Series 02 – Ground Sensor, Wall following & PID

Robotics, BSc Course, 2nd Sem., Dr. Julien Nembrini, Marta Visetti

Handout on March 10th 2025 Due on March 23nd 2025

Reading
Study the lecture notes and source code available on Moodle.

Sections to be completed in the template report

— *Section 2.2 Sensors → Infra-red ground sensor*
— *Section 2.4 Sensors → Additional sensor analysis*
— *Section 3.2 Behaviours → Line-following*
— *Section 3.3 Behaviours → Wall-following*

1 Ground sensor response

On a *real* robot, record the ground sensor response when the robot crosses lines drawn on the ground (Fig 1)

Colored lines make the robot cross perpendicularly and diagonally (the robot trajectory making an approximate angle of 45° with the lines).
Gray lines make the robot cross perpendicularly.

Use the provided controller `S02_ground_record.py` and python script `S02_ground_plot.py` on the colored and gray lines available in the robotics lab. Present the graphs and discuss the results, especially the possibility to distinguish the different lines. Do not forget to mention the robot number and fill in the Section *2.2 Sensors → Infra-red ground sensor* of the report template.

2 Line following

For a single real e-puck, design, implement, test and comment a behaviour that follows a thick black line on the arena.

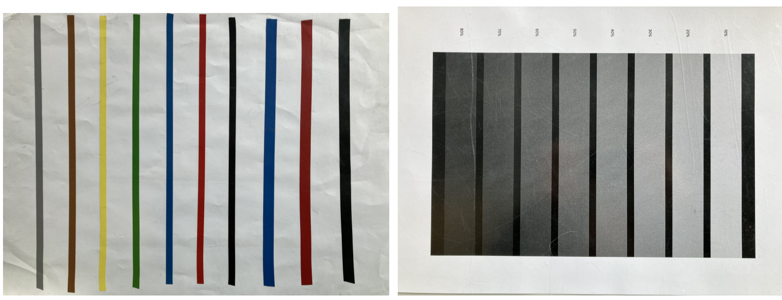

FIGURE 1 – Lines to be crossed for recording ground sensor response : colored lines (left), gray lines (right)

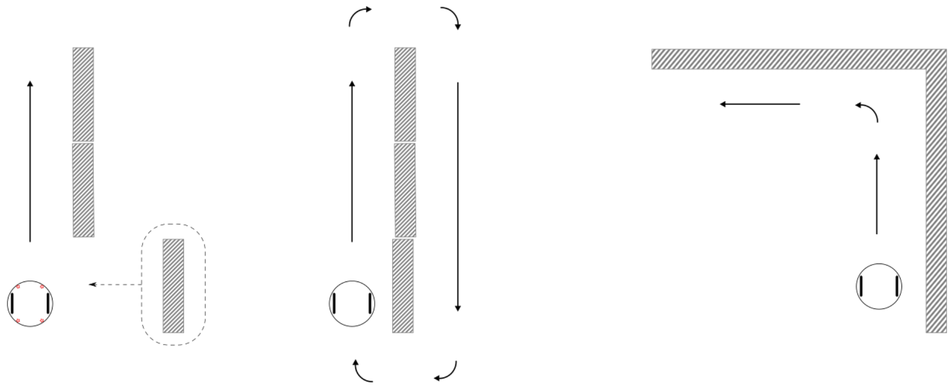

Figure 2 – PID optimization procedure

Follow these steps :

a) Implement a case-base (non-braitenberg) line-following behaviour. Using the suggestion from the course (2 sensors on the line, one sensor outside), consider all sensor situations and the corresponding reaction to stay on the line.
b) Your line-following behaviour should work for both the hexagonal and the right-angled lines available in the robotics lab.
c) Record a video and provide a link to it in your report.

Include your presentation and discussion in Section *3.2 Behaviours → Line-following* of the report template.

3 PID optimization

On a single real e-puck, optimize a wall following PID using the semi-complete implementation provided. At the beginning of the controller you will notice the definition of "General parameters" and "PID parameters". The former define the weights of the different sensors, the desired setpoint, etc. You may change these values, but in such case explain it in your report. The "PID parameters" K and T_d are the values you have to tune to implement the wall following behaviour.

As in Figure 2, your setup should be a straight line of three blocks, which the robot has to circle around (center image). When you start the robot, keep the first block away for the calibration step (left image).

Follow this procedure :

a) With the T_d and T_i values as already defined in the controller, find a good K value that makes the robot approximately follow the wall (hint : $K < 0.1$).
b) Let the robot circle two times around the wall before stopping it. The controller records data.
c) Use the `S02_PID_plot.py` script to understand if your parameter is approximately good or not. Observe particularly the scale of the ds value which should be comparable to acceptable motor speed values.
d) Change the parameter and let again the robot circle two times around the wall before stopping it.
e) Use the `S02_PID_plot_compare.py` script to compare the performance of the new parameter values with the previous one(s). The spread of the speed differential ds should be as small as possible.
f) start again at d) to find the optimal parameter value following bisection optimisation by running the robots with different parameter values, using the `S02_PID_plot_compare.py` script to decide which value is better (goal : limit the spread of the speed differential ds). Be consistent with the number of circles (2 recommended) to allow meaningful comparison. You may delete/rename the `logPID_X.csv` files to reorder the results on the plot (the script functions until $X = 9$).
g) Once the parameter K is optimal, optimize the T_d parameter following the steps d), e) and f) above.

Important :

— You are not requested to optimize the T_I parameter

— The optimized parameters may be dependent on the robot, try to use always the same one.

When the wall following behaviour is optimized, test it with an arena corner (right image above) and note the robot behaviour. Record a video and provide a link to it in your report. Describe the results of your optimization (including the comparison graphs) and discuss your result in Section 3.3 Behaviours → Wall-following of the report template.

4 Additional sensors analysis

On a *real* robot, choose **one** sensor type among the list below :

— Acceleration

— Gyroscope

— Time-of-Flight

— Microphones

Follow these steps :

a) define a testing setup to record an interesting sensor response

b) run test(s) with a real robot

c) describe your testing setup, plot and discuss the results

d) imagine and describe the use of the sensor in a controller

Use the provided controller `S02_add_sensors_record` and python script `S02_add_sensors_plot.py`. Present the graphs and discuss the results (mention the robot number) in Section *2.4 Sensors → Additional sensor analysis* of the report template.