Project of Autonomous robotics module

Sepideh Hadadi, MSCV program semester 2, 2017
Center universitaire Condorcet, Le Creusot, France

Lab 1: Introduction of E-puck simulator

1.1. Introduction and practice

- The Mobile Robotics Simulator Software WEBOTS was explored with E-puck robot and instruction of the lab was followed. The e-puck educational robot was studied using handout provided with the course presentation. It was fully understood and the program introduced in Lab 0 was implemented and tested.
- The sensors and actuators of the mobile robot was read and stored in the file and analysed. The calibration was implemented and executed and the calibration coefficient was found. And example code is attached for reviewing and evaluation. In this part of the lab, e-puck's sample and instruction handout were very useful. The following two sections was explored and coded due to its important in the next stages.
  - Sensors: IR Sensors
  - Actuators: motors, LEDs

A program was implemented that makes the robot follow the wall and stays approximately 1cm away from it. The example video is attached. So I basically used the IR7, IR0, IR1 and IR2 sensors to keep the robot on the right path. So, in every iteration, the values given by the sensors were checked and according to these values and the command was corrected the path of the e-puck. The architecture of the LEDs is shown in Fig.1.

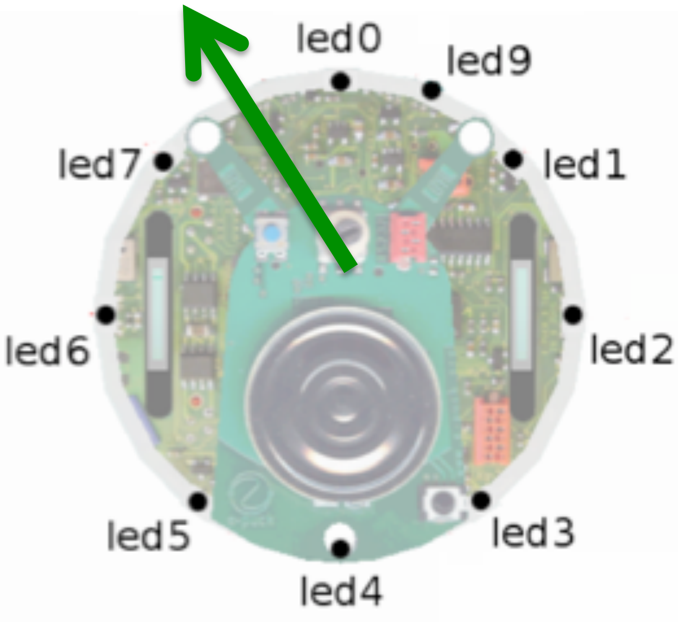

Fig.1 arrangement of the sensors and motion direction

1.2. To-Do tasks and implementation

1) Forward motion was implemented which drive robot forward and stop at 2 cm of the wall
2) Wall following approximately 1cm and the video is attached.

The test setting in the WEBOTS is shown in Fig.2. this world was kept constant for almost all the controller test.

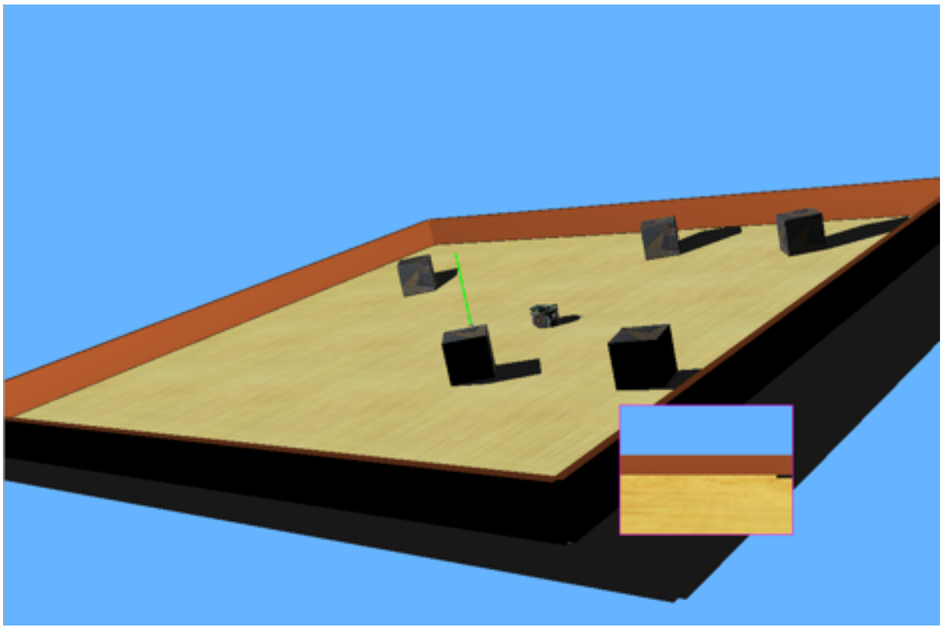

Fig.2 test setup for controller evaluation in the WBOTS

Lab 2: Odometry

2.1. Theory and review

In the section, the odometry concept will be reviewed and implementation will be discussed. After each implementation, the result will be presented or questions will be answered. The Odometry is the use of data from the movement of actuators to estimate change in position over time. In the case of wheeled robots, the odometry data will be derived from the movement of each wheel versus time. The change in position will depend on the rotation of the wheels, on the radius of the wheel and the width between wheels. Fig.3 shows all the relevant information necessary to compute the odometry.

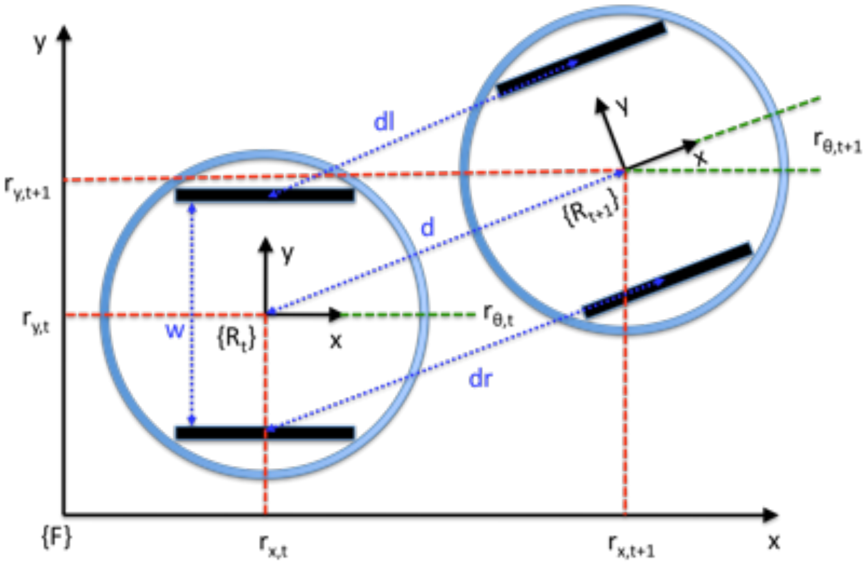

Fig.3 odometry calculation in each time stamp

Before starting odometry calculation let's define following parameters:

$\{F\}$ Fixed coordinate system

$\{R_t\}$ Robot coordinate system at time t

$\{R_{t+1}\}$ Robot coordinate system at time $t+1$

$r_t = (r_{x,t}, r_{y,t}, r_{\theta,t})$ is position and orientation of $\{R_t\}$

$r_{t+1} = (r_{x,t+1}, r_{y,t+1}, r_{\theta,t+1})$ is position and orientation of $\{R_{t+1}\}$

w width between the wheels

Differential distance dr is travelled by right wheel

Differential distance dl travelled by left wheel

Distance d is the distance robots move between $\{R_t\}$ and $\{R_{t+1}\}$

We can compute the next state r_{t+1} using following equations:

$$d = (dr + dl) / 2$$
$$r_{\theta,t+1} = r_{\theta,t} + (dr - dl) / w$$
$$r_{x,t+1} = r_{x,t} + d * cos(r_{\theta,t+1})$$
$$r_{y,t+1} = r_{y,t} + d * sin(r_{\theta,t+1})$$

2.2. To-Do tasks and answers to the questions

The requested exercises were implemented in the e-puck simulation.

This lab is developed based on the e-puck world and controller from the previous lab session. The measures of the encoders are converted to distances travelled by the left and right wheels. The result printed out to be seen in the debug windows. The odometry is computed and robot moves based on that computation. The robot is initialized the position of the robot to r_0 =(0,0,0) at the beginning of the simulation and update it according to the change in the encoder values. The calculation is summarized in the above introduction.

Lab 3. Obstacle avoidance

The obstacle detection was implemented and the result video is attached. In this implementation robot will depart from a given point and will return to the same point. The video of the activities for one robot and one obstacle is attached. The test setting for the obstacle detection is shown in Fig.4.

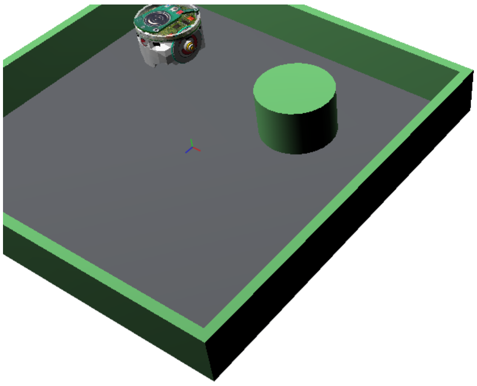

Fig.4 part of the setting used for obstacle detection