

# NXT Hackathon

Learning about robots with lego

Patrick Nagel    Tim Metzler

Autonomous Systems  
Bonn-Rhein-Sieg University  
of Applied Science

14. März 2016

# Table of contents

- ▶ Line Follower - Problem
- ▶ Line Follower - P Controller
- ▶ Line Follower - Sourcecode
- ▶ Line Follower - Demonstration
- ▶ D

# Line Follower - Problem

**Problem:** Follow a black line with the NXT using only light sensors.

**Solution:** Use two light sensors at each side of the line. Each sensor should see the bright surface next to the line. Light sensor values are read every 10ms. If the right sensor is dark we go to the right. If the left sensor is dark we got to the left. If both sensors are dark we have passed the finish line.

# Line Follower - P Controller

**P Controller:** Measure the error from the sensors. Adjust the motor speeds by adding / subtracting a correction value.

Correction value =  $K_p * \text{error}$ ,  $\text{error} \in \{-1, 0, 1\}$

# LineFollower - Sourcecode

```
1 public class FollowTheLine {
2
3     public static final int TOP_SPEED = 200;
4
5     // The constant for the proportional
6     // controller
7     private int kp;
8     // The correction value (how much we slow down
9     // / speed up the motors)
10    private int turn;
11
12    // The error value (-1, 0, 1)
13    private int error;
14
15    // Calibration values
16    private int darkRight;
17    private int darkLeft;
18
19    private int brightRight;
20    private int brightLeft;
21
22    private int triggerRight;
23    private int triggerLeft;
24
25    private LightSensor lsRight;
26    private LightSensor lsLeft;
```

## LineFollower - Sourcecode continued

```
24
25     public FollowTheLine() {
26         // The value of 120 is selected by
           trial and error
27         kp = 120;
28
29         // Initialize the light sensors
30         lsRight = new LightSensor(SensorPort.
           S1);
31         lsLeft = new LightSensor(SensorPort.S2
           );
32
33         // Calibrate the two light sensors
34         calibrateSensors();
35
36         // Start to follow the course
37         followTheCourse();
38     }
```

## LineFollower - Sourcecode continued

```
39
40      /**
41       * Start following the course
42       */
43      public void followTheCourse() {
44
45          Motor.A.setSpeed(TOP_SPEED);
46          Motor.B.setSpeed(TOP_SPEED);
47
48          // Smoother acceleration
49          Motor.A.setAcceleration(2000);
50          Motor.B.setAcceleration(2000);
51
52          // Start moving
53          Motor.A.forward();
54          Motor.B.forward();
55
56          int lightValueRight;
57          int lightValueLeft;
58
59          while (!Button.ESCAPE.isDown()) {
60              error = 0;
61
62              // Small delay
63              Delay.msDelay(100);
```

## LineFollower - Sourcecode continued

```
64
65         // Get the current light values of
           both sensors
66         lightValueRight = lsRight.
           getLightValue();
67         lightValueLeft = lsLeft.getLightValue
           ();
68
69         // If both sensors see dark, we
           reached the finish line
70         if (lightValueLeft < triggerLeft &&
           lightValueRight < triggerRight) {
71             Motor.A.stop();
72             Motor.B.stop();
73             return;
74         }
75
76         if (lightValueRight < triggerRight) {
77             error = -1;
78         }
79
80         if (lightValueLeft < triggerLeft) {
81             error = 1;
82         }
```



## LineFollower - Sourcecode continued

```
83
84         // P-Controller
85         turn = kp * error;
86
87         // Adjust the speed
88         Motor.A.setSpeed(TOP_SPEED + turn);
89         Motor.B.setSpeed(TOP_SPEED - turn);
90
91         // Handle the turn direction
92         if (error < 0) {
93             Motor.A.backward();
94             Motor.B.forward();
95         }
96
97         if (error > 0) {
98             Motor.A.forward();
99             Motor.B.backward();
100         }
101
102         if (error == 0) {
103             Motor.A.forward();
104             Motor.B.forward();
105         }
106     }
```

## *Live demonstration of the Line Follower*

*Thank you!*  
*Any questions?*