Project 1: Line following using the e-puck's camera

Lucie Dross, Chloé Udressy

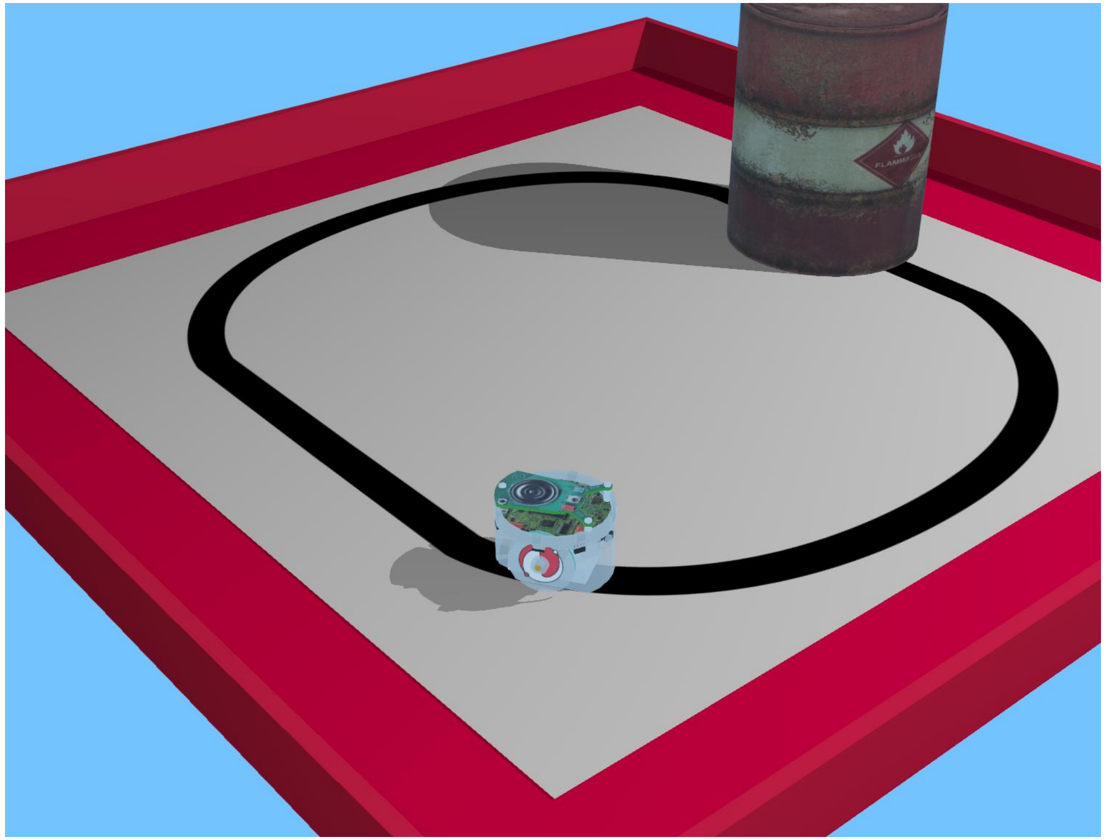
26 May 2020

Professor: Alcherio Martinoli
Supervisors: Chiara Ercolani, Anwar Quraishi

EPFL

Overview

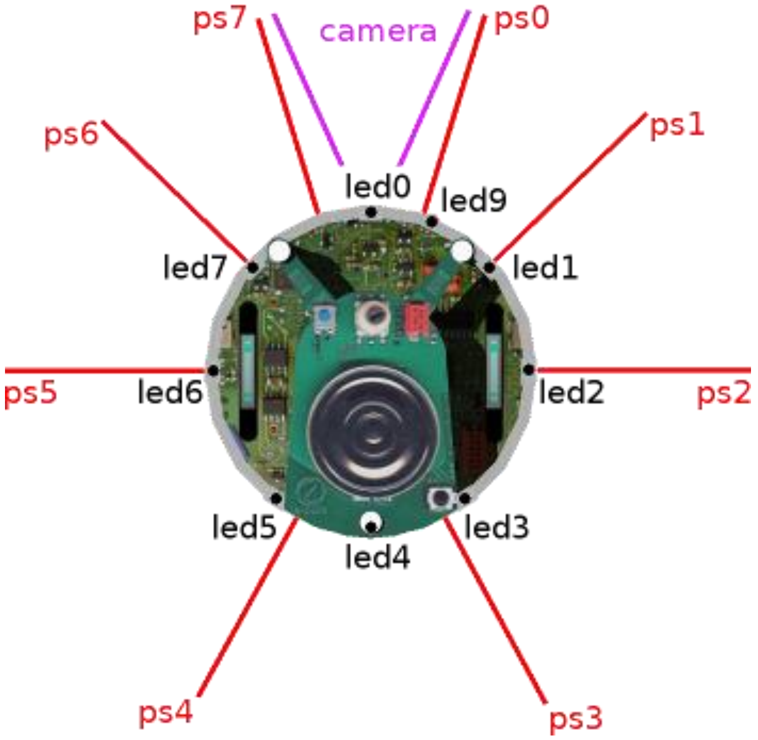
- Introduction
  - E-puck
  - Objectives
- Methods
  - Algorithm implementation
  - Trajectory using odometry and supervisor data
- Experiments and discussion
- Conclusion



E-puck

Sensors and actuators used

- camera RGB (52x39 resolution)
- 8 IR proximity sensors
- motion sensors
- 2 wheel motors

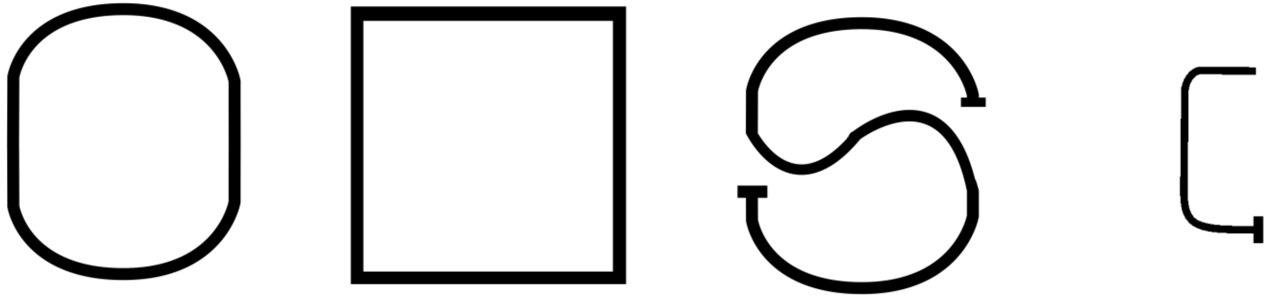


In Webots

- non-linearities and noise of sensors and actuators approximately reproduced

Objectives

- following 4 line shapes



- reacting to line ends
- avoiding and getting around obstacles
- comparing ground-truth and odometry positions

Structure of our algorithm

Initial stage: “passive” line searching

0. 360° rotation

First stage: active line searching and obstacle avoidance

1.1 searching and going to line

1.2 obstacle avoidance

Second stage: line following and obstacle circumvention

2.1 simple line following

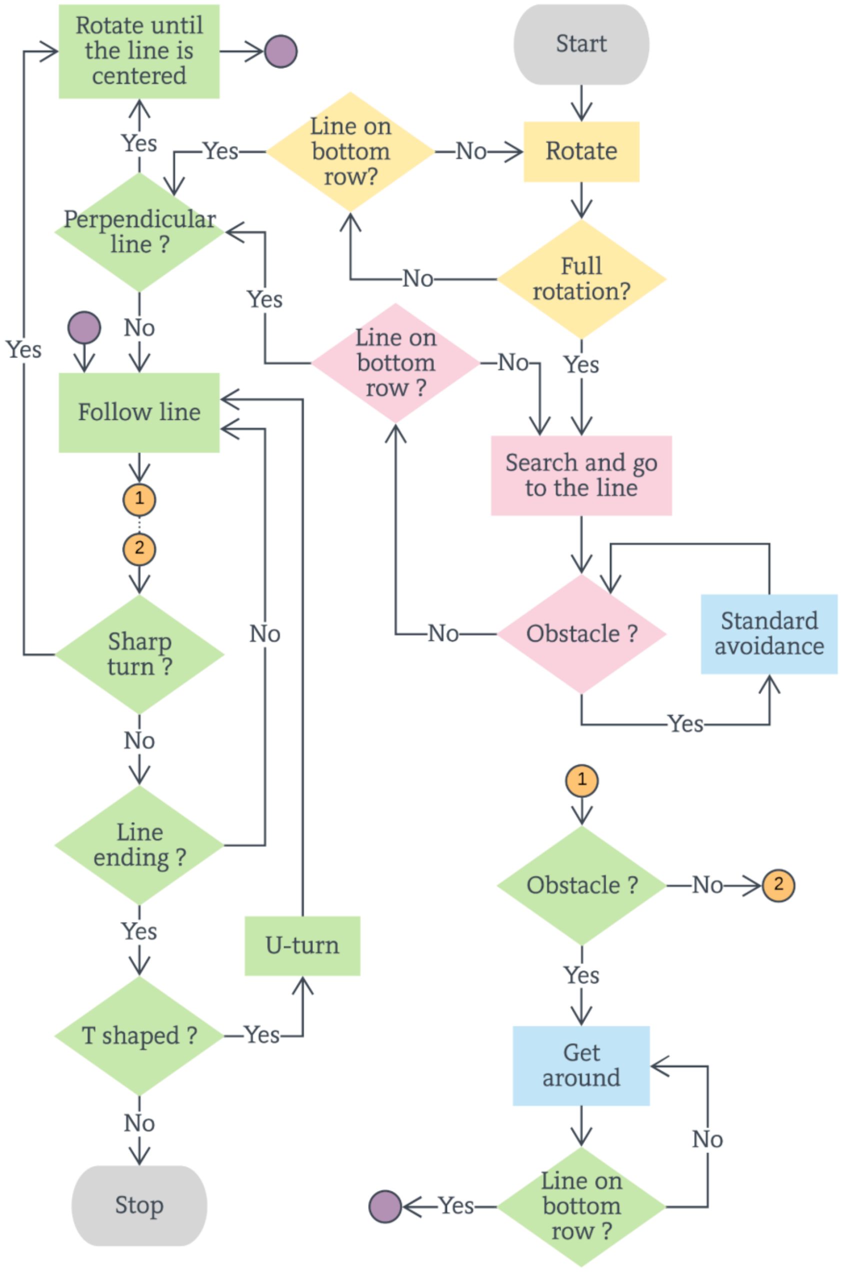
2.2 obstacle circumvention

2.3 sharp turn

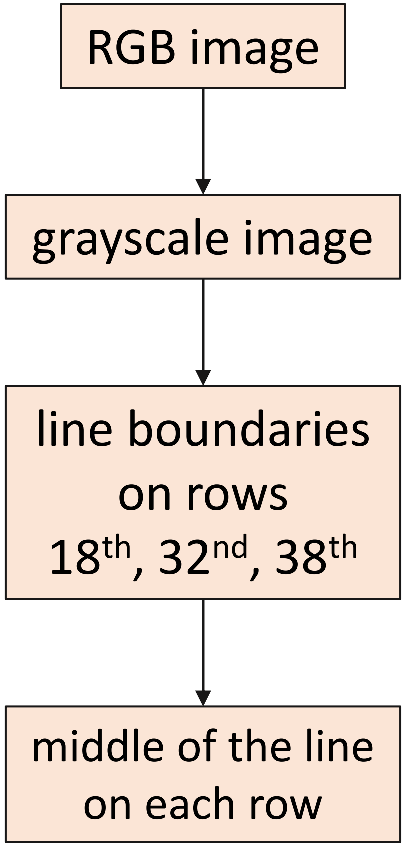
2.4 line ending, T shaped

2.5 line ending, | shaped

2.6 line perpendicular to the trajectory of the robot



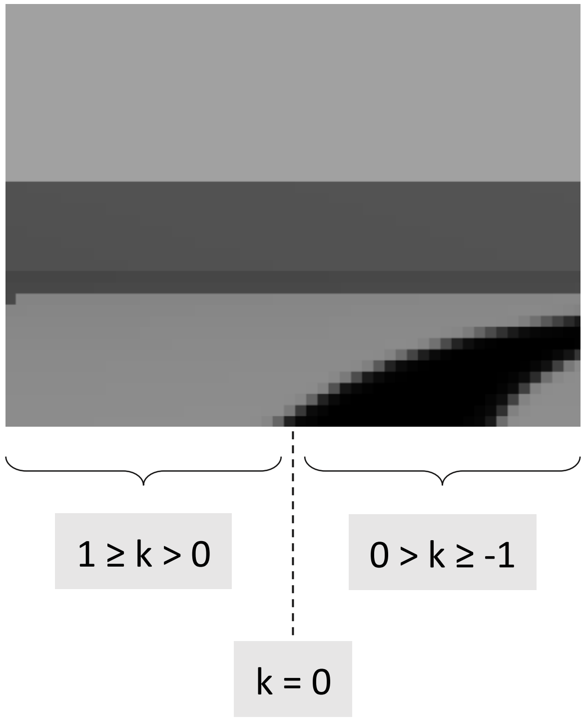
Line searching and following



- proportional controller

$$k = \frac{m_i - m_l}{m_i}$$

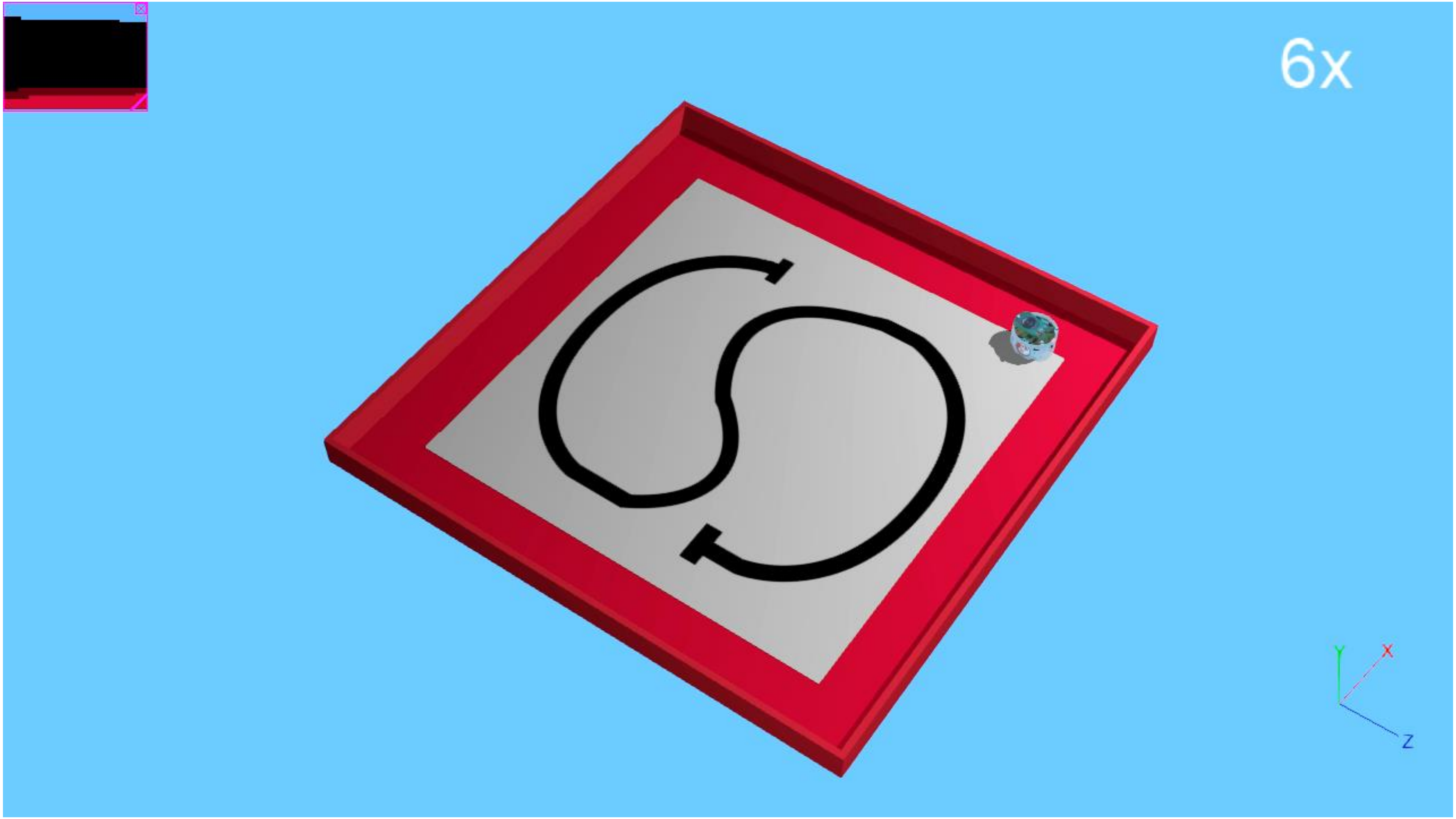
m_i = middle of the image
m_l = middle of the line



- line following: bottom row (38th)

$$v_{\text{left}} = \text{offset} - 35\,k_{br} - 30\,k_{br}{}^{3} - 25\,k_{br}{}^{5} - 15\,k_{br}{}^{7}$$
$$v_{\text{right}} = \text{offset} + 35\,k_{br} + 30\,k_{br}{}^{3} + 25\,k_{br}{}^{5} + 15\,k_{br}{}^{7}$$

- line searching: 18th and 32nd rows


6x
Y
X
Z

Obstacle avoidance

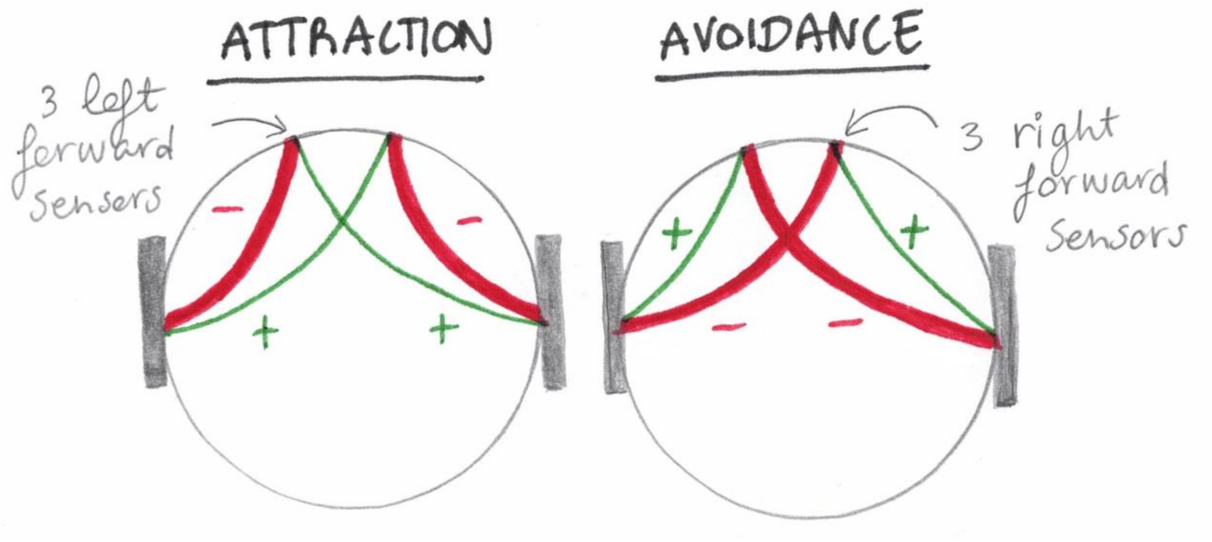
Stage 1 (line searching) ⟶ basic obstacle avoidance

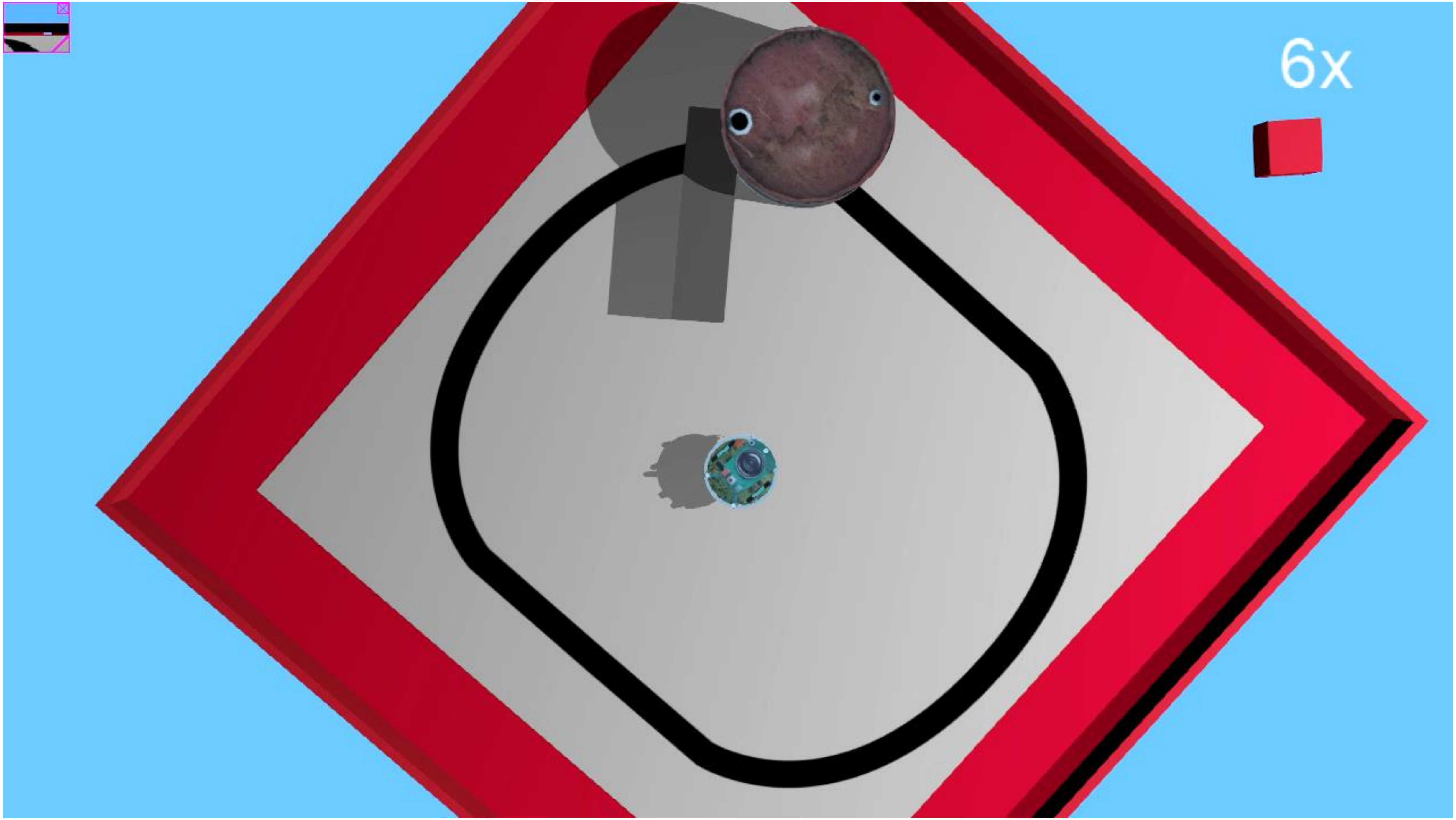
- Braitenberg avoidance

Stage 2 (line following) ⟶ obstacle circumvention

Alternation between:

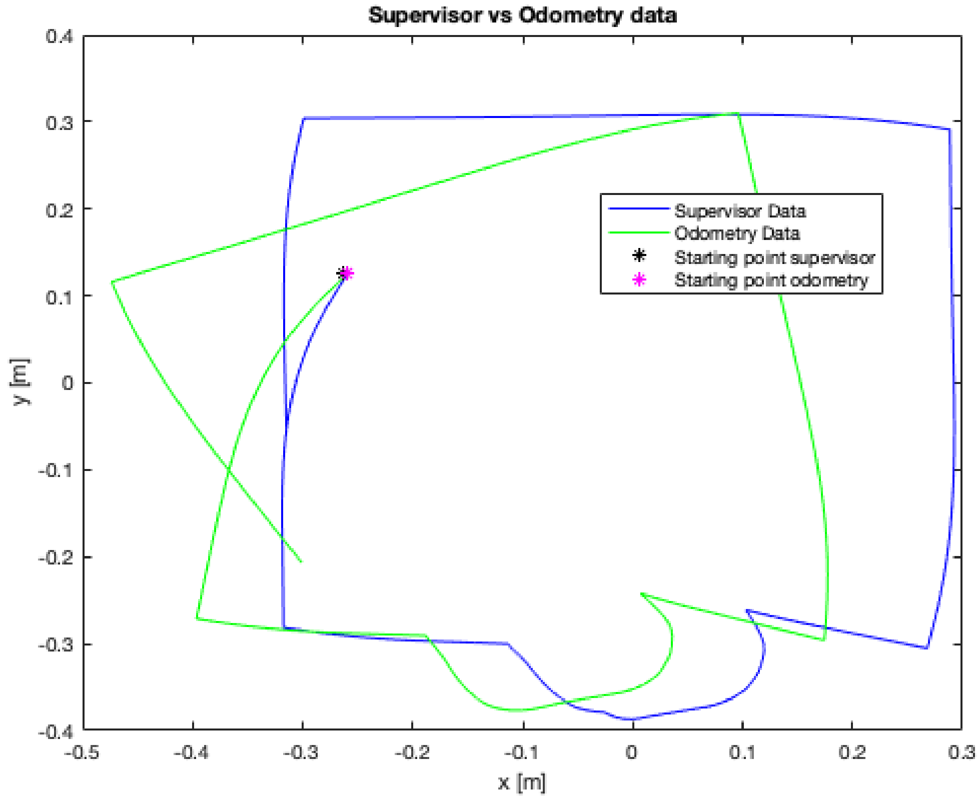
- Braitenberg avoidance
- Braitenberg attraction
- straight motion
- rotations




6x

Supervisor

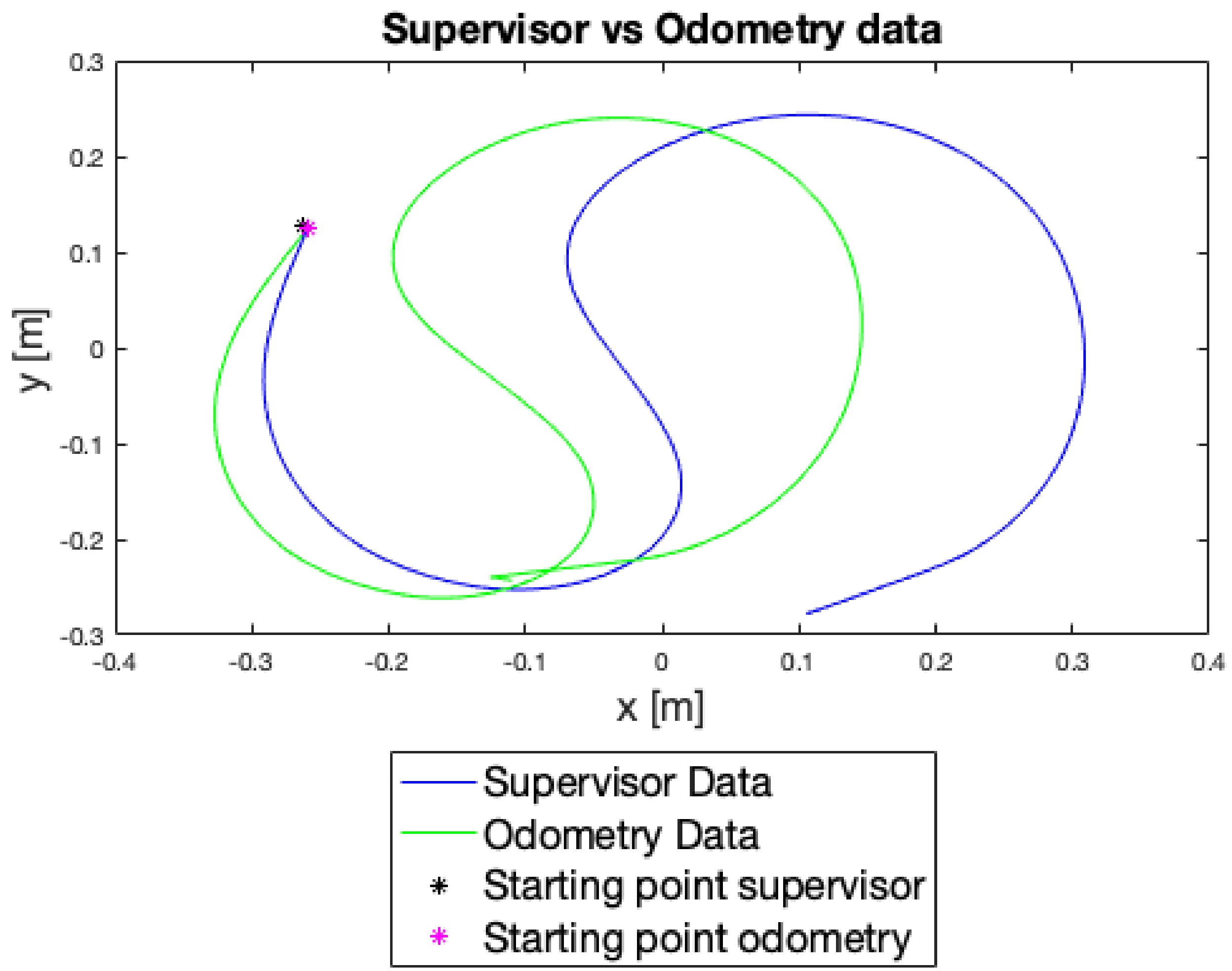
- Position in the global coordinate system
- Ground-truth trajectory effectuated by the robot
- `wb_supervisor_node_get_position(EPUCK)`

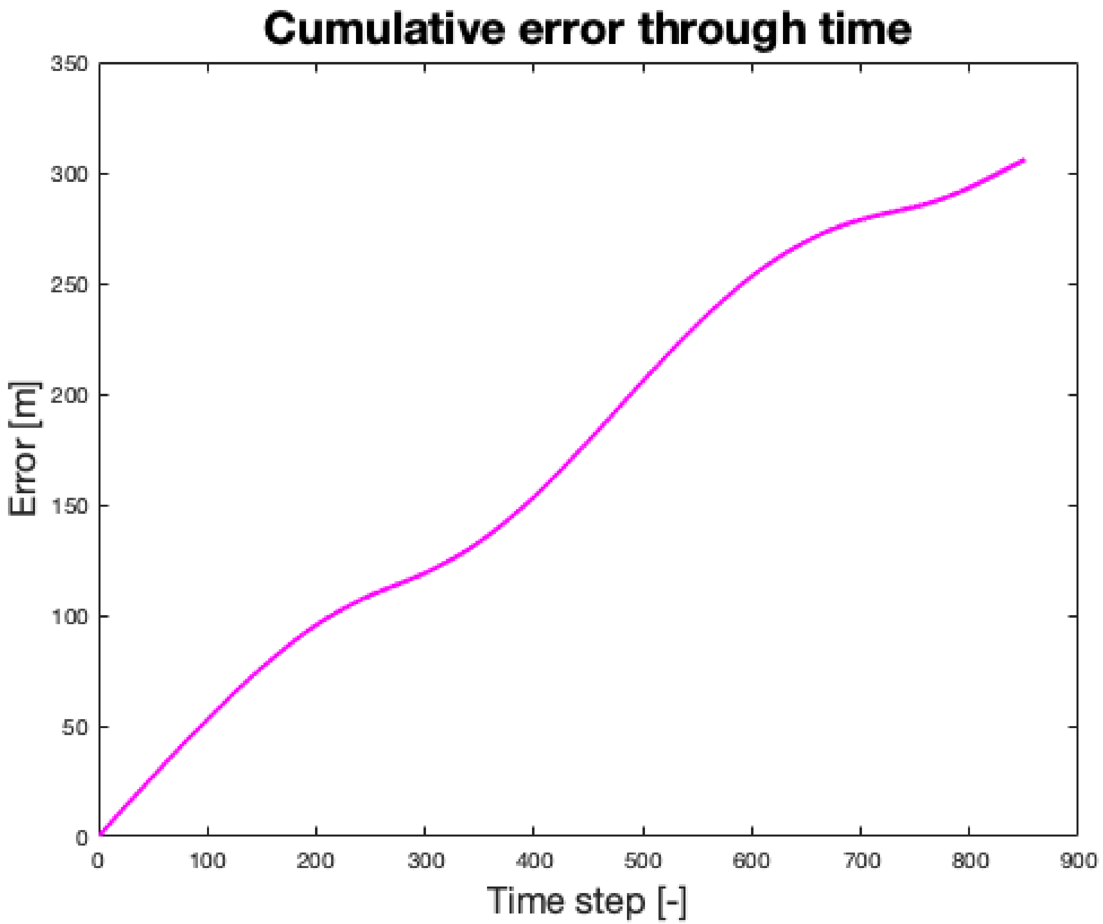


Odometry

- Data from motion sensors and initial position
- Conversion from the robot coordinate system to the global one
- Translation of wheel encoder readings into linear motion

Results


Supervisor vs Odometry data
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
y [m]
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
x [m]
Supervisor Data
Odometry Data
Starting point supervisor
Starting point odometry


Cumulative error through time
350
300
250
200
150
100
50
0
Error [m]
0
100
200
300
400
500
600
700
800
900
Time step [-]

Experiments

- Three experiments on the 4 line shapes in order to evaluate
  - Obstacle circumvention **n°1**
  - Line reaching **n°2**
  - Obstacle avoidance **n°3**

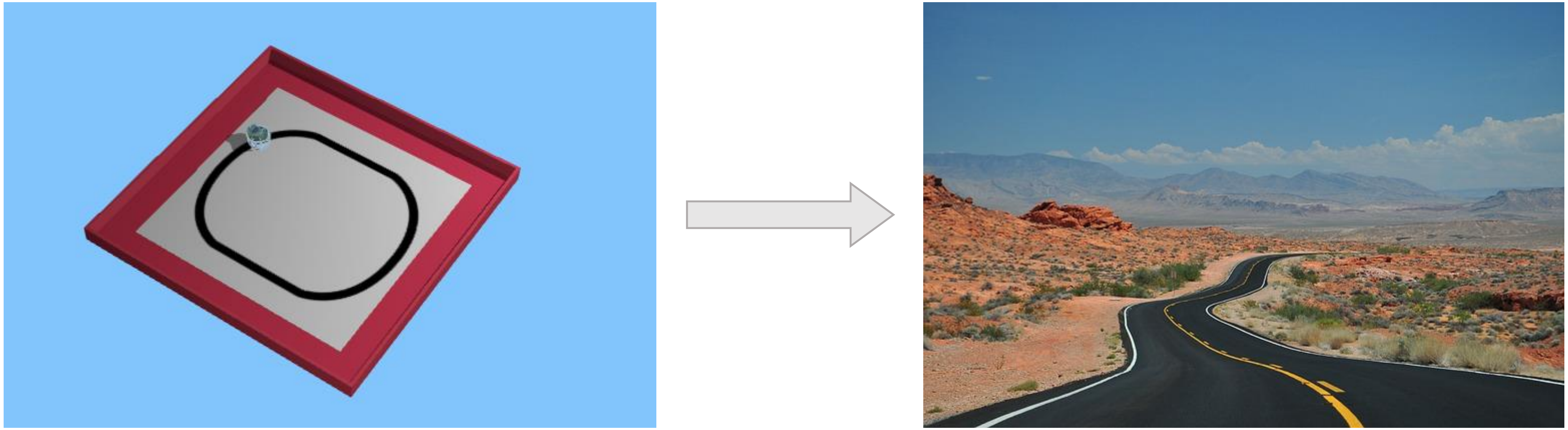
Experiment	n°1	n°2	n°3
Success [%]	63	78	73
Failure [%]	37	22	27
Tries	57	60	60

Weaknesses and improvements

- **Constraints linked to the robot**
  - Noise on distance sensors → low-pass filter
  - Delay between the camera image and the robot position
- **Obstacle circumvention**
  - Alternance of attraction and avoidance behaviours → single set of Braitenberg coefficients mixing attraction and avoidance depending on the distance sensor
- **Line acquiring**
  - Shadows → enhancing the `line_or_shadow` function

From Webots to reality

- Perception-to-action loop: delays!
- Memory much more limited
- Odometry errors more important (wheel diameters, misalignement…)



Conclusion

Thank you for your attention!