Multi-robot navigation in cluttered and dynamic environments
Welcome!

Zeki Doruk ERDEN - Michael PERRET - Mickaël SALAMIN
Plan of the presentation

- Project aim, constraints and metrics
- Strategy
- Simulation
  - Setup & results
- Real robots
  - Implementation, experiments & results
- Conclusion
  - Improvements
Project & Strategy
Aims, constraints, metrics and choices
Project

- Multi-robot navigation in cluttered and dynamic environments
- Two scenarios
  - Static obstacles
  - Different groups crossing each other
- Simulation in Webots & implementation on real e-pucks
Constraints and metrics

### Constraints
- No global positioning
- Use of Range & Bearing module
- Simulation with a compass and relative position

### Metrics
- Normalized between [0 ; 1] <=> [worst ; best]
- *Orientation*: Measures the alignment between the robots,
- *Cohesion*: Measures the dispersion between the robots,
- *Velocity*: Measures the velocity of the center of mass along the direction of the migratory urge
Flock moving in formation

Reynolds’ rules

Graph-based method

Flexible implementation

Rigid implementation
Strategy

- Obstacle avoidance
- Braitenberg architecture

(a) Vehicle 2a  (b) Vehicle 2b  (c) Vehicle 3a  (d) Vehicle 3b

- Graph-based formation
- Combined with migratory urge
Strategy

Flowchart

- Odometry → Send ping → Range & Bearing → Reynolds → Braitenberg → Update speed

Infrared sensors
- Relative position + speed calculation

Wheel speed computation
- Reynolds rules
- Migratory urge
- Braitenberg
Simulation

Simulation setup and results
Obstacles Scenario

- Local neighborhood
- A baseline combination
- Varying single parameter at a time
## Weights

<table>
<thead>
<tr>
<th>CH</th>
<th>D</th>
<th>CS</th>
<th>M</th>
<th>FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>0.5</td>
<td>1</td>
<td>0.02</td>
<td>5</td>
</tr>
</tbody>
</table>

## Performance

<table>
<thead>
<tr>
<th>$o[t]$</th>
<th>$c[t]$</th>
<th>$v[t]$</th>
<th>$p[t]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9429</td>
<td>0.8543</td>
<td>0.6076</td>
<td>0.5055</td>
</tr>
</tbody>
</table>

- **CH**: Cohesion
- **D**: Dispersion
- **CS**: Consistency
- **M**: Migration
- **FS**: Flock Size
Dispersion and Consistency

- No observable effects
- Dispersion accounted for by obstacle avoidance
- Consistency accounted for by common migration urge
Cohesion Weight

- Higher cohesion:
  - Better flocking
  - Lower velocity
Migration Urge

- Higher migration:
  - Larger velocity
  - Worse flocking
Reliable operation for all sizes
Well scalable

<table>
<thead>
<tr>
<th></th>
<th>$o[t]$</th>
<th>$c[t]$</th>
<th>$v[t]$</th>
<th>$p[t]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.9429</td>
<td>0.8543</td>
<td>0.6076</td>
<td>0.5055</td>
</tr>
<tr>
<td>FS=3</td>
<td>0.9676</td>
<td>0.9171</td>
<td>0.6354</td>
<td>0.5734</td>
</tr>
<tr>
<td>FS=8</td>
<td>0.9554</td>
<td>0.8433</td>
<td>0.5938</td>
<td>0.4808</td>
</tr>
<tr>
<td>FS=12</td>
<td>0.9455</td>
<td>0.8473</td>
<td>0.5739</td>
<td>0.4629</td>
</tr>
</tbody>
</table>
Ten additional worlds for obstacle scenario
- Harder to navigate
Statistical Analysis of Performance

- Reliable operation observed with baseline
- Lower performances
  - Harder than original world

<table>
<thead>
<tr>
<th></th>
<th>( o[t] )</th>
<th>( c[t] )</th>
<th>( v[t] )</th>
<th>( p[t] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>0.9589</td>
<td>0.8841</td>
<td>0.5898</td>
<td>0.4937</td>
</tr>
<tr>
<td>Worst</td>
<td>0.9103</td>
<td>0.6942</td>
<td>0.5480</td>
<td>0.3796</td>
</tr>
<tr>
<td>Mean</td>
<td>0.9366</td>
<td>0.8136</td>
<td>0.5717</td>
<td>0.4413</td>
</tr>
<tr>
<td>Variance</td>
<td>0.0003</td>
<td>0.0041</td>
<td>0.0003</td>
<td>0.0017</td>
</tr>
</tbody>
</table>
Crossing Scenario

- Baseline weights perform good
Smaller flocks are better

Low chance of getting stuck

<table>
<thead>
<tr>
<th></th>
<th>$o[t]$</th>
<th>$c[t]$</th>
<th>$v[t]$</th>
<th>$p[t]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.9198</td>
<td>0.9002</td>
<td>0.5415</td>
<td>0.4655</td>
</tr>
<tr>
<td>FS=3</td>
<td>0.8683</td>
<td>0.9121</td>
<td>0.5713</td>
<td>0.4901</td>
</tr>
<tr>
<td>FS=8</td>
<td>0.6596</td>
<td>0.8521</td>
<td>0.4299</td>
<td>0.3288</td>
</tr>
</tbody>
</table>
Reynolds weights have no structural effect
- Performance depends on specific configuration – random
- Reynolds weights influence configuration
- Variation in performance: Not as robust as obstacles scenario
3

Real robots

Implementation, experiments, results
Implementation

- Adaptation to e-puck library
- Range and bearing: libIrcom
- Obstacle avoidance: sensor sensitivity
- Noisy environment: reduced speed
Experiments

- Two scenarios
  - Static obstacles: switched-off e-pucks
  - Mobile obstacles: two crossing teams
- Different starting configurations
Experiments – Static obstacles
Results

- Very short communication range
- Limited neighborhood radius
- Robots can rapidly lose their teammates
- Noisy, inaccurate and sparse decoded messages
- Transparent robots
- Discrete space coverage with proximity sensors
- IR signals make the flock hardly scalable
Results

Performance (visual assessment)
- Odometry was accurate enough
- Efficient obstacle avoidance
- Cooperative navigation
- Flocking behavior would be more evident with additional robots
Comparison with simulation

Communication
- Simulation: infinite range, unlimited bandwidth, accurate
- Real world: short range, sparse, noisy

Navigation
- Same method: Reynolds + Braitenberg
- Similar flocking behavior and obstacle avoidance
- Scalability: consistent in simulation, very hard in reality
Conclusion
Improvements and Applications

- Complex optimization due to interdependence of algorithm parameters
- Radio communication would make our solution more robust, reactive and scalable
- Still good results with IR emitters-receivers
- Simple environments compared to real applications
- Low-cost solution for multi-robot navigation in cluttered and dynamic environments
THANKS for your attention!

Any questions?