Distributed Intelligent Systems

Multi-robot navigation in cluttered and dynamic environments

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Group 10:

Mathilde Bensimhon Daniel Almeida Dias
Pauline Maury Laribière Hugo Grall Lucas

Professor: Alcherio Martinoli
H.T.A. : Duarte Da Cruz Baptista Dias
T.A. : Quraishi Anwar Ahmad

Contents

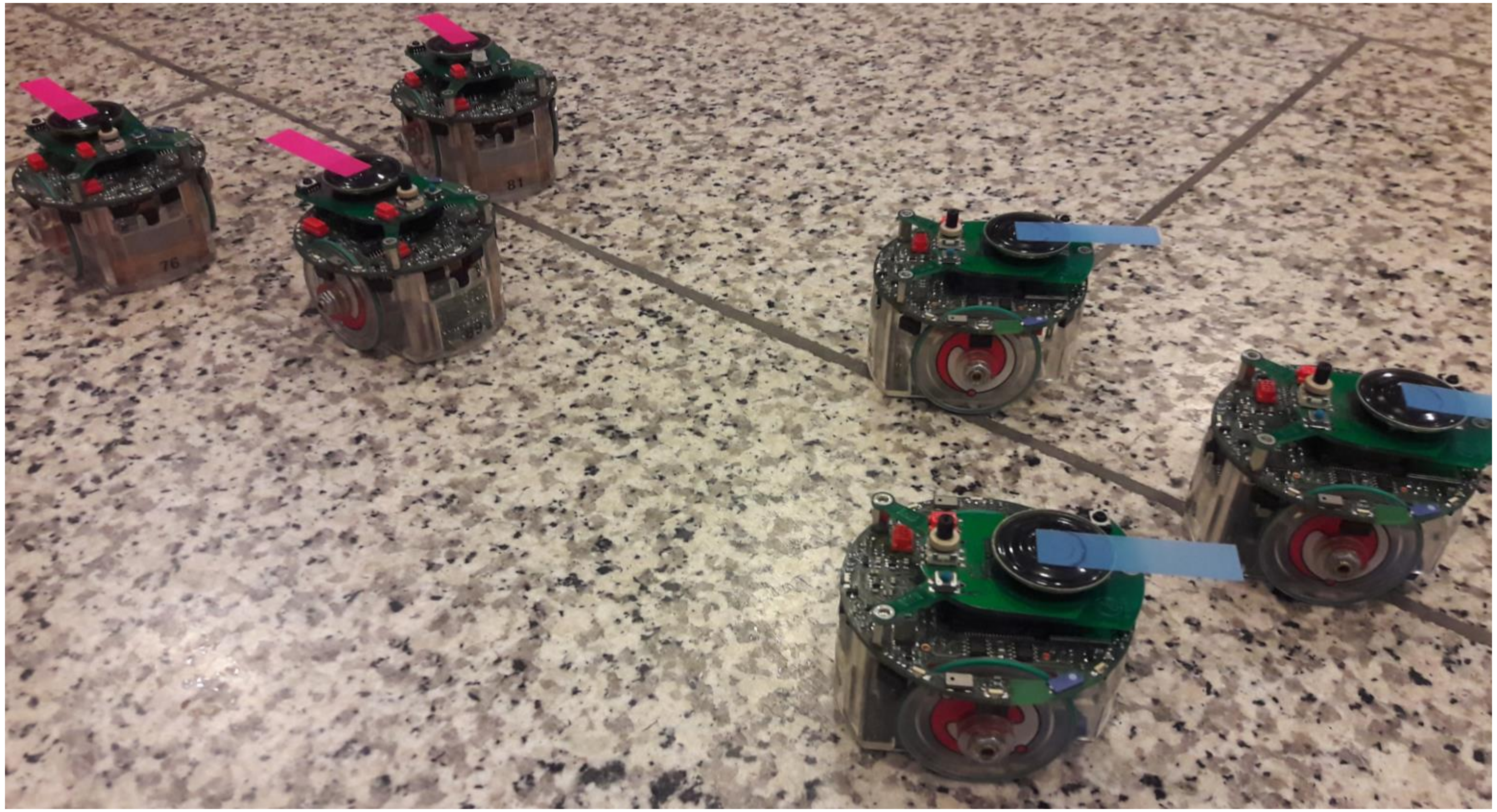
Controller architecture
- Controller overall
- Weights adapter
- Migration
- Braitenberg
- Reynolds
- Join

Simulation
- Scenario A
- Scenario B

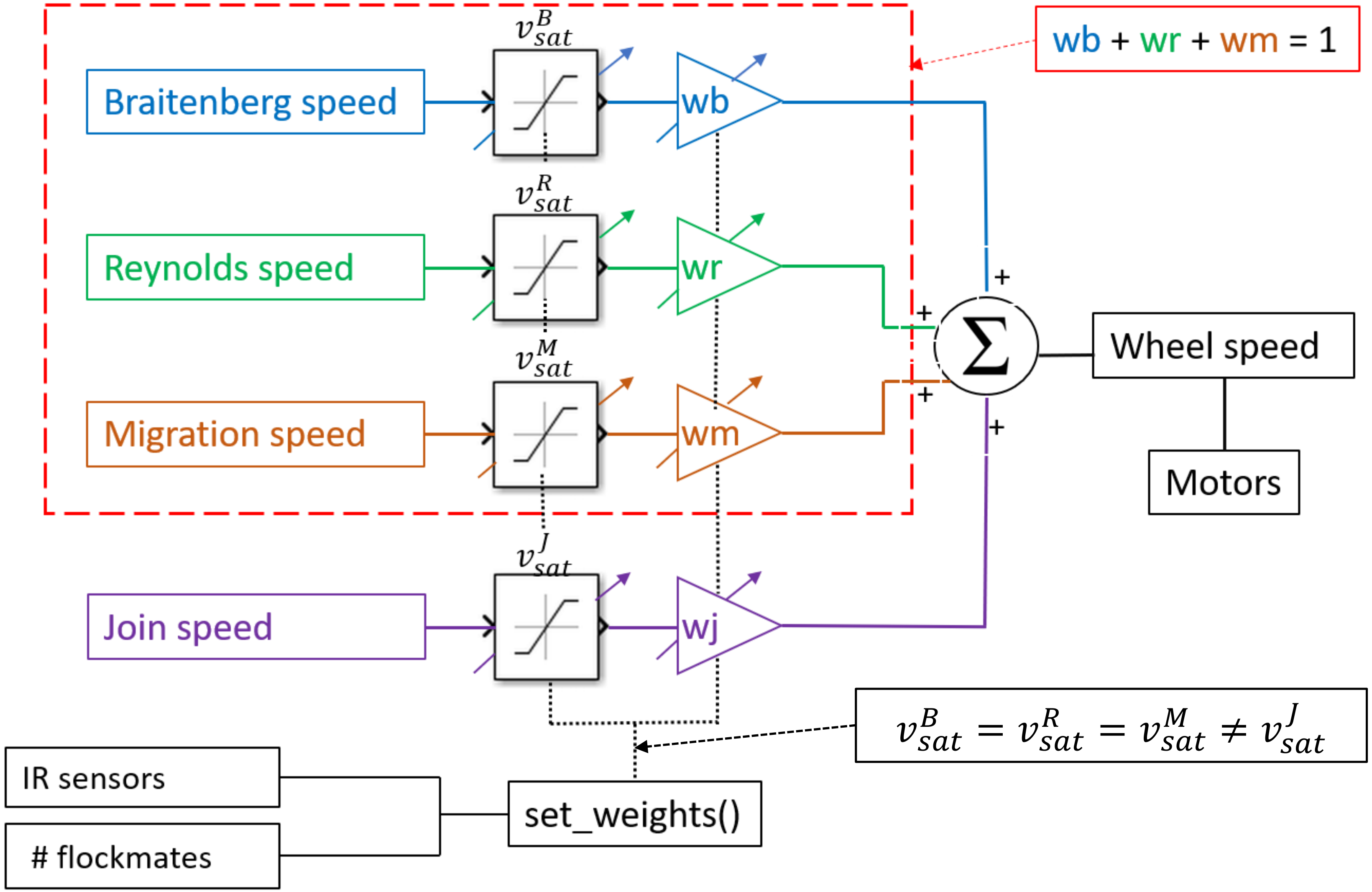
Real E-pucks
- Scenario A
- Scenario B

Scalability

Conclusion



Controller architecture



Architecture:

Main speed controller

- Braitenberg
- Reynolds
- Migration

Additional speed

- Join

Weights adapter

- Set_weights()

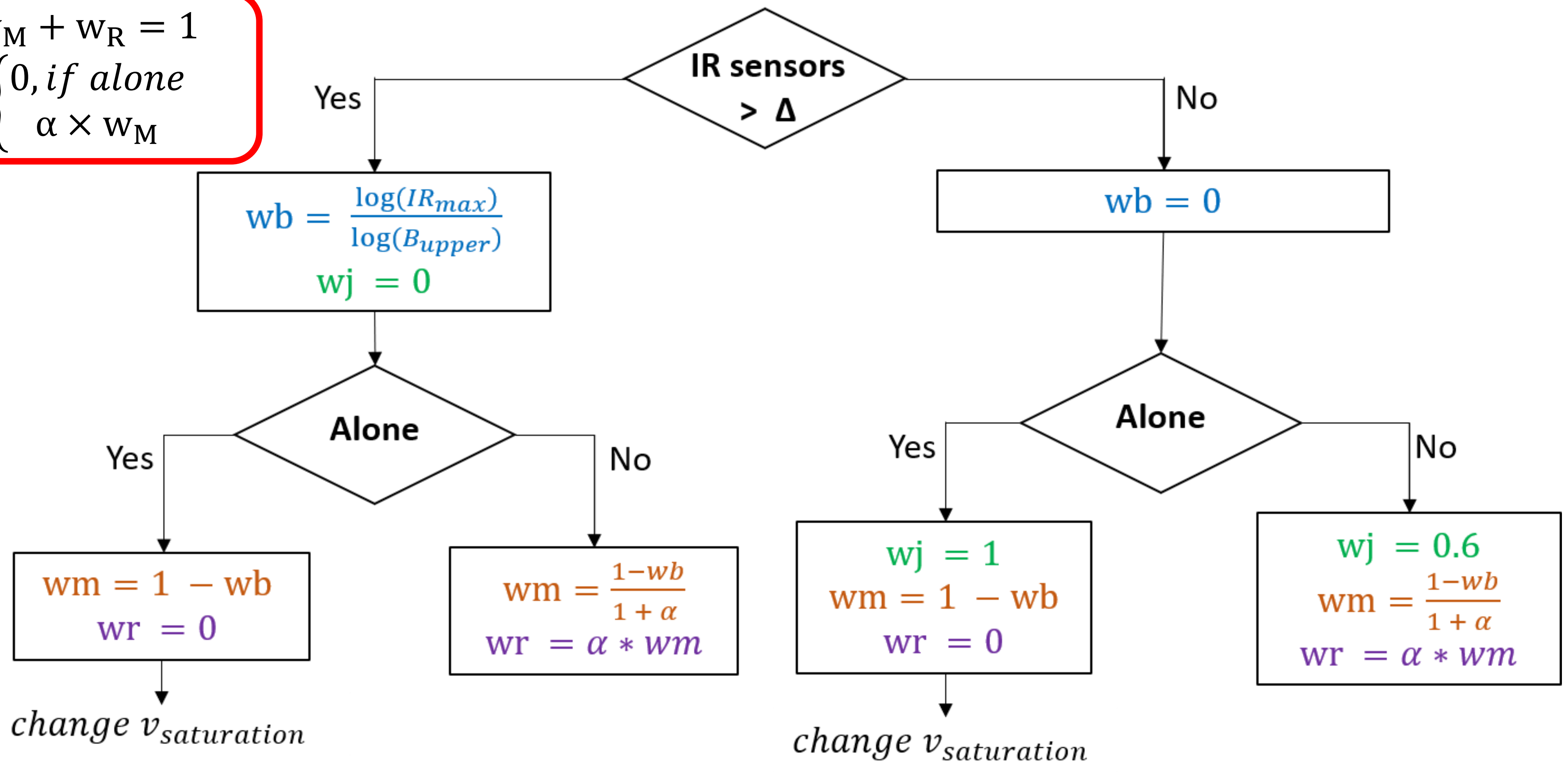
Set_weights()

- State machine based
- Change the gains
- Change the saturation speed

Weights adapter: set_weights

$$w_B + w_M + w_R = 1$$

$$w_R = \begin{cases} 0, if\ alone \\ \alpha \times w_M \end{cases}$$



Control Law:

$$v_{wheel} = w_b \times v_{Brait} + w_r \times v_{Reyn} + w_m \times v_{Migr} + w_j \times v_{Join}$$

Algorithms

Simulation

Algorithm 1 Simulation controller

```
while FOREVER do
    Reinitialize weights and speeds
    Listen and Send messages
    Compute Reynold Speed
    Compute Braitenberg Speed
    Compute Migration Speed
    Compute Join Speed
    Compute weights
    Compute final speed
    Update Odometry
end while
```

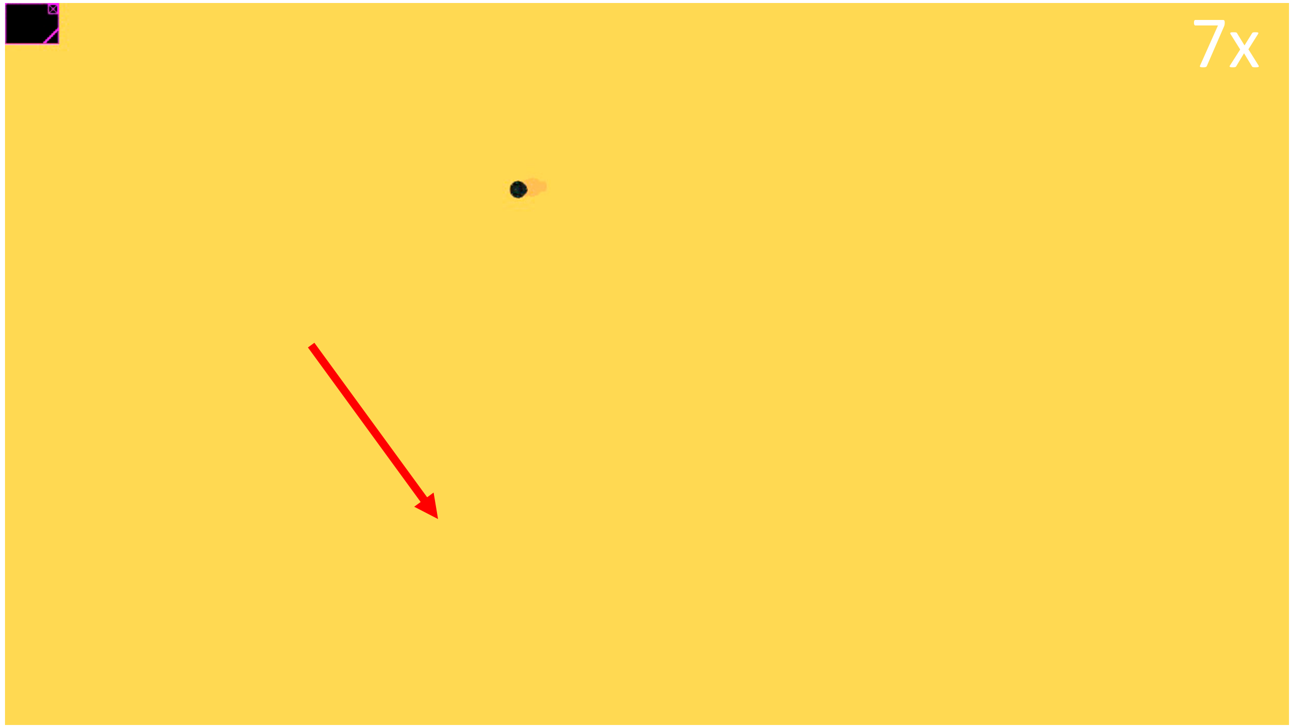
Real implementation

Algorithm 2 Real controller

```
while FOREVER do
    Reinitialize weights and speeds
    for 40 ms do
        Emit messages
    end for
    for 80 ms do
        Listen messages
    end for
    Compute Reynold Speed
    Compute Braitenberg Speed
    Compute Migration Speed
    Compute Join Speed
    Compute weights
    Compute final speed
    Update Odometry  (Real time step)
end while
```

Migration

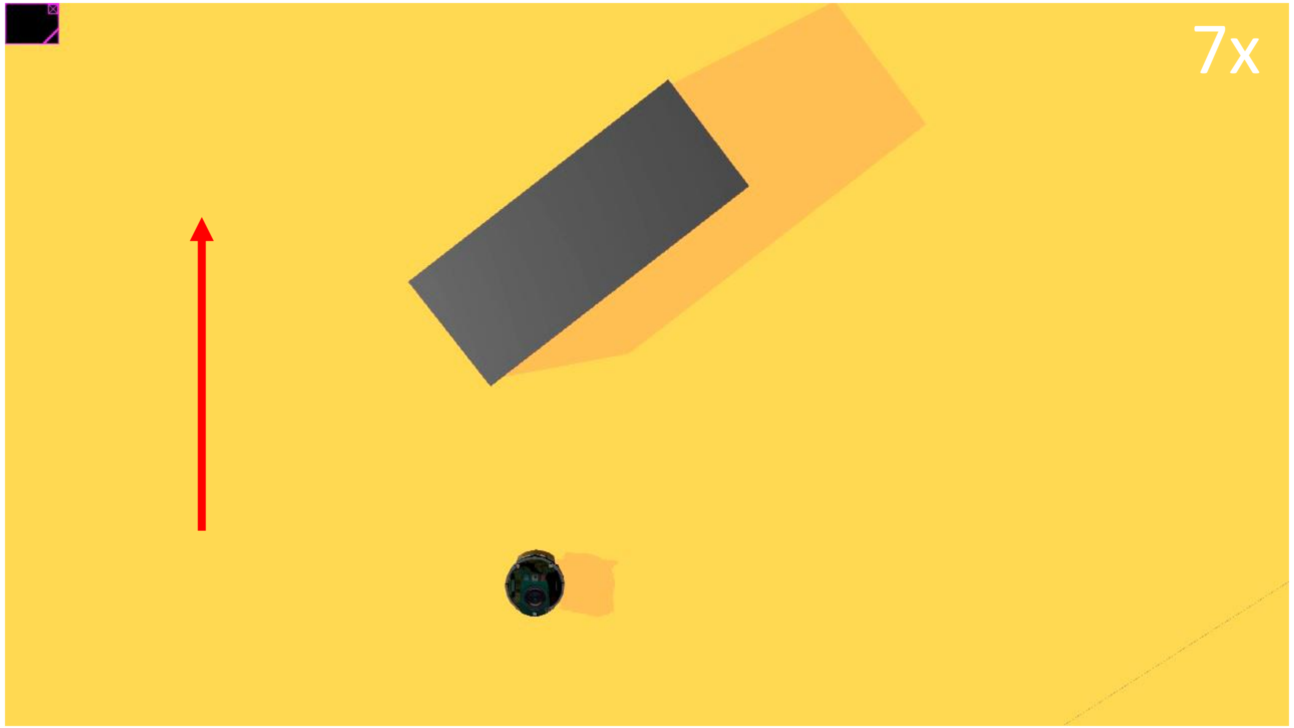
Simulation



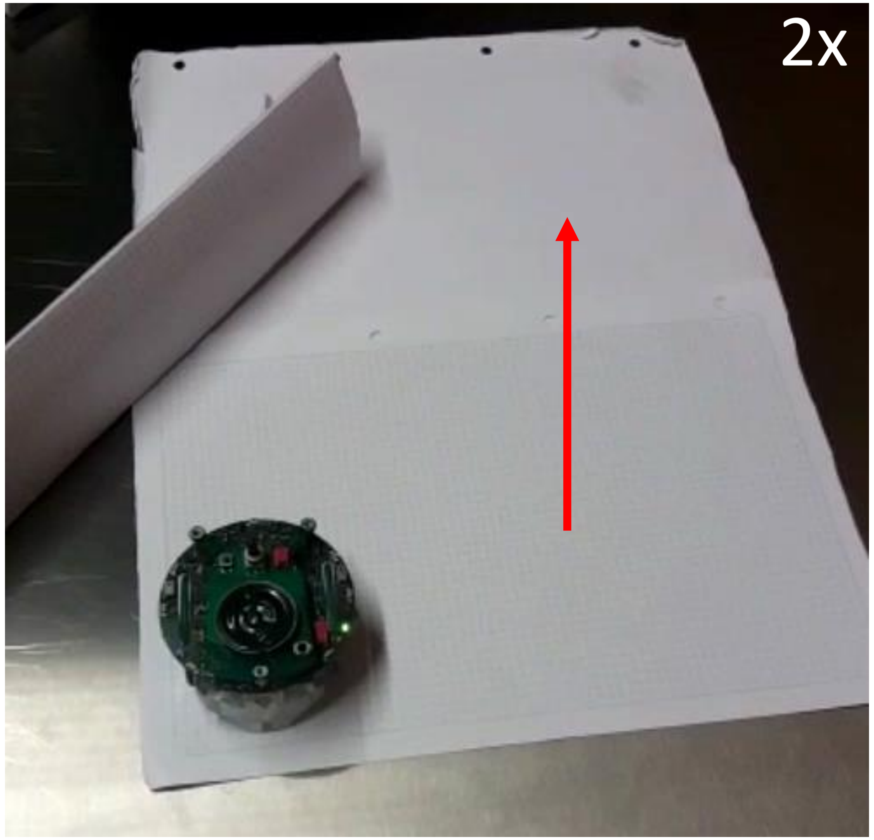
Real implementation

Migration + Braitenberg

Simulation



Real implementation



Weights

Motor	IR0	IR1	IR2	IR3	IR4	IR5	IR6	IR7
Right	16	9	6	3	-1	-5	-6	-8
Left	-12	-8	-5	6	2	4	5	9

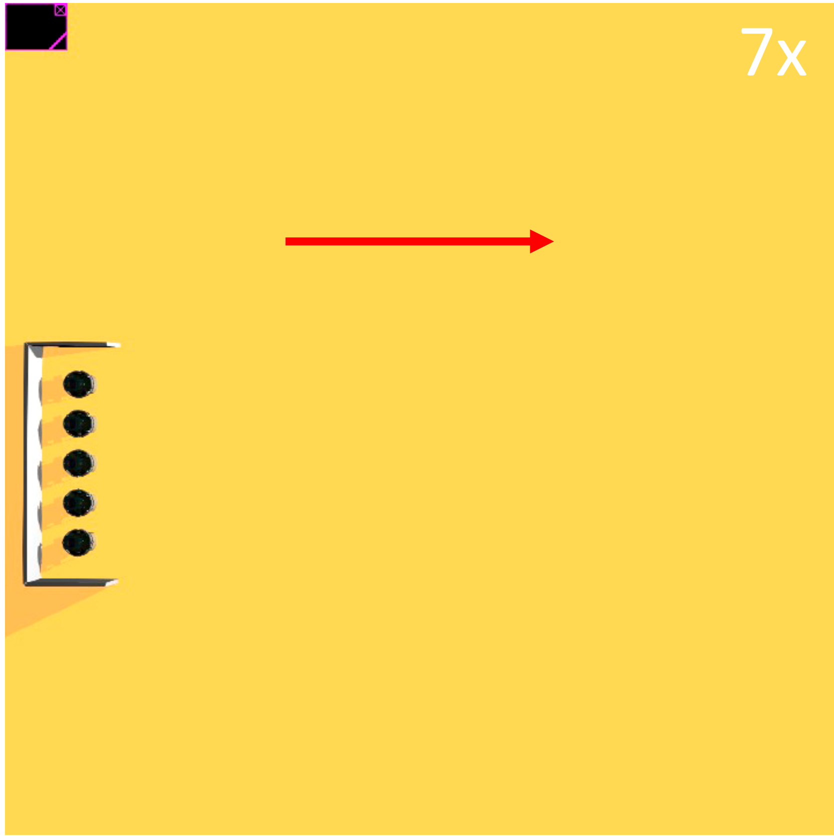
Reynolds: Real implementation

Cohesion

Dispersion

Reynolds + Migratory

Simulation



Recall: $w_R = \alpha \times w_M$

- $\alpha < 1$: more Migration
- $\alpha > 1$: more Reynolds

Solution: **Heterogeneity**

$$\alpha = \alpha_0 + (Robot_{id} + 1) \times \frac{2}{5}$$

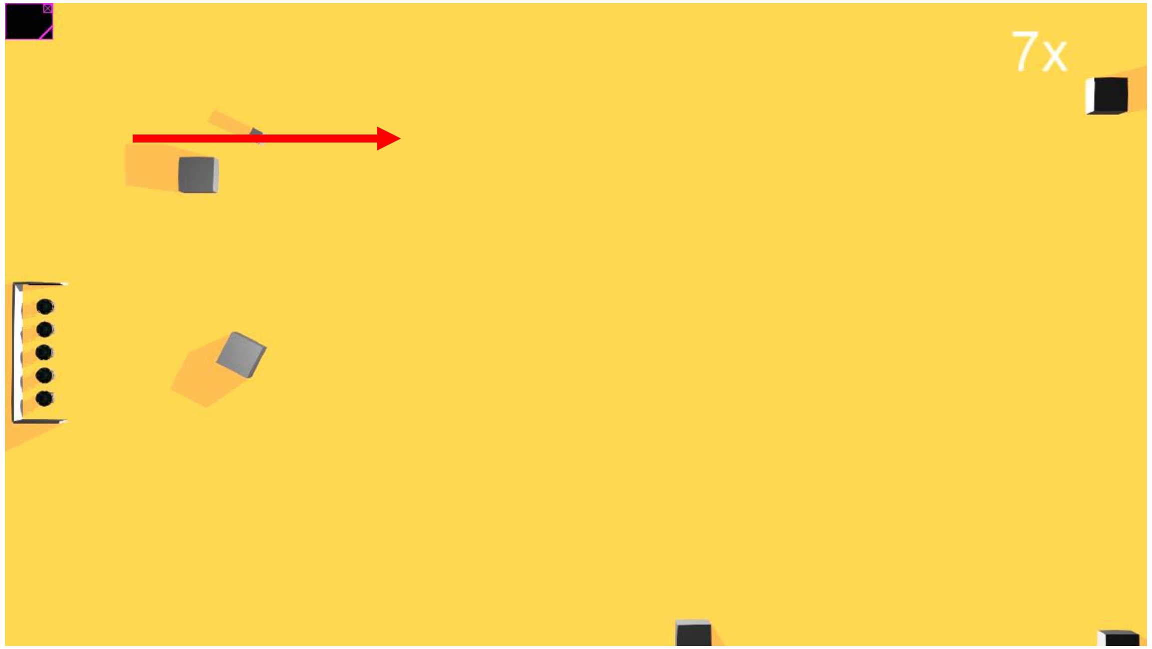
Parameters:

Reynolds' rules	Threshold	Weights
Cohesion	10 cm	7.0
Dispersion	12 cm	7.0
Alignement	-	4.0

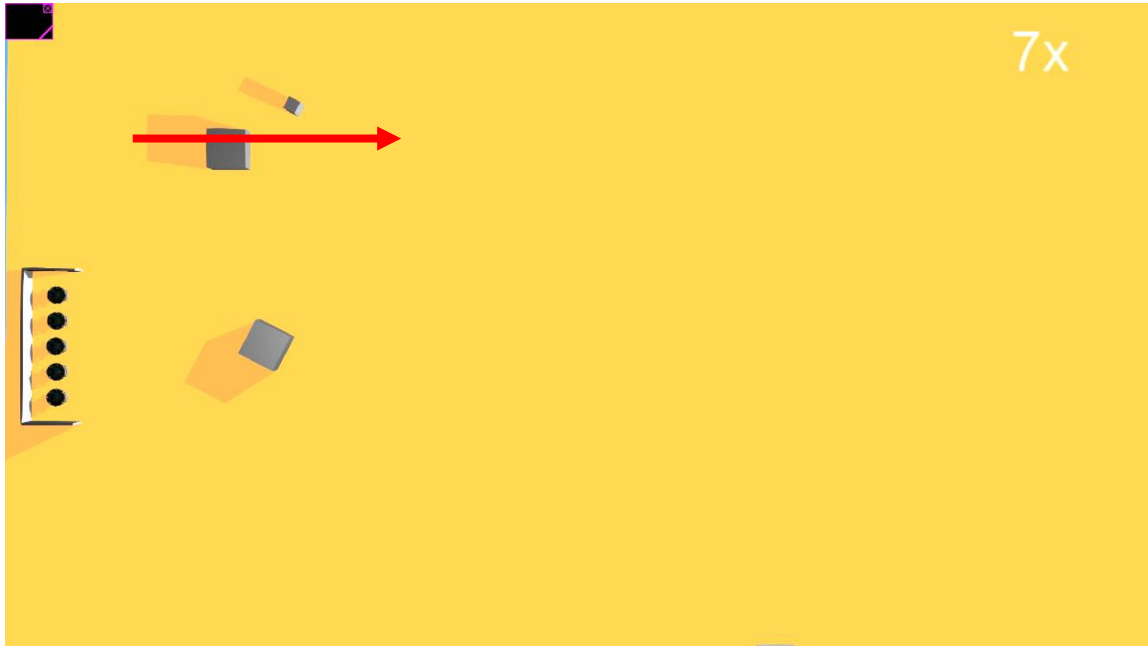
$\boxed{\alpha_0 = 1.5}$

Complete controller

Without Join

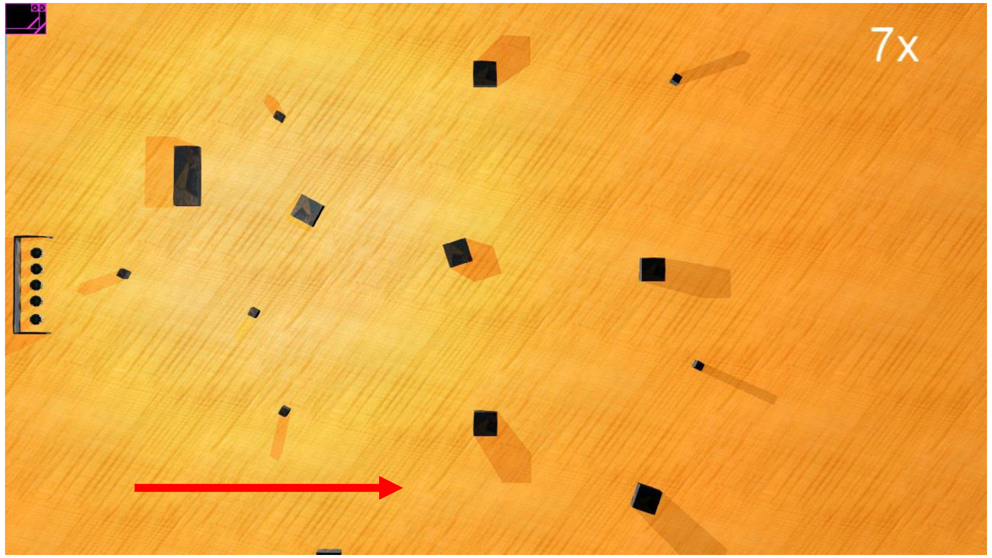


With Join

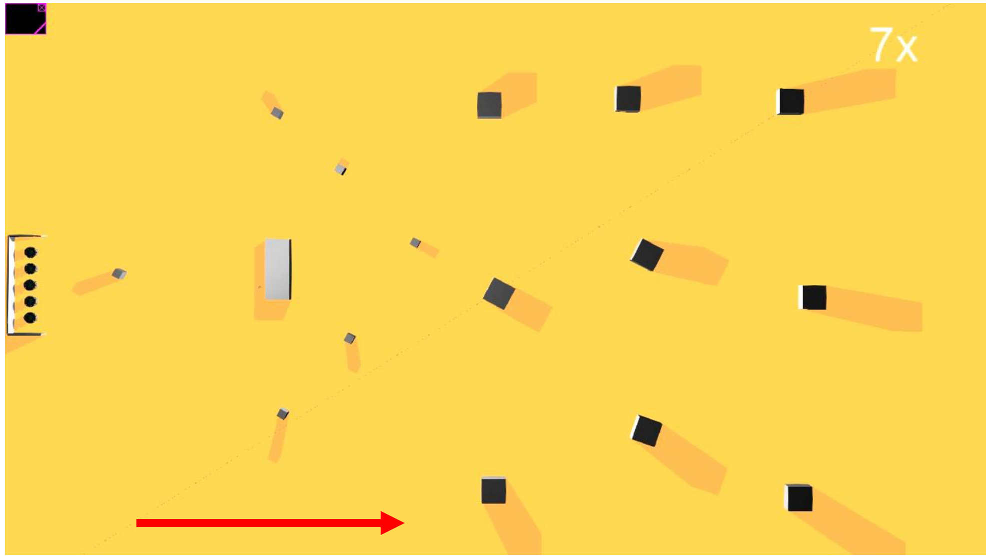


Simulation: Scenario A

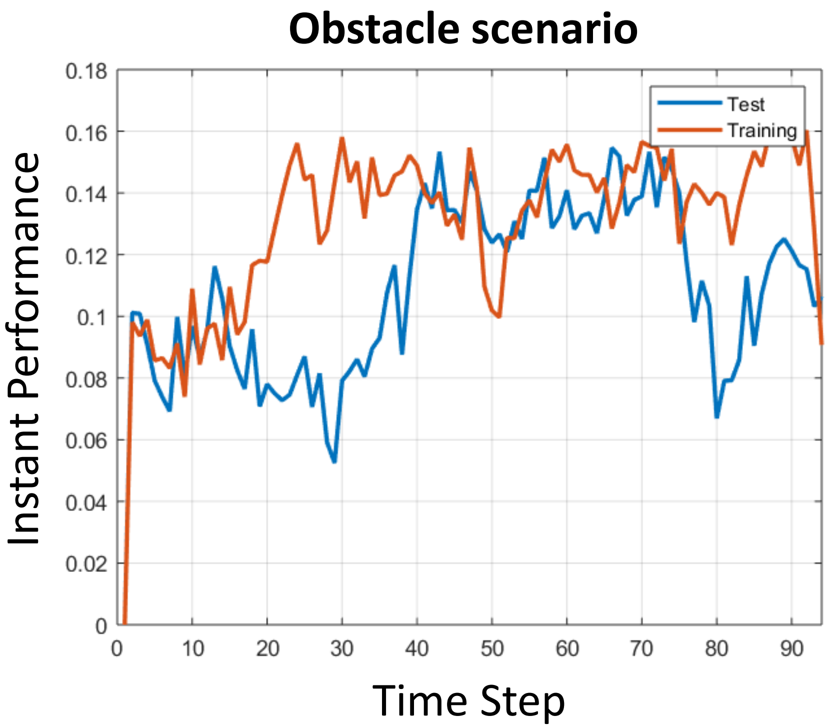
Training



Test

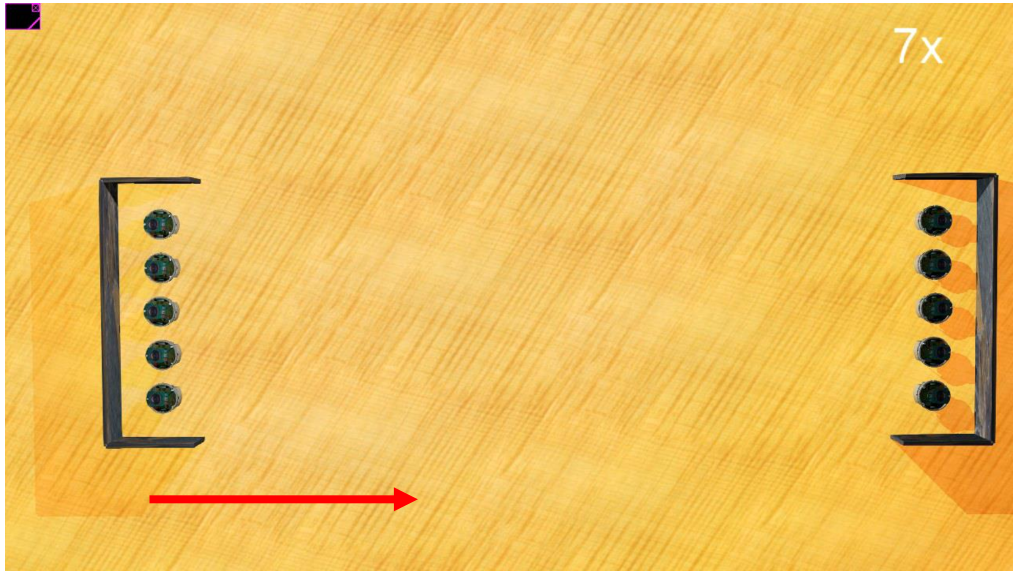


Fitness	Training [%]	Test [%]
Orientation	99.94	99.87
Velocity	17.53	17.47
Cohesion	91.44	91.16
Overall	0.038	0.039



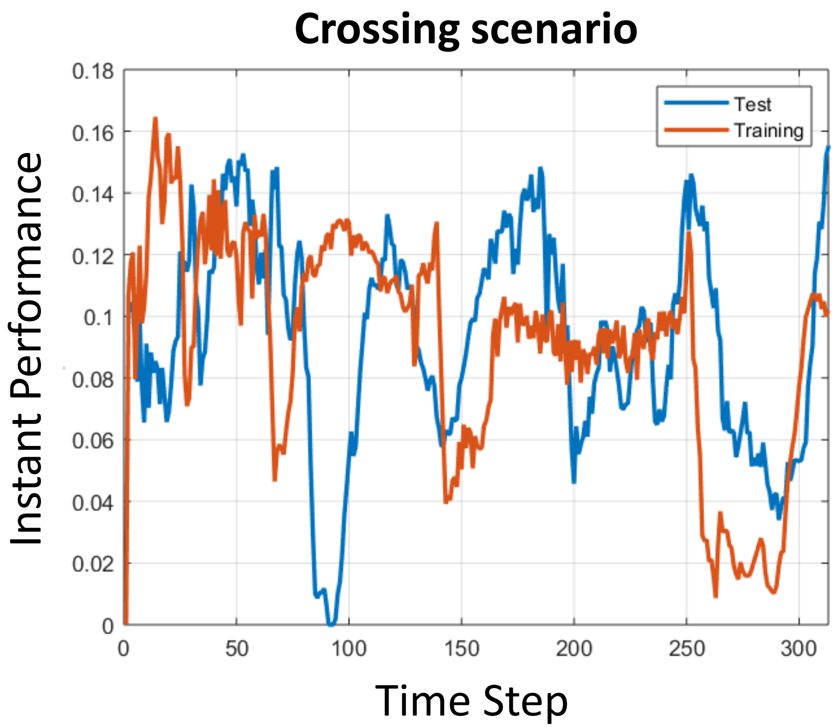
Simulation: Scenario B

Training

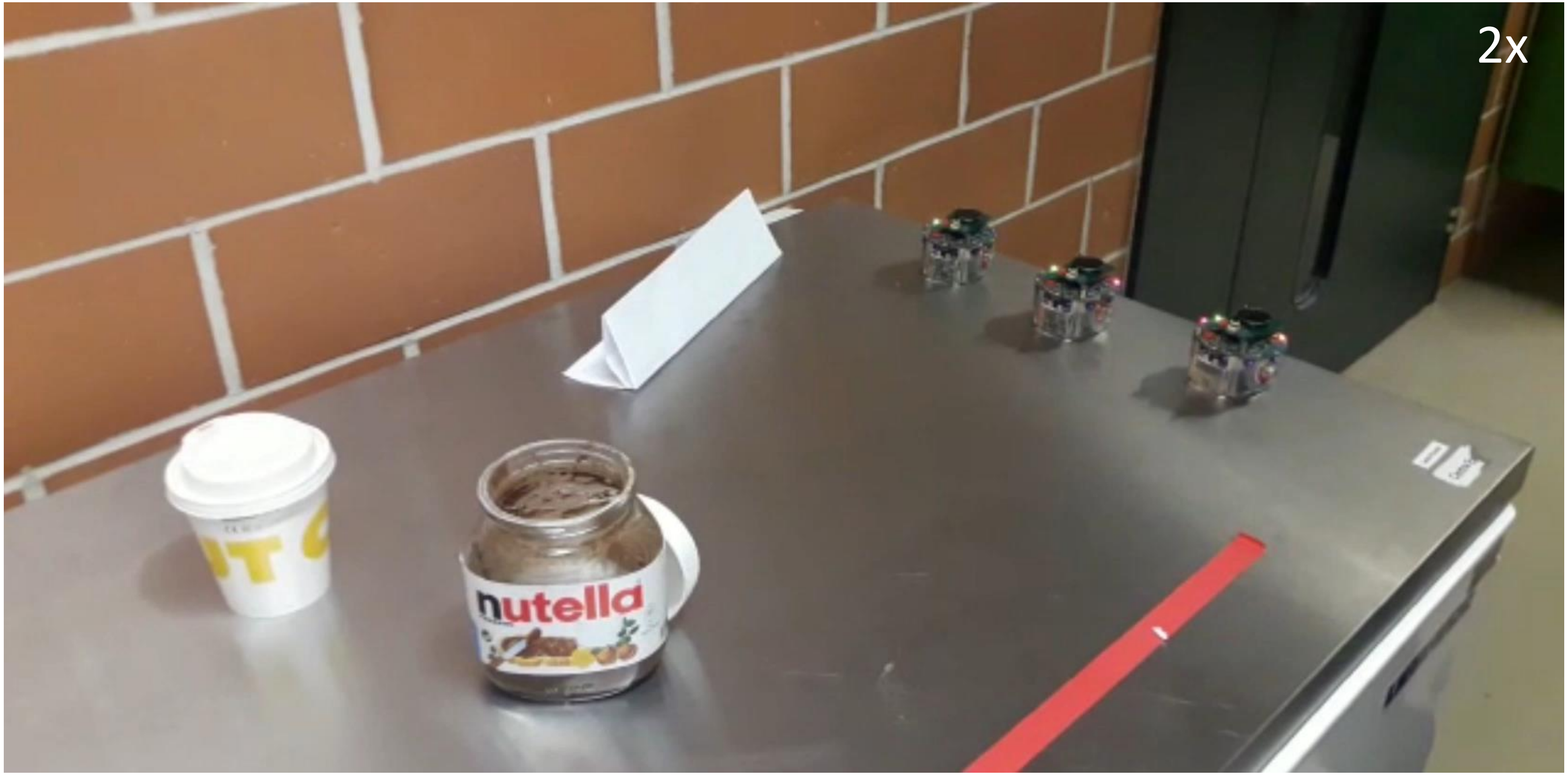


Test

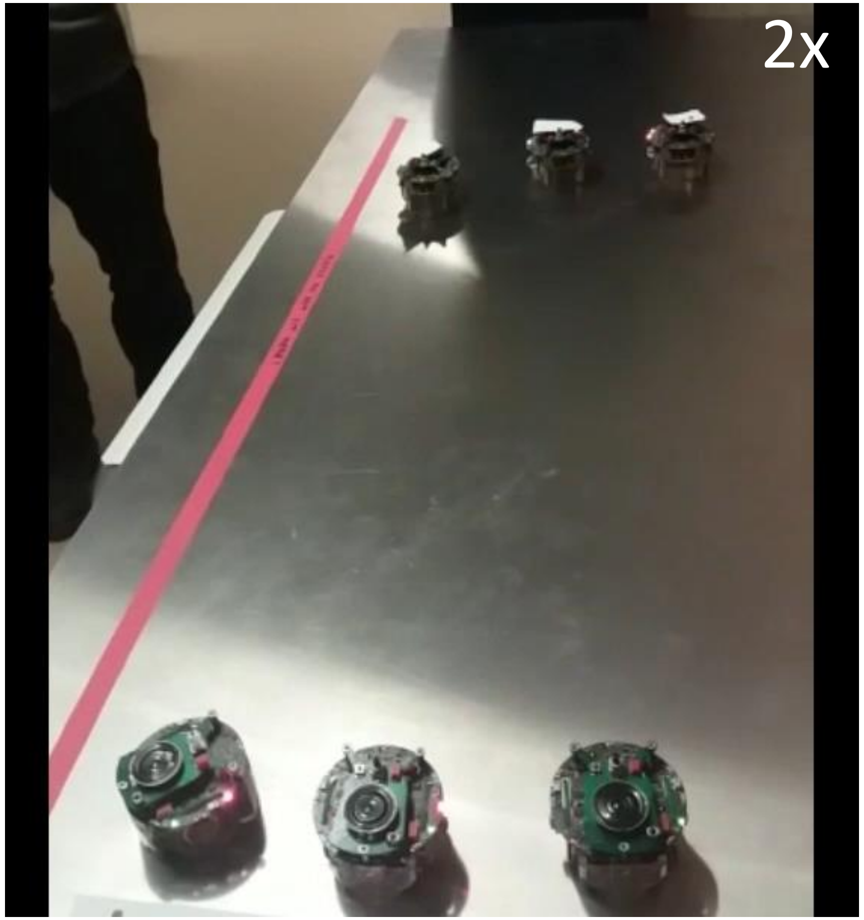
Fitness	Training [%]	Test [%]
Orientation	99.65	98.48
Velocity	11.3	17.61
Cohesion	91.15	91.36
Overall	0.074	0.078



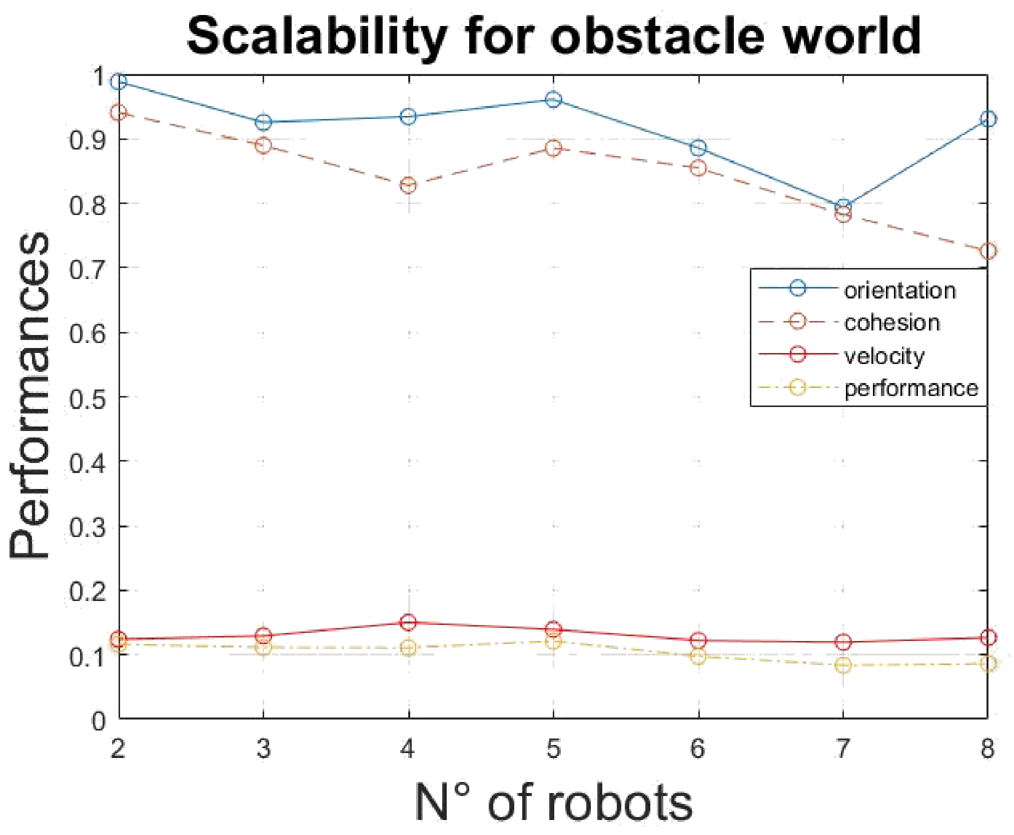
Real e-puck: Scenario A

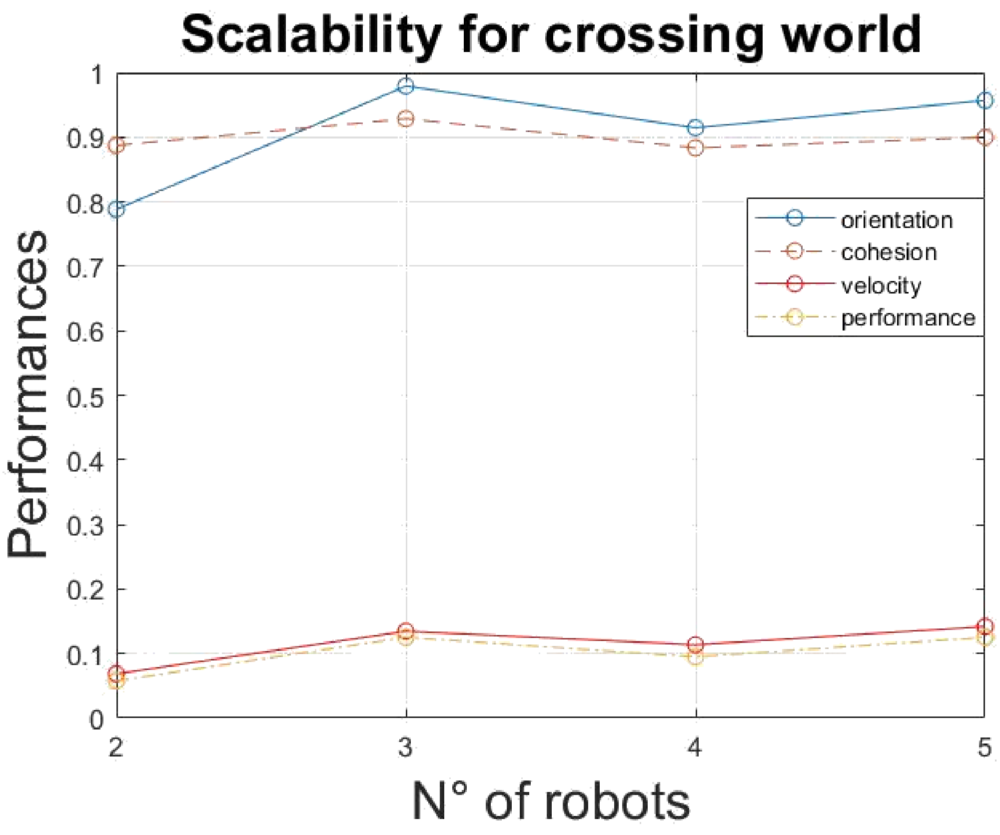


Real e-puck: Scenario B



Scalability





Conclusion

Results discussion

- Obstacle better than crossing
- Robust to change of flock size

Possible improvement

- Optimization algorithm (ex: PSO, etc)
- Optimization Hardware in the loop
- Communication strategy (synchronization)

Metrics corrections

Obstacle scenario

Fitness	Training [%]	Test [%]
Orientation	99.94	99.87
Velocity	54.78	54.59
Cohesion	91.44	91.16
Overall	0.1188	0.1219

Crossing scenario

Fitness	Training [%]	Test [%]
Orientation	99.65	98.48
Velocity	35.31	52.13
Cohesion	91.15	91.36
Overall	0.2188	0.2438