Finding a light source with an epuck robot

Fanny Gretillat, Jean-André Davy-Guidicelli, Leily Moser
Braitenberg model

\[ speed_{left} = speed_0 + \sum_{i=0}^{n} \alpha_{left,i} \left(1 - \frac{value_i}{range}\right) \]

\[ speed_{right} = speed_0 + \sum_{i=0}^{n} \alpha_{right,i} \left(1 - \frac{value_i}{range}\right) \]

with \( speed_0 \) : initial speed, \( n \) : number of sensors, \( \alpha \) : weight of the \( i^{th} \) sensor, \( value \) : measured value by the \( i^{th} \) sensor, \( range \) : range of the sensor values.
Braitenberg coefficients

Wall following:

<table>
<thead>
<tr>
<th>Sensor number</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left wheel</td>
<td>120</td>
<td>75</td>
<td>-30</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>-90</td>
<td>-180</td>
</tr>
<tr>
<td>Right wheel</td>
<td>-180</td>
<td>-90</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>-30</td>
<td>75</td>
<td>120</td>
</tr>
</tbody>
</table>

Wall corner:

<table>
<thead>
<tr>
<th>Sensor number</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left wheel</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>-100</td>
<td>-100</td>
<td>-100</td>
<td>-100</td>
</tr>
<tr>
<td>Right wheel</td>
<td>-125</td>
<td>-100</td>
<td>-100</td>
<td>-100</td>
<td>100</td>
<td>100</td>
<td>75</td>
<td>100</td>
</tr>
</tbody>
</table>

Light following:

<table>
<thead>
<tr>
<th>Sensor number</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left wheel</td>
<td>0</td>
<td>15</td>
<td>330</td>
<td>460</td>
<td>-460</td>
<td>-330</td>
<td>-15</td>
<td>0</td>
</tr>
<tr>
<td>Right wheel</td>
<td>0</td>
<td>-25</td>
<td>-300</td>
<td>-440</td>
<td>440</td>
<td>300</td>
<td>25</td>
<td>0</td>
</tr>
</tbody>
</table>

Wall turning:

<table>
<thead>
<tr>
<th>Sensor number</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left wheel</td>
<td>250</td>
<td>200</td>
<td>40</td>
<td>-15</td>
<td>-20</td>
<td>-65</td>
<td>-75</td>
<td>-90</td>
</tr>
<tr>
<td>Right wheel</td>
<td>-105</td>
<td>-70</td>
<td>-65</td>
<td>-15</td>
<td>-15</td>
<td>40</td>
<td>190</td>
<td>265</td>
</tr>
</tbody>
</table>
Obstacle detection

Condition:

\(\text{ps\_value} > 250\)
Light following

Condition:

ps_value < 250
Video
Obstacle avoidance

Classical obstacle avoidance

Avoidance by wall following
Turn to be aligned

Wall turning  Wall following
Wall following

Condition:

Wall corner

Condition:

\[ \text{ps\_value}[0] + \text{ps\_value}[7] > 600 \]
Wall end

Condition:

\[(\text{ps\_value}[5] \text{ OR } \text{ps\_value}[2] > 1000) \text{ AND } (\text{ps\_value}[1] \text{ AND } \text{ps\_value}[6] < 300)\]
Dead End

Condition:

\((\text{ps\_value}[1] \ \text{AND} \ \text{ps\_value}[6] > 300) \ \text{AND} \ (\text{ps\_value}[0] \ \text{AND} \ \text{ps\_value}[7] > 200)\)
Simulation in Webots
Adaptation to real e-puck

- Adaptation of the ranges

- Adaptation of the limit distance value for obstacle detection

- Obstacle’s surface influence

- Light not always detected
Limits and possible improvements

• Limits:
  - Works only for vertical walls
  - Not efficient for cylindrical obstacles
  - Importance of the surface of the obstacle
  - The light source has to emit IR radiation

• Possible improvements:
  - Adjust the code for other obstacle shapes and incline
  - Adapt the range of the sensor values at each iteration
  - Implement an end when the robot finally reaches the light
Video