Distributed Intelligent Systems

Aurélien Brun, Raphael Renevey, Malo Simondin
Controller description

- Flock organized in **leader-follower** fashion => Line formation

- One **leader** sets direction with **migratory urge**, while **avoiding obstacles**

- The **Follower** with ID n follows the e-puck with robot ID n-1, while avoiding obstacles
Finite State Machine: Leader

IR Sensors: IR values
- Value > threshold
  - Obstacle Avoidance State
  - Orientation Controller
  - Lateral Displacement Controller
  - Odometry

IR Communication: Communication with follower
- Value < threshold
  - Straight State
  - Orientation Controller
  - Lateral Displacement Controller
  - Odometry

Braitenberg Controller: v left, v right
Motor Actuators: v left, v right
Finite State Machine: Follower

- IR Sensors
  - IR values
  - Transition Rule
    - value > threshold
    - value < threshold
  - Communication with follower
    - Send
    - Receive

- Obstacle Avoidance State
- Follow State
  - Leader Orientation Controller
  - Leader Distance Controller
    - v left
    - v right
  - Leader Range and Bearing

- Braitenberg Controller
  - v left
  - v right

- Motor Actuators
Navigation: Leader Controller

Obstacle Avoidance:

\[
velocity_{left, right} = KP_B \times \sum b_i \times \left(1 - \frac{IR \text{ sensor}_i}{Range}\right) \pm KP_{\theta - oa} \times \Delta \theta
\]

Straight:

\[
velocity_{left, right} = SV \pm KP_{\theta - s} \times \Delta \theta \pm KP_y \times y
\]

KP: proportional controller constants
SV: straight velocity
\(\Delta \theta\): difference in leader orientation regarding its initial one
y: lateral displacement of the e-puck regarding its initial position
Navigation: Follower Controller

Obstacle Avoidance:

\[ velocity_{left, right} = KP_B \sum b_i \left( 1 - \frac{IR \ sensor_i}{Range} \right) \pm KP_{\theta-oa} \Delta \theta \]

Straight:

\[ velocity_{left, right} = SV \pm KP_{\theta-s} \Delta \theta \pm KP_u (u - d_{leader}) \]

KP : proportional controller constants
SV : straight velocity
\( \Delta \theta \) : difference in orientation between the leader and the follower
u : distance between leader and follower
d_{leader} : wanted distance between leader and follower
Braitenberg Tweaking for Followers

- The follower controller ignores sensor values in the direction of nearby e-pucks
- Intervals of orientation are defined for each sensor
- A distance threshold is defined

=> Avoids e-pucks to see each-others as obstacles
Webots Simulation:

- Why?
  - Crucial to testing and evaluation our concepts

- Performance
  - A supervisor computes performance according to 3 metrics

- Communication
  - Infinite range, unlimited bandwidth, accurate

- Sensors
  - No noise
Scenario 1: Flock avoids obstacles

- 4 additional worlds
- Different obstacle shapes
- Various obstacle orientations
- Denser obstacles
Webots simulation: World 2
Scenario 2: flock crossing

World 1

Frontal crossing

World 2

Perpendicular crossing
Crossing Scenario: Frontal collision

Speed x3
Crossing Scenario: Perpendicular collision

Speed x3
Performance Evaluation: Results

<table>
<thead>
<tr>
<th></th>
<th>World 1</th>
<th>World 2</th>
<th>World 3</th>
<th>World 4</th>
<th>World 5</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Fitness</td>
<td>0.24</td>
<td>0.16</td>
<td>0.21</td>
<td>0.22</td>
<td>0.20</td>
<td>0.22</td>
</tr>
<tr>
<td>Cohesion</td>
<td>0.81</td>
<td>0.83</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
</tr>
<tr>
<td>Orientation</td>
<td>0.99</td>
<td>0.88</td>
<td>0.95</td>
<td>0.95</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>Velocity</td>
<td>0.34</td>
<td>0.21</td>
<td>0.26</td>
<td>0.27</td>
<td>0.25</td>
<td>0.27</td>
</tr>
</tbody>
</table>

❖ Performance: Supervisor assessment

❖ **Cohesion**: high due to strong urge in proportional controller

❖ **Orientation**: Very high due to line formation

❖ **Velocity**: Relatively low due to low speed

❖ Performance: Visual assessment

❖ **Obstacle Avoidance**: efficient. No bumping into walls

❖ **Migratory urge**: Well balanced, no conflict with obstacle avoidance
Real Robots: Implementation Challenges

- Adaptation to e-puck library
- Range & bearing: libIrcom functions
- Obstacle avoidance: sensor sensitivity & noise
- Imperfect communication
Real World Setup

❖ An arena was created to test our controllers

❖ Different materials tested: wood, cardboard and paper

❖ Tested different obstacle scenarios and one frontal crossing scenario
Obstacle scenario:

Speed x4
Crossing scenario:

Speed x4
Real Robots: Results

❖ Communication

❖ Noisy and inaccurate: messages not always received
❖ Discrete space coverage with proximity sensors
❖ Scalability: communication shouldn’t be a problem

❖ Performance (visual assessment)

❖ Odometry was accurate enough: migratory urge very satisfying
❖ Efficient obstacle avoidance
❖ Robots didn’t often lose teammates
Conclusion

❖ Project Recapitulation:

❖ Good teamwork: task & work-load repartition satisfying
❖ Webots simulation resilient to different worlds ; good performances
❖ Real e-pucks resilient to different scenarios

❖ Possible improvements

❖ Test scalability with more robots
❖ Implement: “When follower loses leader, leader waits for a while”
❖ Improve resilience to difficult obstacles such as dead-end
❖ Particle Swarm Optimization : find the right velocity and coefficients
Thank you for your attention. Questions?