E-PUCK IN A MAZE NAVIGATION WITH SIMPLE ROAD SIGNS

Sara Beaudet, Jael Locher, EPFL
Goals

Challenge the capacity of the e-puck robot to:

- Exit a simple maze while avoiding obstacles
- Analyse simple road signs & take action of movement
- Estimate its position through odometry
- Compare with position data from a supervisor
- Software simplicity and efficiency

Tools

- E-puck robot (simulated)
- Webots simulation program
- Matlab → image processing

Turn 180°  Turn left 90°  Turn right 90°

E-puck – proximity sensors
Methods:

Controller design:
- Obstacle avoidance with Braitenberg controller
- `wb_robot_step`

Odometry:
- Correction factor

<table>
<thead>
<tr>
<th>Table I. Braitenberg Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>proximity sensors</td>
</tr>
<tr>
<td>wheel left</td>
</tr>
<tr>
<td>wheel right</td>
</tr>
</tbody>
</table>
Methods:

Data processing:
- Sampling the image matrix by columns & rows
Methods:

Data processing:
- Sampling the image matrix by columns & rows
- Apply the kiss_fft algorithm
Methods:

Data processing:
- Sampling the image matrix by columns & rows
- Apply the kiss_fft algorithm
- Compute the weighted sum on each vector

\[ SUM = \sum_{n=0}^{N/2} f_n \cdot A_n \]

- **SUM**: weighted sum
- **N**: Number of Fourier coefficients
- **f_n**: n-th frequency
- **A_n**: amplitude of the n-th frequency
Methods:

Data processing:
- Sampling the image matrix by columns & rows
- Apply the kiss_fft algorithm
- Compute the weighted sum
  \[ SUM = \sum_{n=0}^{N/2} f_n \cdot A_n \]
- Take the decision of movement

\[ \text{if } (SUM_{col} = SUM_{rows} = 0) \]
\[ \text{turn around} \]

\[ \text{else if } (SUM_{col} > SUM_{rows}) \]
\[ \text{turn left} \]

\[ \text{else} \]
\[ \text{turn right} \]
Experiments

Test 1

START

END
Experiments
Results test 4
Results test 2
Results & Discussion

<table>
<thead>
<tr>
<th>Distance traveled [m]</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.592</td>
<td>6.001</td>
<td>5.788</td>
<td>20.972</td>
</tr>
<tr>
<td>Duration [s]</td>
<td>58.944</td>
<td>139.392</td>
<td>131.520</td>
<td>484.064</td>
</tr>
<tr>
<td>Average velocity [m/s]</td>
<td>0.044</td>
<td>0.044</td>
<td>0.044</td>
<td>0.043</td>
</tr>
<tr>
<td>Number of decisions</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>23</td>
</tr>
<tr>
<td>Nb. of correct decisions</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>23</td>
</tr>
<tr>
<td>Number of loops</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>RMSE X</td>
<td>0.0051</td>
<td>0.0237</td>
<td>0.0129</td>
<td>0.0965</td>
</tr>
<tr>
<td>RMSE Y</td>
<td>0.0079</td>
<td>0.0543</td>
<td>0.0080</td>
<td>0.0668</td>
</tr>
</tbody>
</table>

RMSE between the supervisor’s coordinates and the coordinates collected by the robot by odometry.
Conclusion

Achievements:
- E-puck correctly analyses road signs
- Successfully exits the maze

Improvements needed:
- for odometry and controller design
- on 180° turns and unconstrained paths

Next steps:
- Test the hardware e-puck
- Cope with the limited RAM capacities
THANK YOU FOR YOUR ATTENTION!

Questions?
Methods:

Position estimation by odometry:
- relative position of the e-puck
- global position of the e-puck
- correction factors

\[ \xi_I = \begin{bmatrix} x_I \\ y_I \\ \theta_I \end{bmatrix} \begin{bmatrix} \frac{R}{2} \sin(\dot{\varphi}_1 + \dot{\varphi}_2) \\ \frac{R}{2} \cos(\dot{\varphi}_1 + \dot{\varphi}_2) \\ \frac{R}{2} (\dot{\varphi}_1 - \dot{\varphi}_2) \end{bmatrix} \cdot \Delta t + \begin{bmatrix} x_{I,\text{previous step}} \\ y_{I,\text{previous step}} \\ \theta_{I,\text{previous step}} \end{bmatrix} \]

- \( \xi_I \): position in the inertial coordinates system
- \( R \): wheel's radius
- \( \varphi_1 \): rotational speed of the right wheel
- \( \varphi_2 \): rotational speed of the left wheel
- \( l \): distance between the wheels and the center of the robot
- \( \Delta t \): time step
- \( \xi_{I,\text{previous step}} \): inertial coordinates from the previous step

Supervisor:
- global position of the e-puck
- time to exit the maze
- Pause and reset the simulation after exit.
Results & Discussion

given

checked

given

unknown

<table>
<thead>
<tr>
<th>no noise</th>
<th>low noise</th>
<th>medium noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>stripes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>black</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>