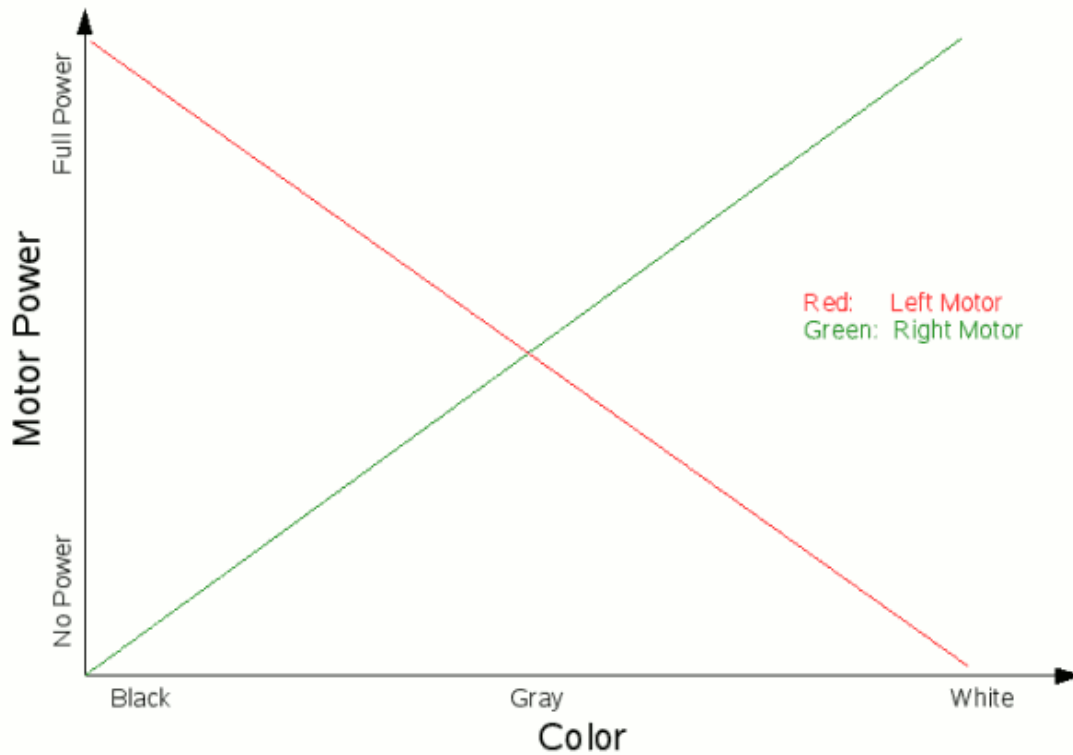
The approach is very simple and effective. However, as the robot only distinguish between black and white, the resulting movement looks as if it would be drunk.

Fuzzy Logic variant

The idea behind the new version is, not only to distinguish between black and white, but to consider various degrees of grey. If the sensor is exactly on the boundary, it will see a colour that is between black and white. The more the sensor gets on the line, the blacker the perceived colour is. On the other hand, the more the sensor gets on the white surface, the whiter the perceived colour is. This is called Fuzzy Logic as we do not consider absolute states (Boolean Logic, here "Black and White") but different degrees of those states (Whiter, Blacker).

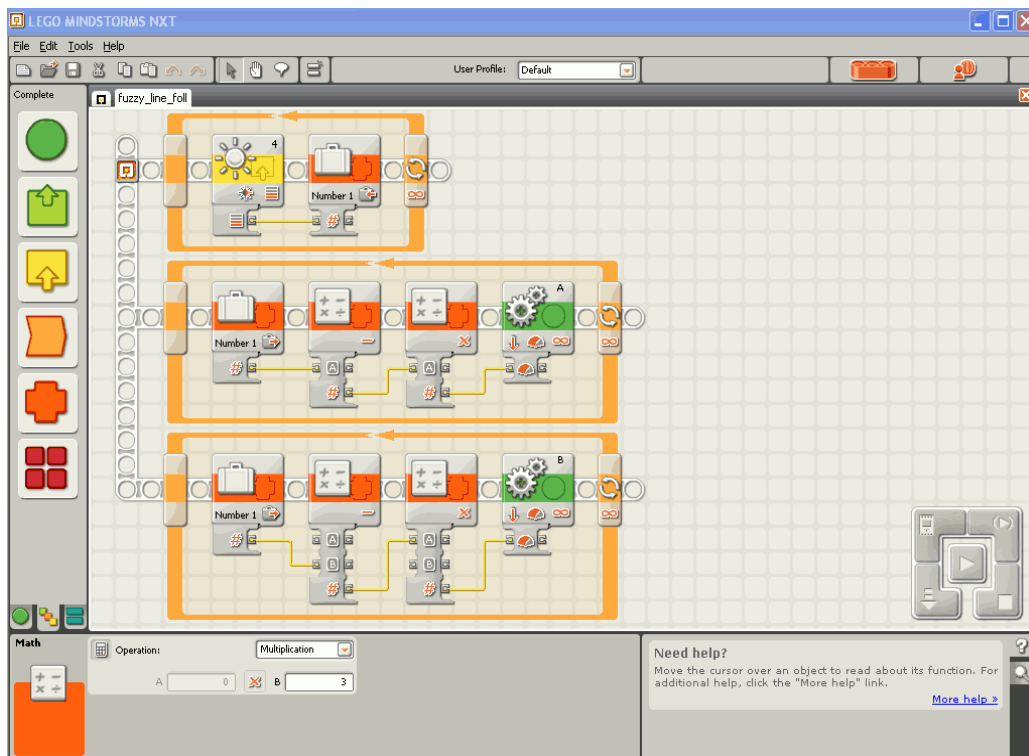
Now we are not limited anymore to turn only left and right, but can turn more right (blackier colour) or more left (whiter colour). In this way, if exactly on the colour boundary, the robot runs straight ahead.

To achieve this, the left and right motors are controlled according to the diagram below:



Implementation in NXT-G Code

Implemented in NXT-G Code that looks as follows:



The program consists of three loops that run at the same time. The first loop reads the current reflected light value and stores it in the variable called "Number 1". The second and third loop control the left and right motors.

The reflected light sensor gives a value between 0 and 100. A typical value for a black surface is around 20 and for a white surface something around 60. These values depend on the ambient light conditions and the distance of the

sensor from the surface. The motor power itself as well is a value between 0 and 100.

To produce the motor commands as described in the schema, the following calculations have to be done:

- $A_Power = (Sensor_Value - Black_Value) * Speed_Factor$
- $B_Power = (White_Value - Sensor_Value) * Speed_Factor$

For $Black_Value=20$, $White_Value=60$ and a $Speed_Factor$ of 2.5 we get the following results:

Sensor_Value	A_Power	B_Power
20	0	100
30	25	75
40	50	50
50	75	25
60	100	0

Fine-tuning

If you program your NXT with the above program, and the robots does run as expected, check the following:

1. Make sure you have a robot with the two drive motors connected to the ports A and B.
2. Your light sensor should be around 1cm above the surface.
3. Take a reading of what your robot sees as black and replace the value in the subtraction block of the second loop with it.
4. Take a reading of what your robot sees as white and replace the value in the subtraction block of the third loop with it.
5. Start with a multiplication factor of 1 (2nd and 3rd loop) and increase it slowly.