



Institut für Informatik
der Universität Zürich
Andreasstr. 15
CH-8050 Zürich

Artificial Intelligence Lab

Prof. Dr. Rolf Pfeifer, pfeifer@ifi.unizh.ch
Marc Ziegler, mziegler@ifi.unizh.ch
Nathan Labhart, labhart@ifi.unizh.ch
Maik Hadorn, hadorn@ifi.unizh.ch
Dorit Assaf, assaf@ifi.unizh.ch

Introduction to Artificial Intelligence

Winter Term 2006/2007

Introduction to LEGO Mindstorms NXT

Practical Exercise 2

Note:

This is the second practical exercise, which should help you become acquainted with the synthetic methodology of robot experiments. Before you start this exercise, please read the “Tutorial for Programming the LEGO Mindstorms NXT” by Mike Rinderknecht, where in the first part you get an overview of the hardware and in the second part a short introduction to the LEGO programming environment. You can find this document on the class website. Don’t be afraid if this exercise represents your first exposure to a new programming environment and a new kind of LEGO bricks. You have the chance to write a program from scratch as well as to modify the agent’s morphology. In addition, there are plenty of assistants that are eager to help you. THE ANSWERS HAVE NOT BE HANDED IN.

Fundamental Idea

The practical exercise has three main parts. First, you will get familiar with the new programming environment. You will write a first, very simple program. The second part has its focus on the morphology of the agent. Within a few restrictions, you are free to change any part and connection. Finally, you bring together morphology and controller. The goal of changing the morphology of the agent and its controller is to travel as far as possible. By a little competition between the groups at the end of the lesson, we will see, which of your phenotypes runs farthest.

Insiders' tip: Each group gets only **one** set of new batteries at the beginning of the lesson. Use them economically.

Part one

In part one you can only change the controller.

1. Read and follow carefully the tutorial. **Note:** After starting the NXT software (see tutorial 3.1 Starting a Program) choose **OneMotor.rbt** as the name of your new program. Make sure that before you download the program to your brick (see tutorial 3.5 Downloading a Program to the NXT Brick) you have chosen the following setting for the motor-block to adapt the program of the tutorial to your agent: port: B, direction: backward, power: 75. Save your program and upload it onto the brick. Let the agent run, observe its behaviour, and record your observations.

2. Remove now the connection between the two motors as shown in Fig. 1. Let the agent run (**OneMotor.rbt**), observe its behaviour and record your observations. Change the controller in such way that the agent shows a similar behaviour as in step 1. What is different? Save your new program as **TwoMotors.rbt**.

3. Try now to improve (remember, the goal is to move as far as possible in a given time) the controller still **without** changing the morphology of the agent. What do you realize?



Figure 1

Part two

In part two you only change the morphology.

- 4. Use **only** your program **TwoMotors.rbt** in part two. Make sure that the settings are as follows: ports: B and C; direction of both motors: backwards; power of both motors: 75. Reconnect the two motors by using the connection removed in step 2 (Fig. 1). The two motors should be phase-delayed by 180° (as at the beginning). Let the agent run, observe its behaviour, and record you observations.

- 5. Change the feet as shown in Fig. 2. Note: The feet should be on the outside of the leg. Let the agent run, observe its behaviour, and record your observations. Compare the behaviours shown in step 4 and 5.

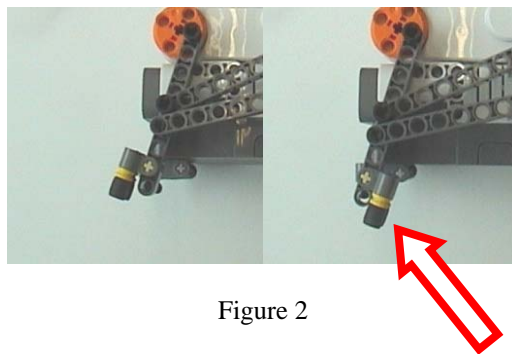


Figure 2

Part three

In part three you are free to change the morphology as well as the controller.

- 6. The only restriction in morphology is to keep the brick-motor block setup unchanged. Keep in mind that by the end of the lesson, we will measure how far your agent gets in 10 seconds.

<http://mindstorms.lego.com/>, LEGO Mindstorms website