LINE FOLLOWING USING THE E-PUCK’S CAMERA

Signals, Instruments and Systems
Course Project

FEVRE Pierre, HAUTEFAYE Elise, MAIRE Loé
Professor : MARTINOLI Alcherio
Supervisor : ERCOLANI Chiara
PLAN

• Introduction
• Webot
• E-puck
• Method
• Experiments and results
• Conclusion
INTRODUCTION

Goals:

- Follow the line and about-turn
- Obstacle avoidance
- Find the line

On webot and e-puck
C programming language and signal processing
Camera: 52*39 RGB
- Black pixels are [0,0,0] in RGB
- Matrix form

8 infrared proximity sensors
- sensor_value[i]
Camera : 40*40 RGB :
- Black pixels between 15 and 75
- Vector form [0 : 1600]

8 infrared proximity sensors

Webot to e-puck :
- Same based concept
- Different libraries
- Adapation of threshold values
Switch function between obstacle avoidance and line following:

If (sensor[i] > 6000)

{ operation_mode = OBSTACLE;
  break;
}
else operation_mode = LINE;

Case obstacle avoidance:
- Using Braitenberg equation

\[
\text{Speed}_\text{left} += 2 \times \text{braitenberg}\_\text{coefficient}[j][0] \times (1 - \text{sensor}[j] / \text{RANGE})
\]
\[
\text{Speed}_\text{right} += 2 \times \text{braitenberg}\_\text{coefficient}[j][1] \times (1 - \text{sensor}[j] / \text{RANGE})
\]
Case line following:
- Analyze of two squares of 5*5 pixels
- Counting black pixels with 2 for loops

About-turn:
- Analyze of an entire line
- If no black pixels: make an about-turn

Speed of wheels:
If (pixel_black_left > pixel_black_right)
{
    left_speed = 0.7*speed;
    right_speed = speed;
}
end
Find the line:
- Analyze of a line at the top: long distance
- Analyze the 2 squares at the bottom: short distance
- Speed: two times higher than before
EXPERIMENTS AND RESULTS

Different shapes:
- Find the line easily with S shape
- Follow the line easily with O shape

Different colors:
- Similar results

Different line width:
- About-turn easily with wider line
- Follow the line easily with wider line
Obstacle avoidance:
- Detect it regardless line width
- About-turn easily with wider line

The e-puck doesn’t get around the obstacle
- Make an about-turn
- Follow the line
PROBLEMS ENCOUNTERED

About-turn :
  - the e-puck may detects an end of line when it’s on a continue line

Find Line :
  - the e-puck search the line only at the beginning of the program

Avoid obstacle :
  - the e-puck does an about-turn in front of an obstacle
CONCLUSION

- Need to test different algorithms
- Behavior changing according to e-puck
- Debugging and improve the state machine

Improvements:
- Optimization of the code (e-puck’s memory restricted)
- More IR sensors on the e-puck
- Fourier transform and filter