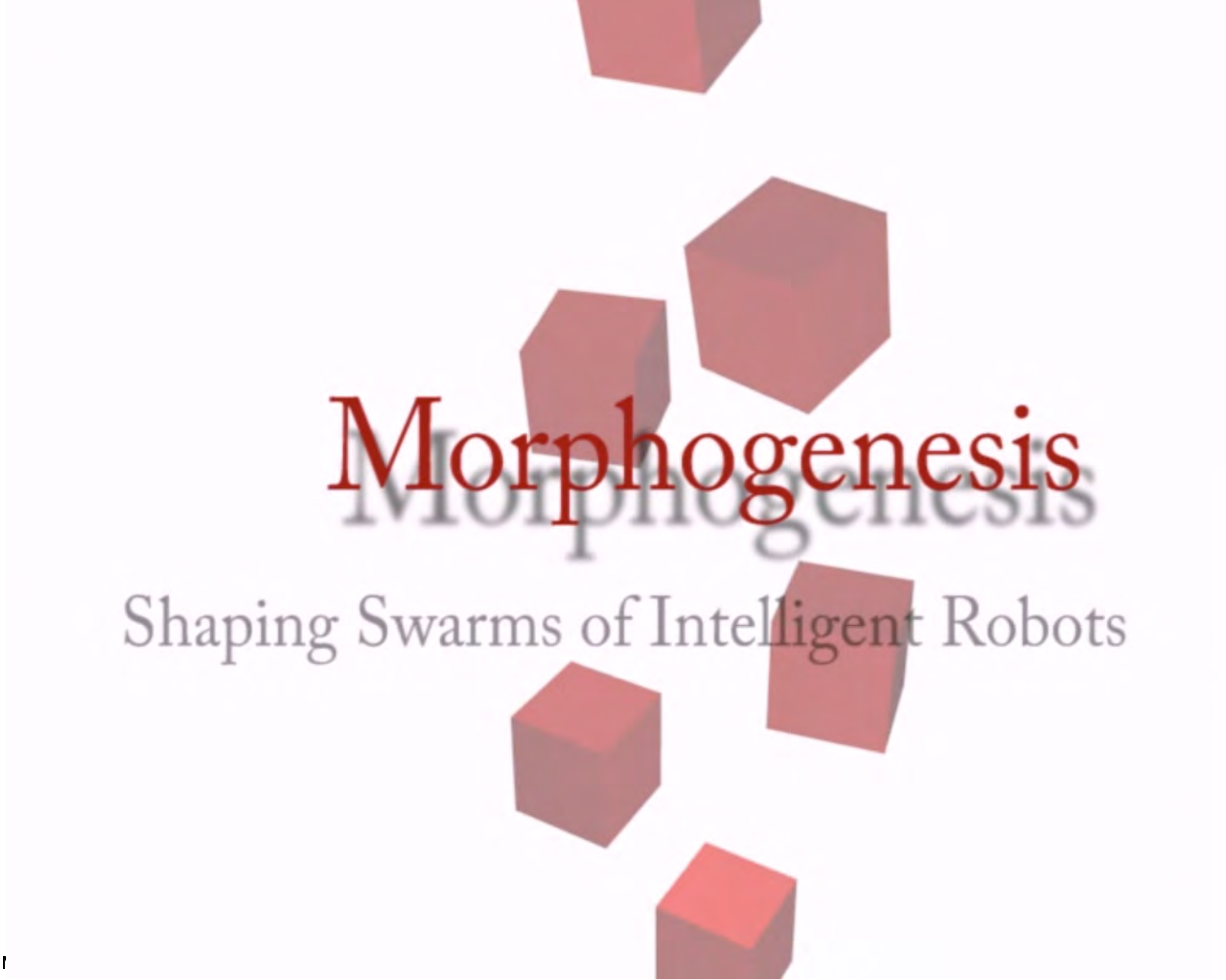
Swarm Robotics

Prof. Marco Dorigo


Morphogenesis
Shaping Swarms of Intelligent Robots

WHAT IS SWARM ROBOTICS?

- Swarm robotics studies how a large number of embodied agents can be designed and controlled so that a desired collective behavior results from local interactions among the agents and between the agents and the environment in which they act
- The main goal pursued is the development of physical swarms exhibiting collective behaviors that are fault tolerant, scalable and flexible

WHAT IS SWARM ROBOTICS?

Swarm robotics is embodied swarm intelligence

- it studies how to design, build and control swarms of cooperating robots
- and how to use swarm intelligence principles – decentralised control, local sensing and communication, self-organisation – to control them

THE SWARM ROBOTICS PROBLEM

How to control a large number of robots (robot swarm) so that:

- the robots cooperate to perform given tasks
- the robot swarm is tolerant to Byzantine robots
  (i.e., malfunctioning or malicious robots)
- the robot swarm is scalable
  (i.e., amount of work produced can be adapted by increasing/decreasing the number of robots without need for reprogramming)

HOW TO CONTROL A ROBOT SWARM ?

Centralised control: one central controller has full knowledge about all system components and is in charge of controlling them

- in favour: a lot of experience on how to program centralised systems
- against: single point of failure, difficult to scale

Self-organization: no centralised control (control is distributed over the system components), local interactions, local communication

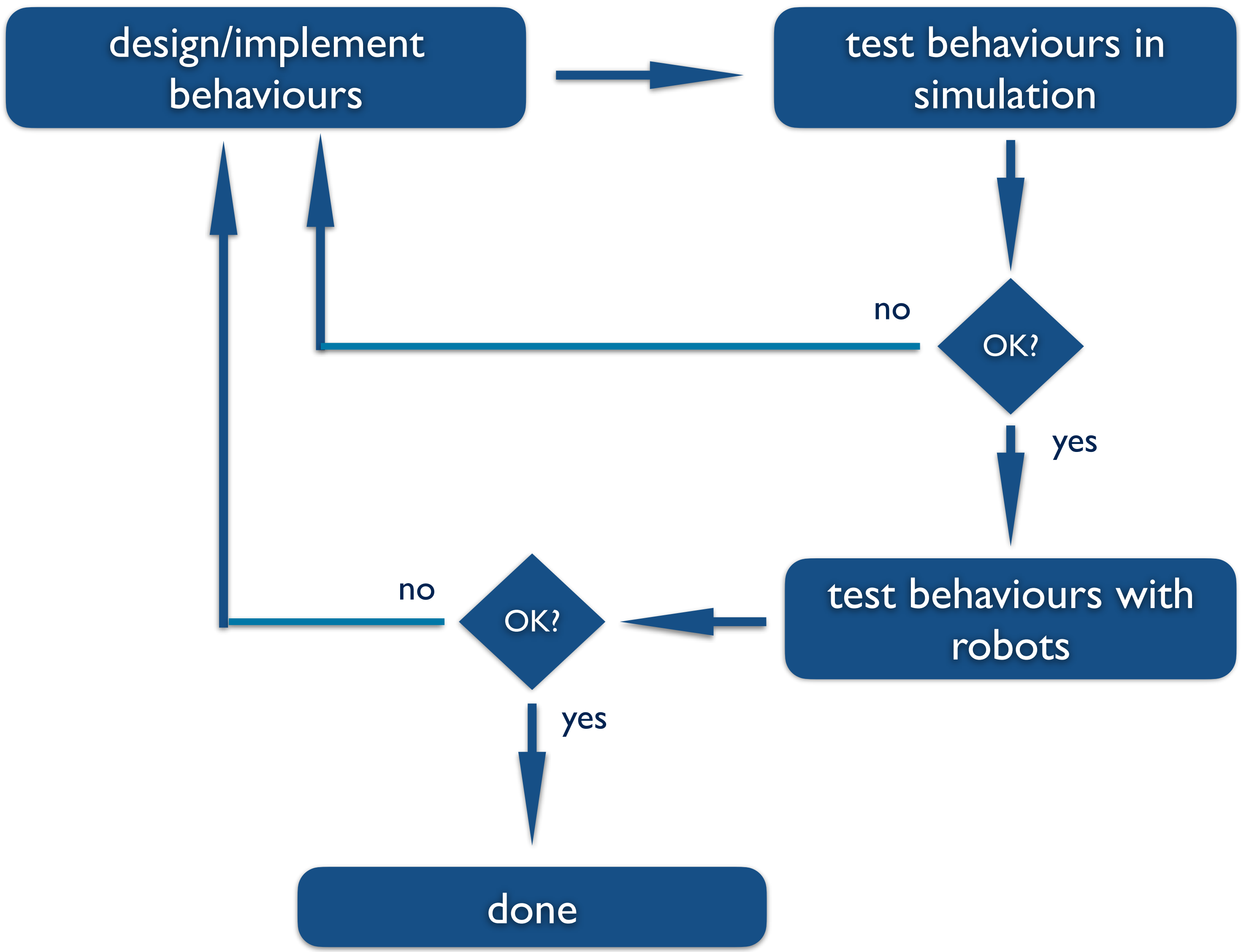
- in favour: coherent with our goals (fault tolerance, scalability)
- against: micro-macro problem

THE MICRO-MACRO LINK PROBLEM

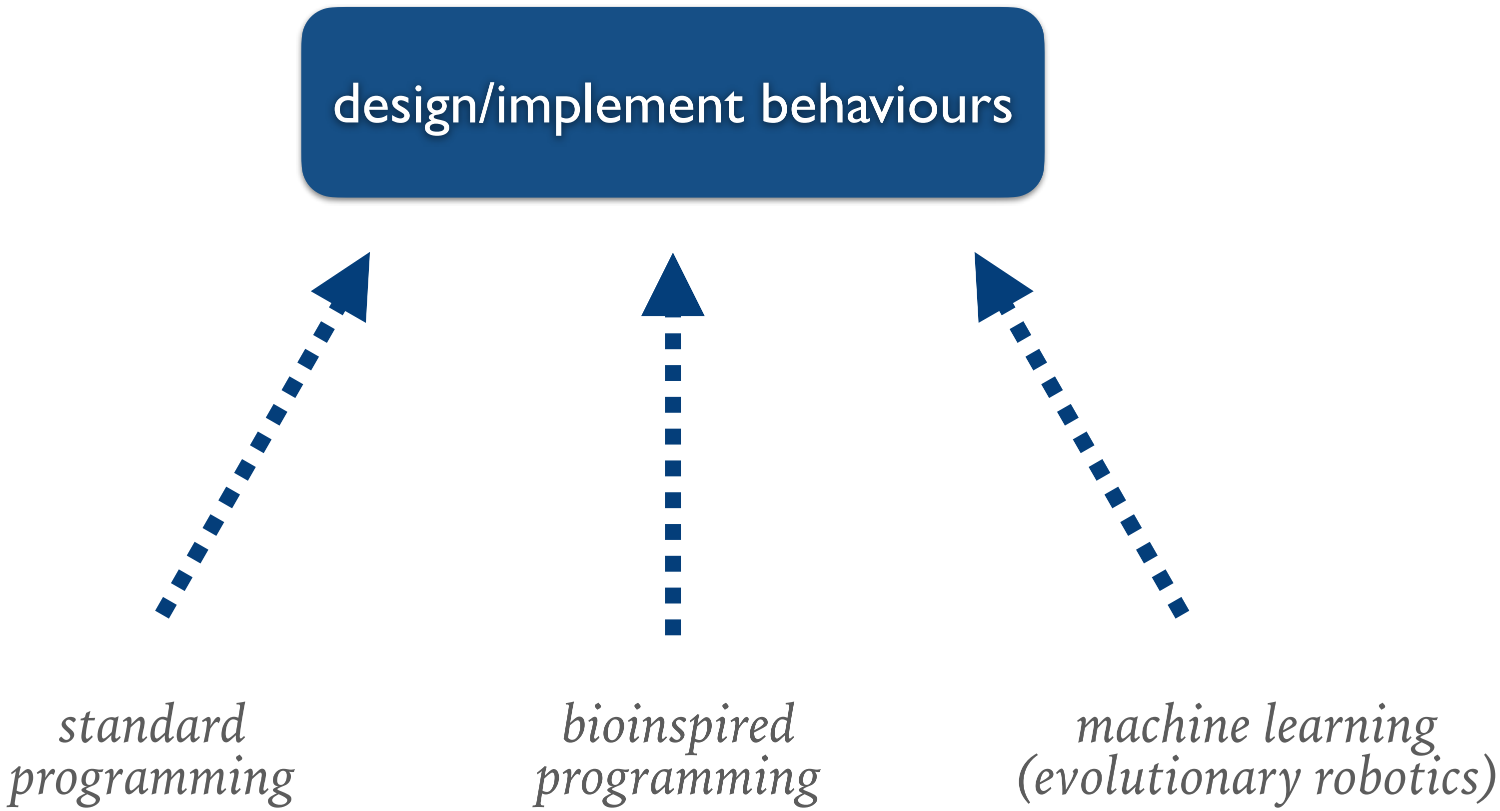
How to program the single robots so that the swarm behaves as you want

The goal is to program the swarm as a whole, but single robots are programmed

HOW DO WE PROGRAM A SELF-ORGANISED ROBOT SWARM?



HOW DO WE PROGRAM A ROBOT SWARM?

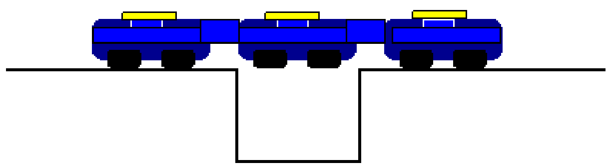


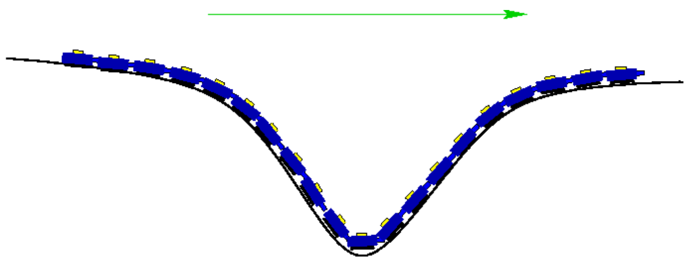
What is a swarm-bot?

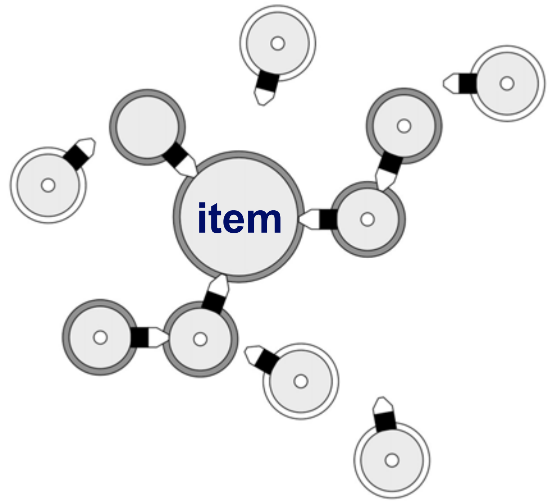
- A “**swarm-bot**” is an artifact composed of a number of simpler robots, called “**s-bots**”, capable of **self-assembling** and **self-organizing** to adapt to its environment
- S-bots can **connect to and disconnect** from each other to self-assemble and form structures when needed, and disband at will

What should a swarm-bot be able to do?

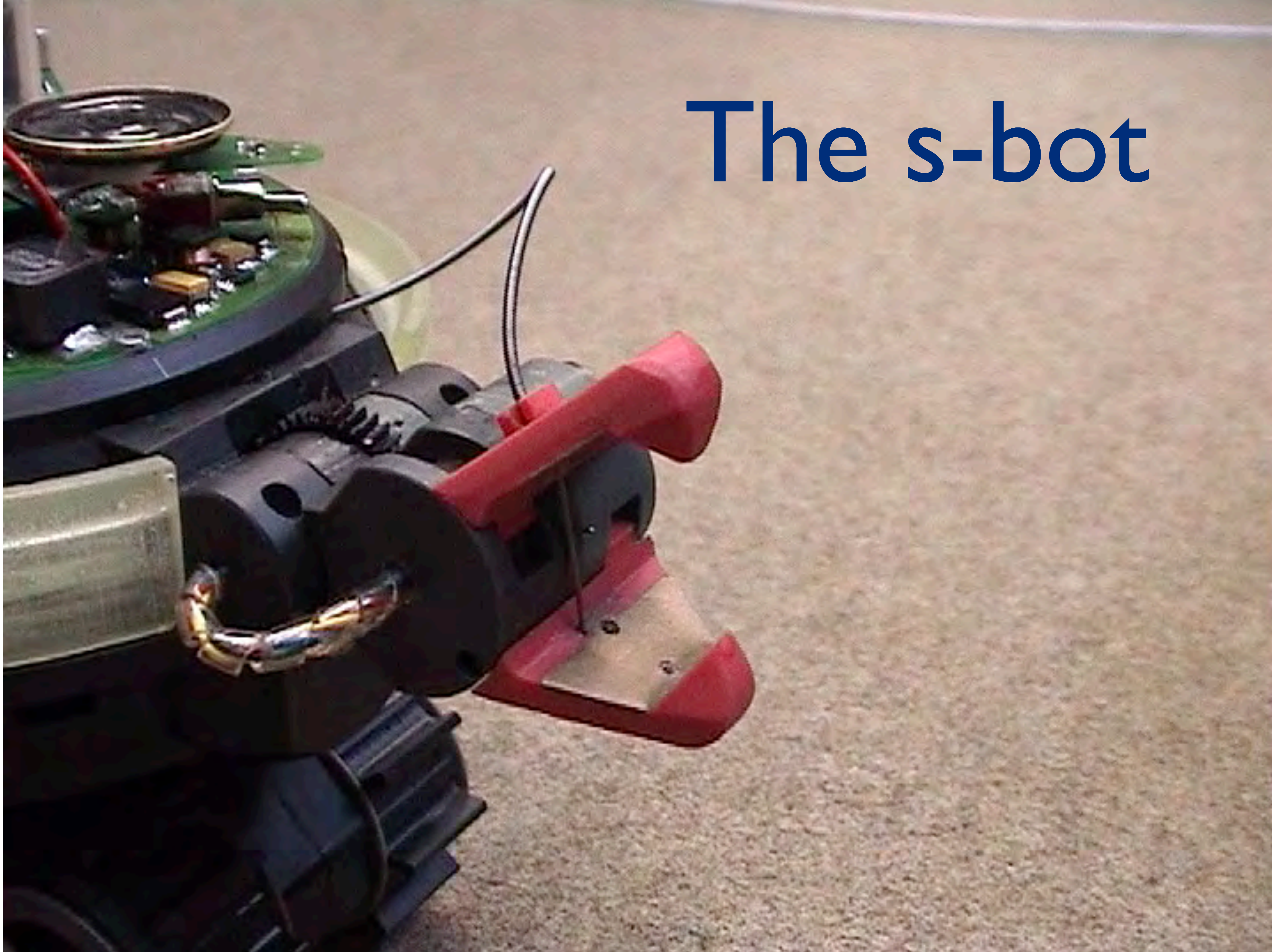
- Move in formation to overcome obstacles that a single s-bot cannot overcome alone
- Retrieve an item that is too heavy for a single s-bot

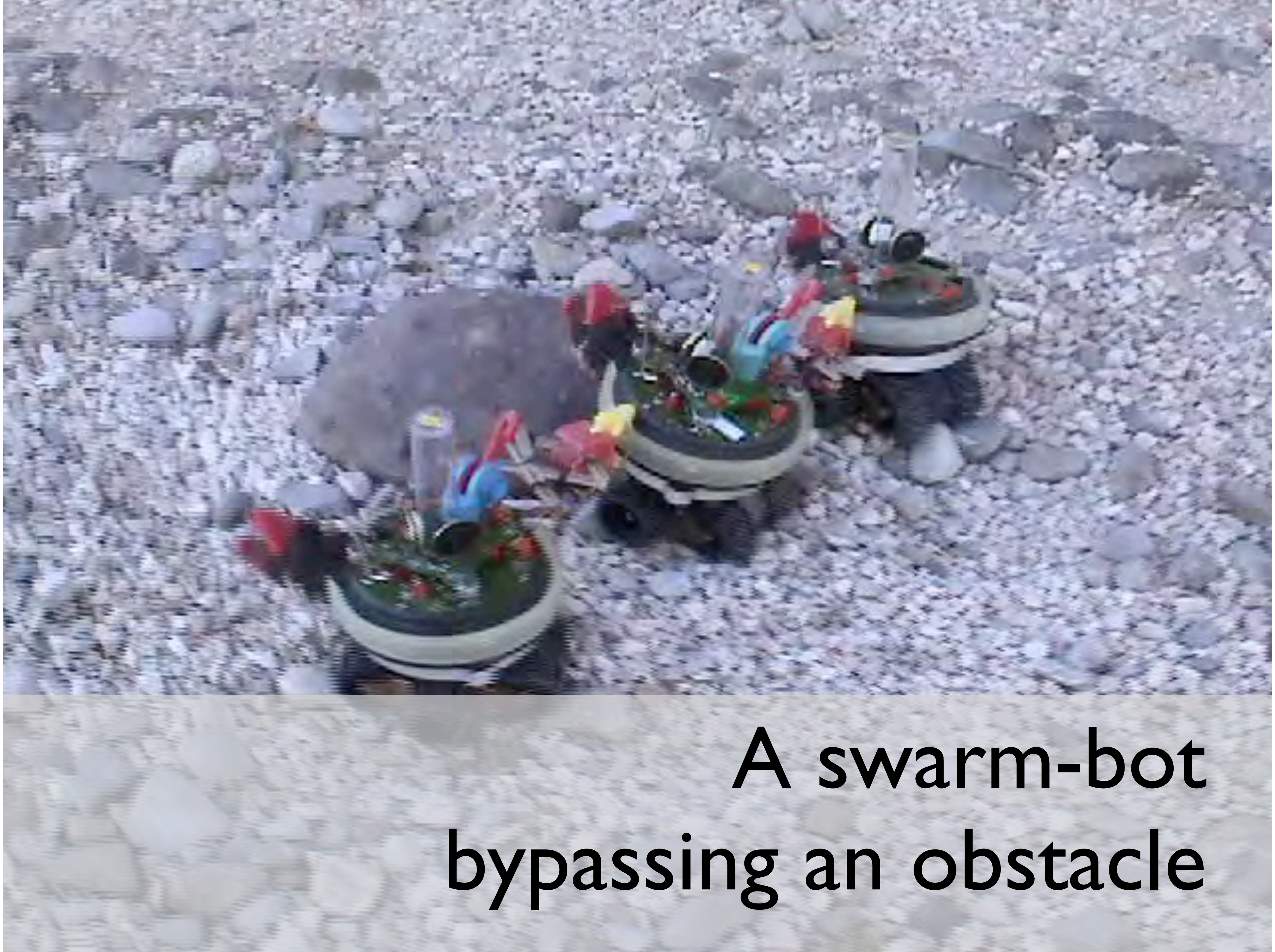






The s-bot





A swarm-bot bypassing an obstacle

Technological motivations

- Parallelism:
  Different robots can perform different task at the same time
- Fault tolerance:
  Simple agents are less prone to failure. When a robot breaks down another one can take over. No single point-of-failure
- Cost:
  Simple robots are cheaper to build than complex robots
- Scalability:
  Add more robots, get more work done

How to get there

- Redundant hardware:
  multi-agent system
- Decentralized control system:
  self-organized and distributed control
- Decentralized communication system:
  individuals use only local information

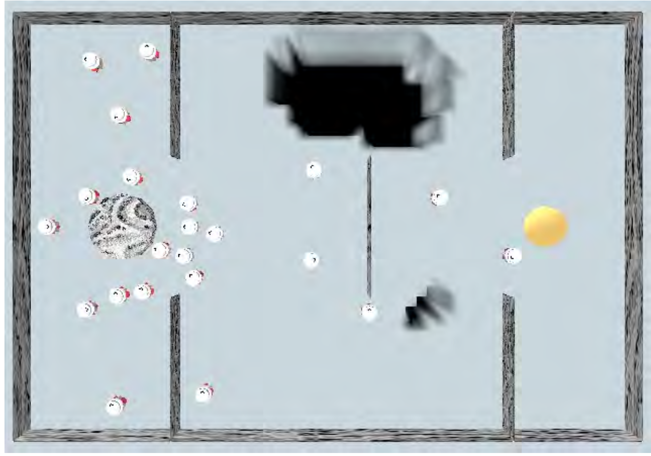
Biological inspirations

- **Inspiration 1:**
  Social insects can perform coordinated, colony-level tasks despite the lack of a global communication system
  We exploit three mechanisms that lie behind this coordination:
  stigmergy, ***self-assembly***, ***collective transport***
- **Inspiration 2:**
  Robot controllers implemented as ***neural networks*** whose weights are learned by ***evolutionary computation*** techniques
- **Inspiration 3:**
  Type of considered tasks

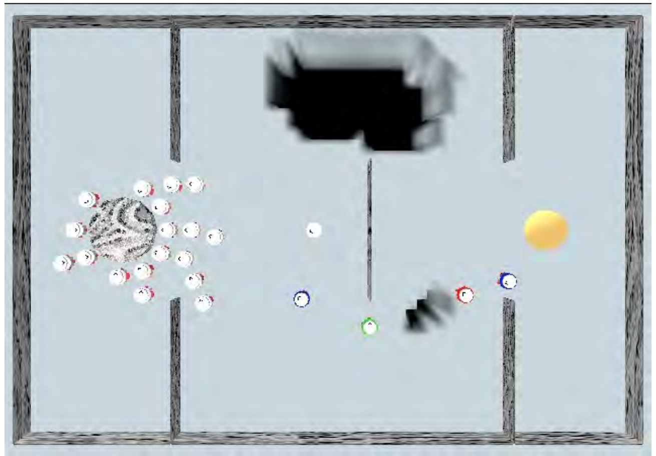
Swarm-bots

The scenario

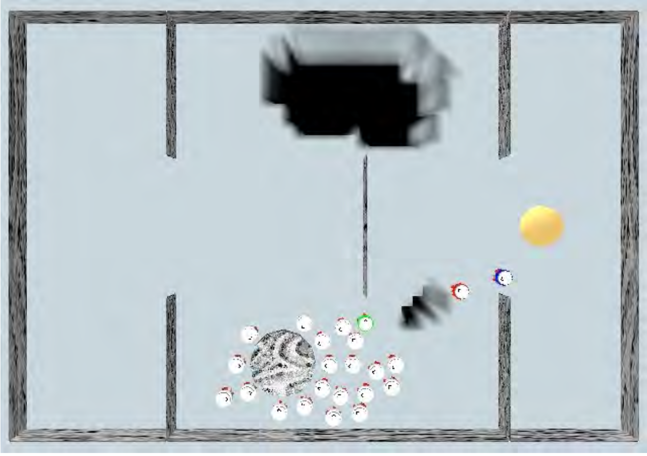
Find object and aggregate around it

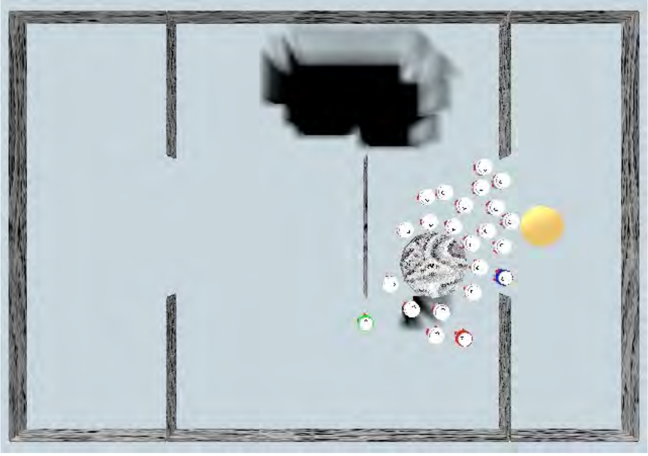


Pull object and search for goal



Change shape and move in a coordinate way avoiding obstacles





Swarm-bots

Our scenario

Find object and aggregate around it

Pull object and search for goal

Controllers development: methodology

- Develop a simulation model of the hardware
- Define the basic behaviors to be developed
- Use either

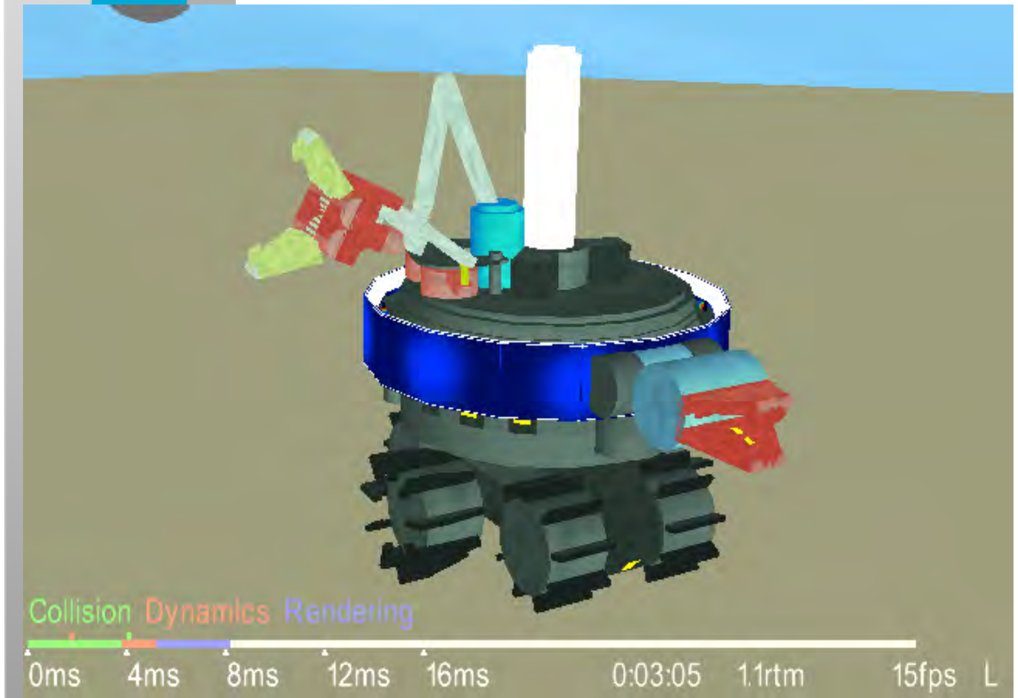
  hand-coded behavior-based architectures

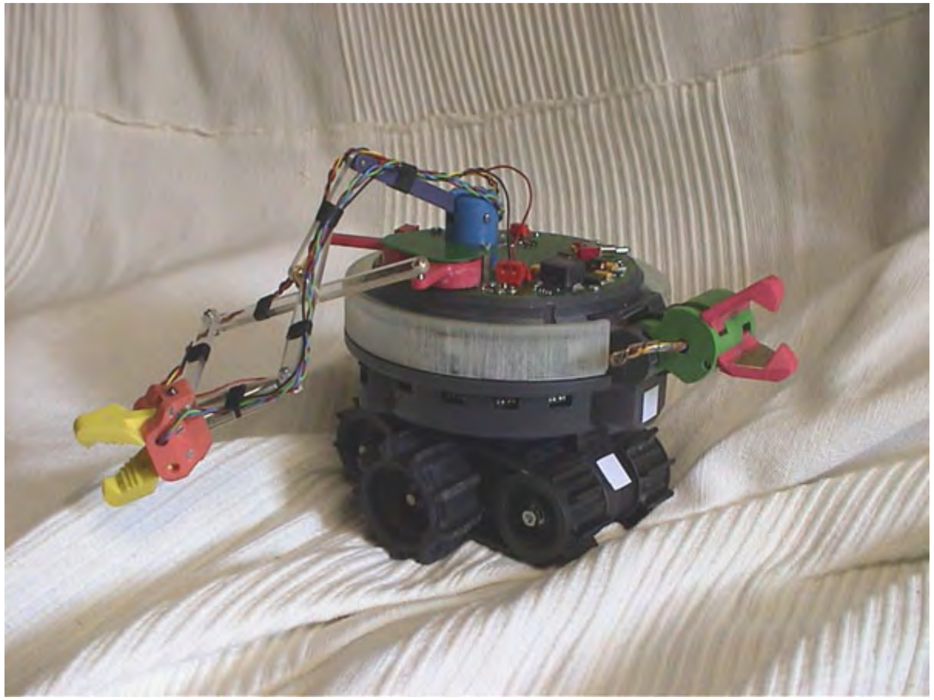
  or

  artificial evolution of neural networks

  to synthesize the basic behaviors in simulation that can be ported to the real *s-bots*
- Download and test the obtained controllers on the real *s-bots*

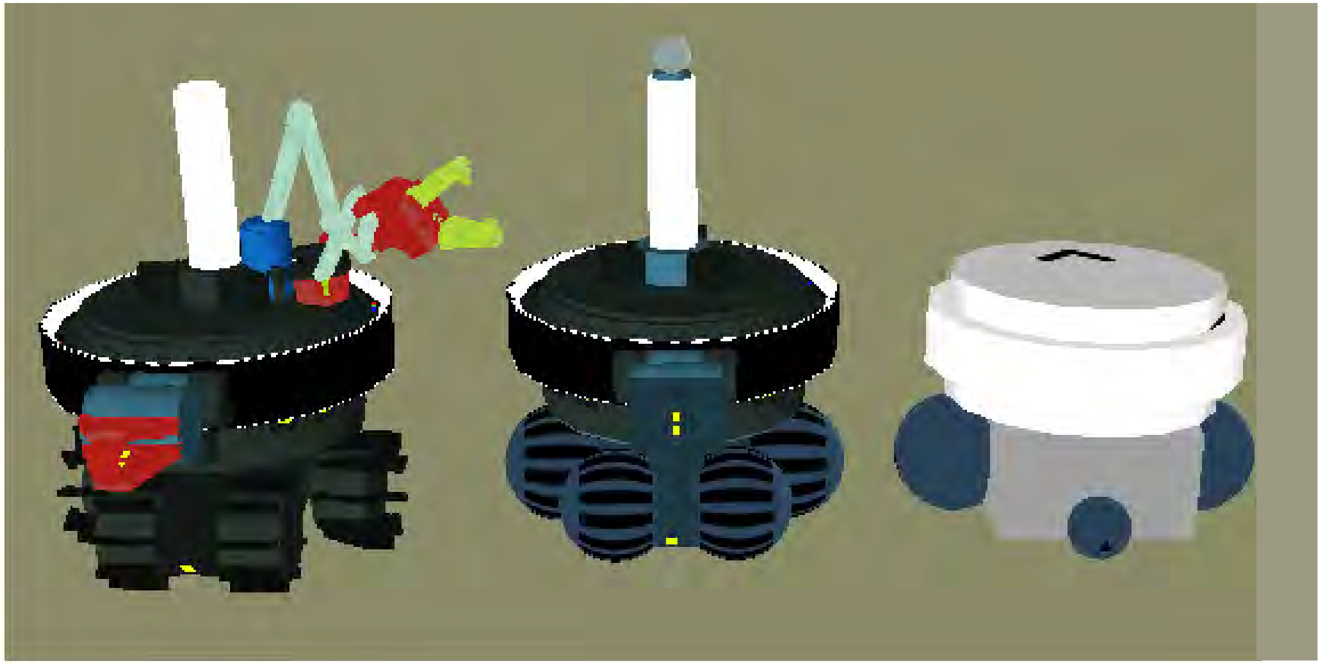
Simulation model





Swarm-bots

Different levels of detail

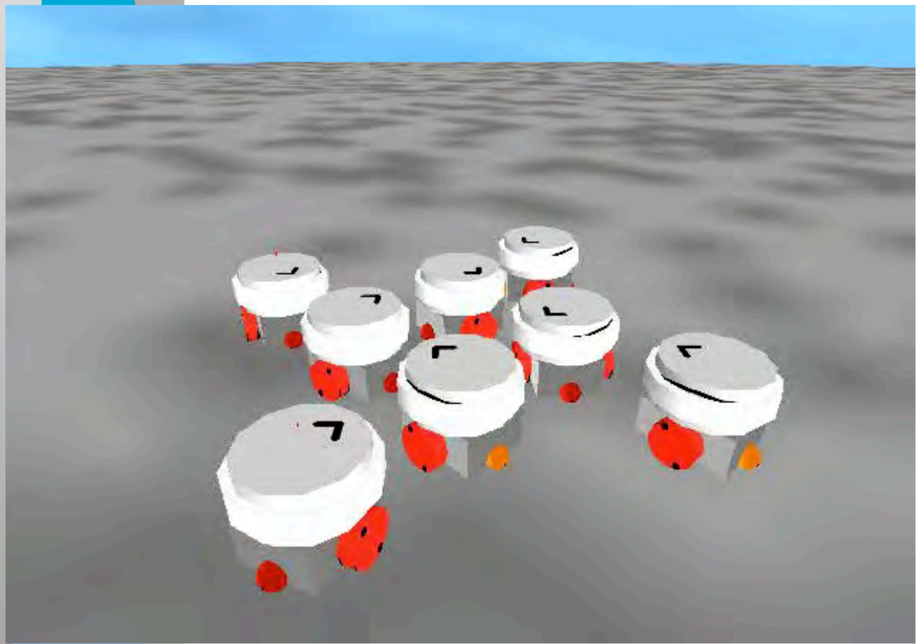


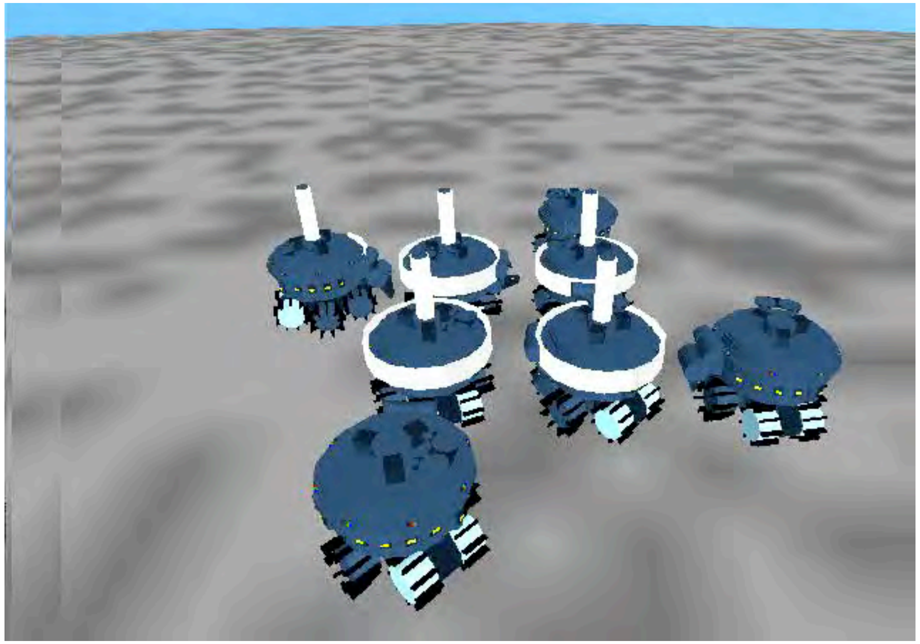
detailed medium simple

Matching detailed simulation model with real s-bot

Swarm-bots

Matching between simple and detailed models



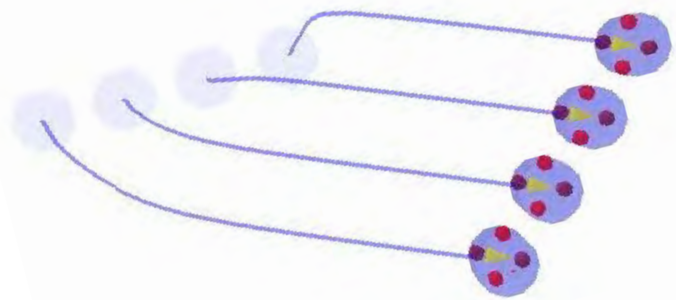


Definition of behaviors for the scenario

- Coordinated motion
- Self-assembly
- Cooperative transport
- Goal search and path formation

Coordinated motion

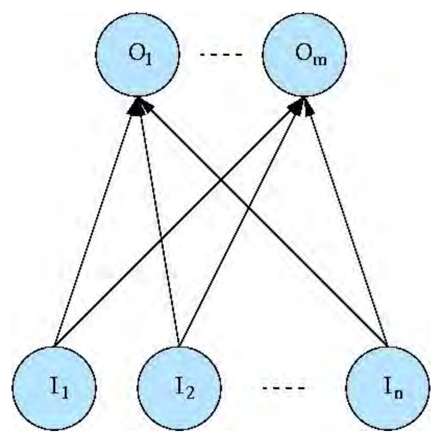
- Four ***s-bots*** are connected in a swarm-bot formation
- Their chassis are randomly oriented
- The ***s-bots*** should be able to
  - collectively choose a direction of motion
  - move as far as possible
- Single-layer perceptrons are evolved as controllers



The traction sensor

- Connected ***s-bots*** apply pulling/pushing forces to each other when moving
- Each ***s-bot*** can measure a traction force acting on its turret/chassis connection
- The traction force indicates the mismatch between
  - – the average direction of motion of the group
  - – the desired direction of motion of the single s-bot

turret



traction sensor

The evolutionary algorithm

- Binary encoded genotype
  - 8 bits per real valued parameter of the neural controllers
- Generational evolutionary algorithm
  - 100 individuals evolved for 100 generations
  - 20 best individuals are allowed to reproduce in each generation
  - Mutation (3% per bit) is applied to the offspring
- The perceptron is cloned and downloaded on each ***s-bot***
- Fitness is evaluated looking at the swarm-bots performance
  - Each individual is evaluated with equal starting conditions

Fitness evaluation

- The fitness F of a genotype is given by the distance covered by the group:

$$F = \frac{\| X(t) - X(0) \|}{D}$$

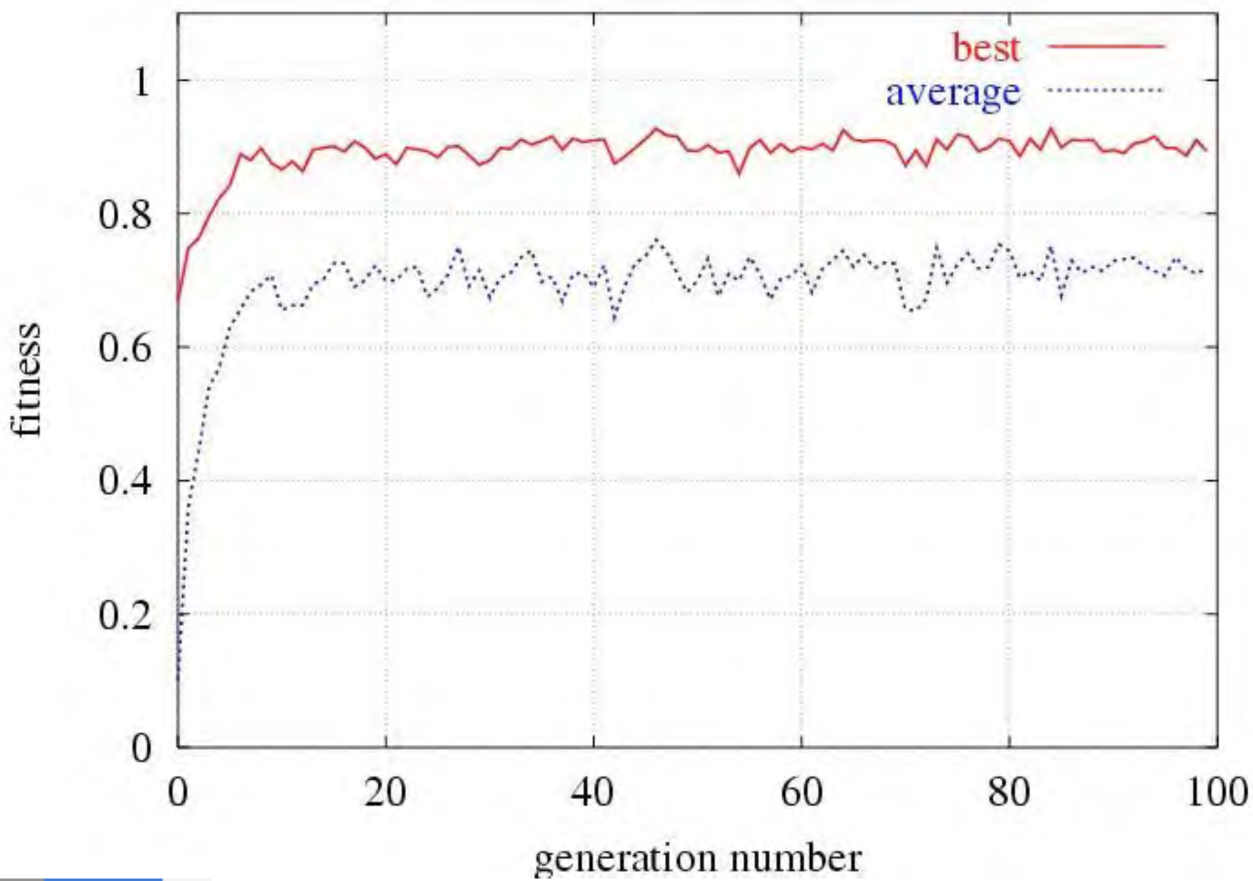
  where $X(t)$ is the coordinate vector of the center of mass at time t, and D is the maximum distance that can be covered in 150 simulation cycles

- Fitness is evaluated 5 times, starting from different random initializations
- The resulting average is assigned to the genotype

Swarm-bots: Coordinated motion

Results

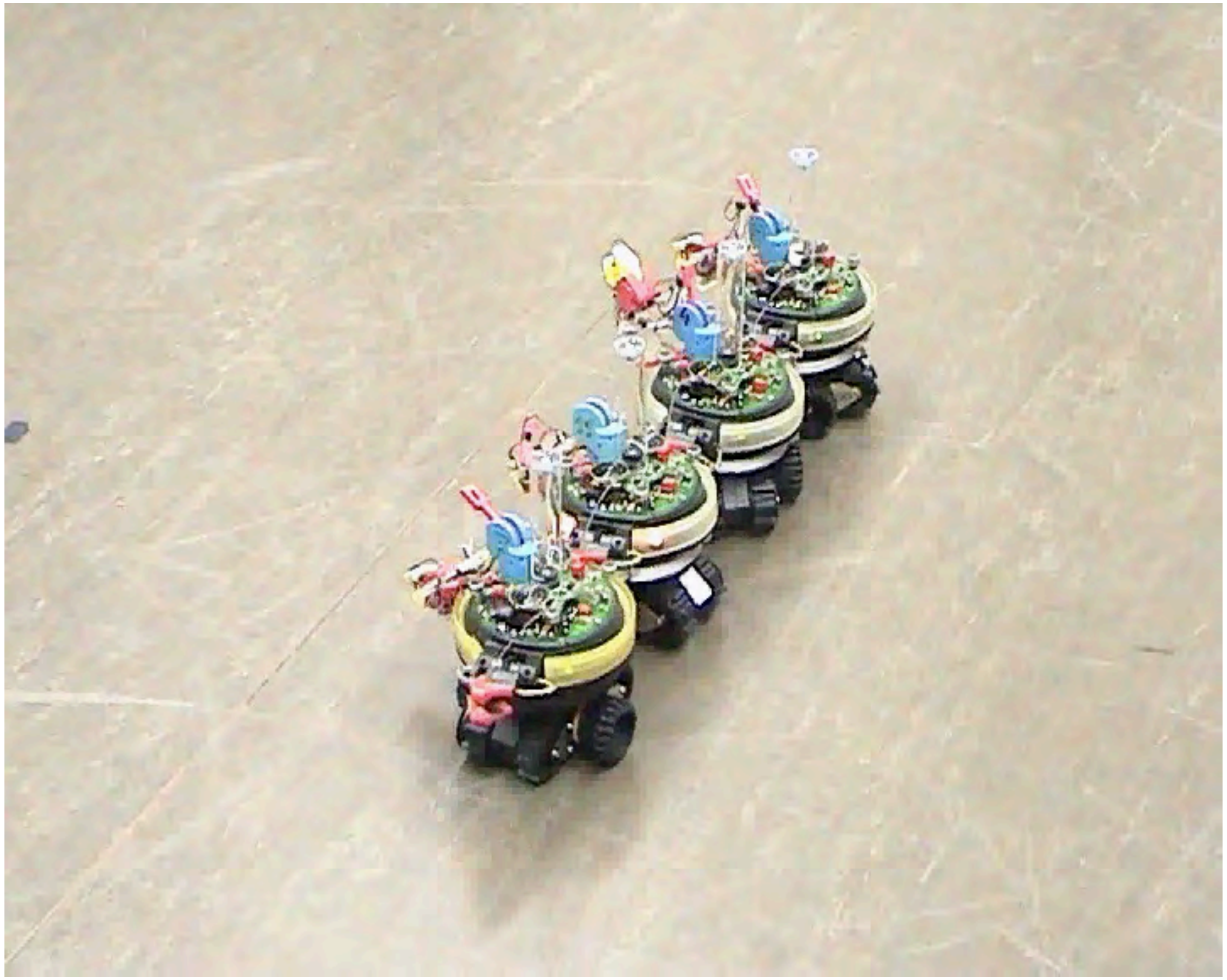
Average fitness



Post-evaluation

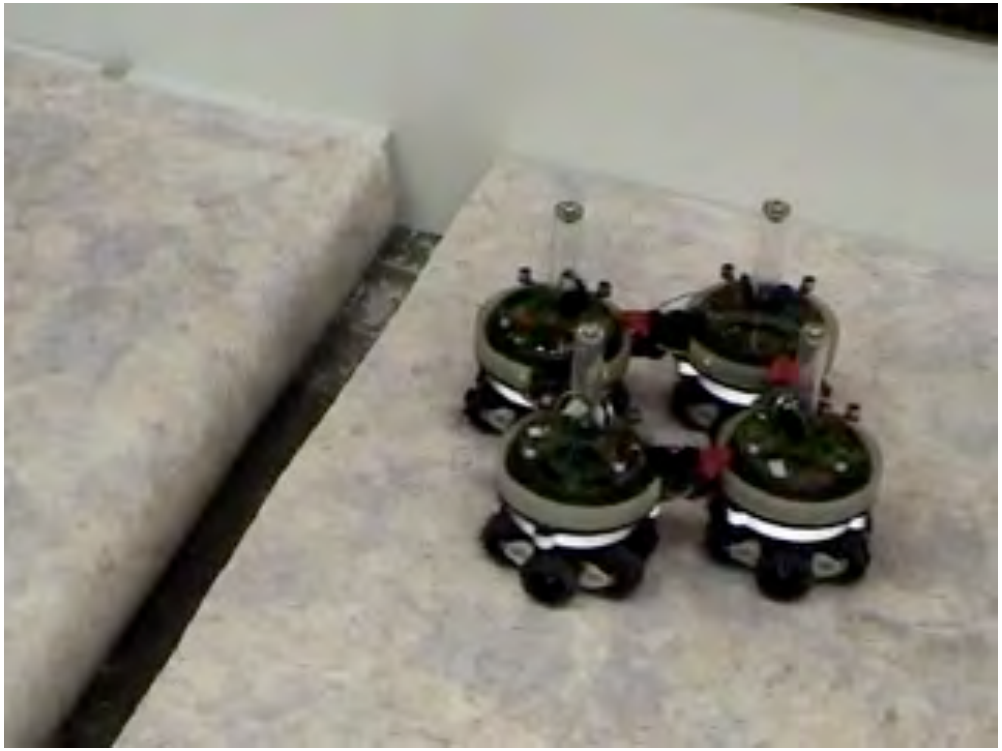
Replication	Performance
1	0.87888
2	0.83959
3	0.88338
4	0.71567
5	0.79573
6	0.75209
7	0.83425
8	0.85848
9	0.87222
10	0.76111

Porting to real s-bots



Swarm-bots: Coordinated motion

Real s-bots



flexibility

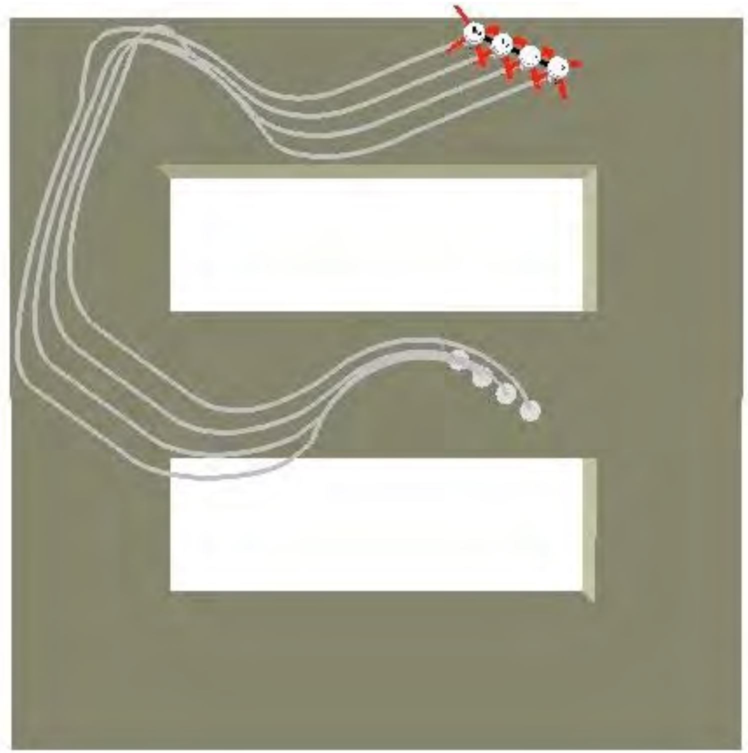
Scalability

scalability

flexibility and scalability

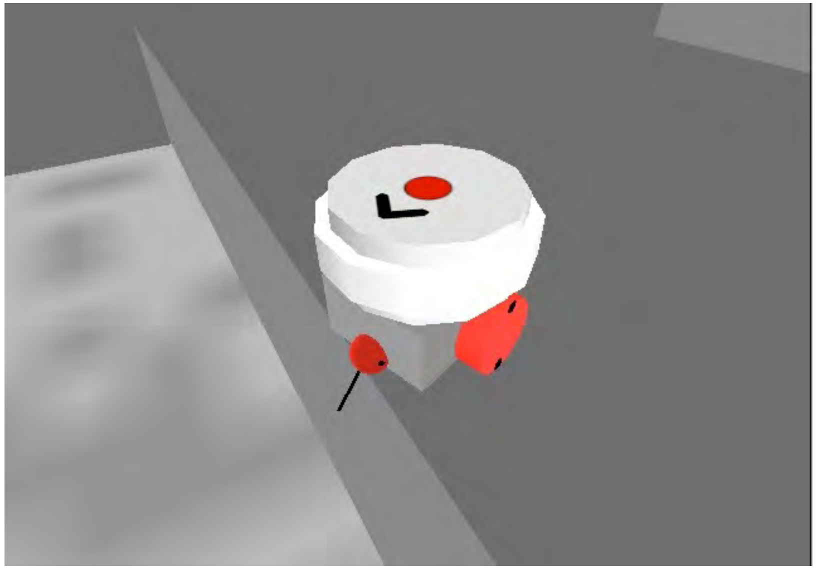
Hole avoidance

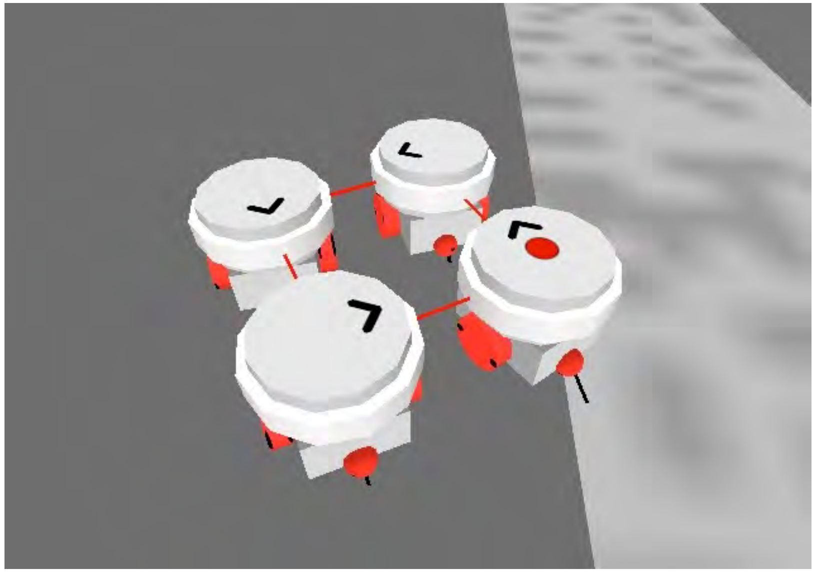
- Coordinated motion of the ***swarm-bot***
- Avoidance of holes (detection, communication, re-organization)



Swarm-bots: Hole avoidance

The s-bot model





- Sensors
  - Ground sensors → perception of the hazard
  - Traction sensor → perception of pulling/pushing forces
  - Microphones → perception of sound signals
- Actuators
  - Motors, loudspeaker

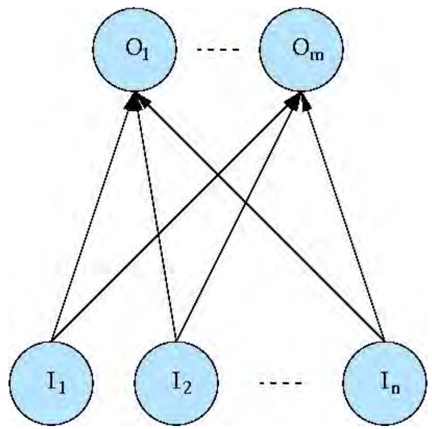
Evolution of communication

- We provide ***s-bots*** with the possibility to explicitly signal the presence of a hole
  - Activation of the speaker directly linked to the ground sensor activation (reflex)
  - The microphones are used to detect if any ***s-bot*** is signaling the presence of a hole (one additional input to the NN)
- Evolution is in charge of shaping the reaction to the sound signal
- Communication speeds up the collective avoidance after the detection of a hole

The evolutionary algorithm

- Generational evolutionary algorithm
  - 100 individuals evolved for 200 generations
  - 20 best individuals are allowed to reproduce in each generation
  - Mutation is applied to the offspring (5% mutation prob for each gene)

- Binary Encoded Genotype
  - 8 bits per connection weight of the neural network controller



- The neural network is cloned and downloaded on each ***s-bot***
  - Each genotype is evaluated 12 times
  - Different environments and different ***swarm-bot*** configurations

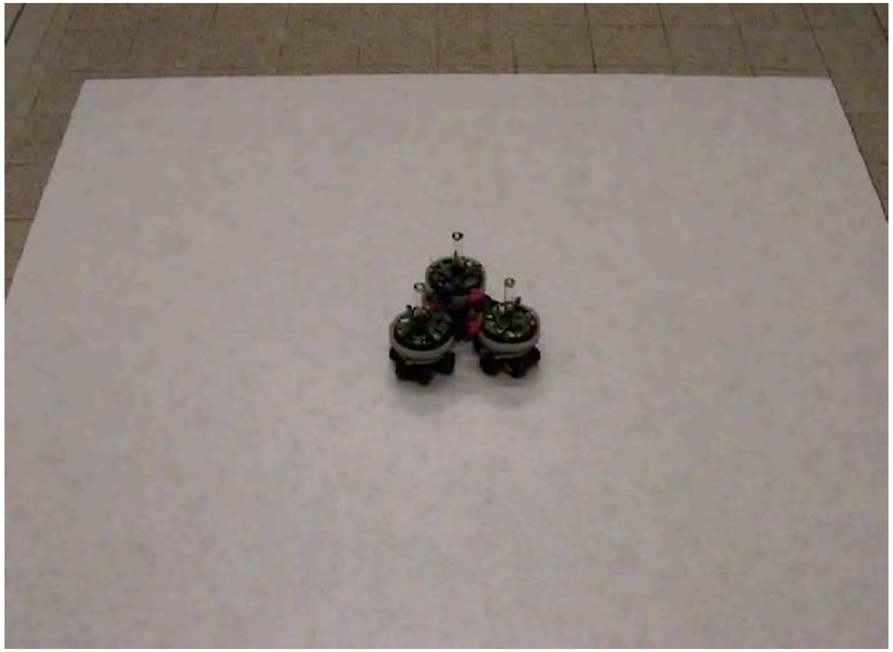
The fitness computation

- Fitness is computed using only local information directly accessible to the ***s-bots***
- Individual fitness is a function of these components:
  - Coordinated motion and hole avoidance
    - Minimization of perceived traction
    - Minimization of time spent near a hole
  - Fast motion (wheels' speed sum)
  - Straight motion (wheels' speed difference)
- The fitness of the genotype corresponds to the worst individual fitness among the ***s-bots***

Swarm-bots: Hole avoidance

Porting to real s-bots

- No substantial difference between the simulated and the real-world controller: The neural controller evolved in simulation is directly used to control the real ***s-bots***
- Test with real swarm-bots
  - Square arena with open borders (1.8 m side)
  - 3 s-bots in triangular formation
  - 20 replication
- Obtained results:
  - Avoidance successful (not a single fall)
  - Coordination always achieved
  - Coordinated motion works even if one ***s-bots*** is partially suspended



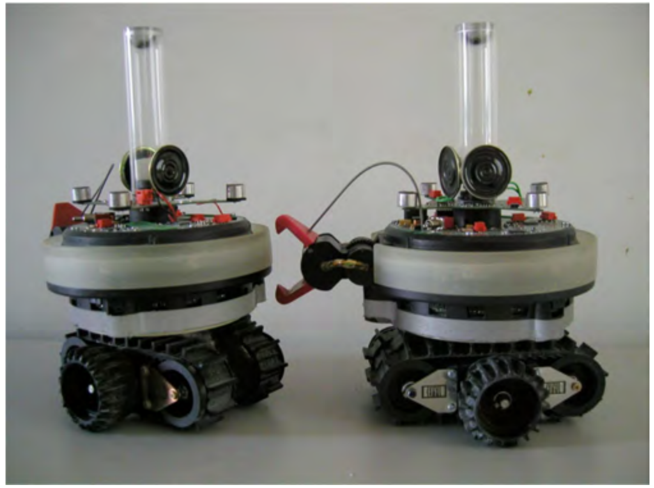
Swarm-bots

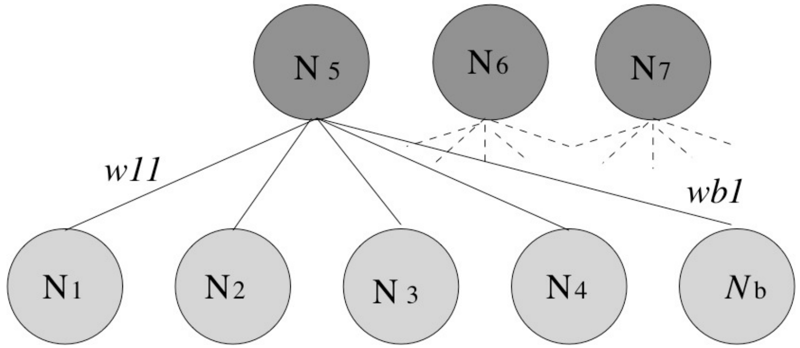
Self-assembly

Algorithm I - The assembly module

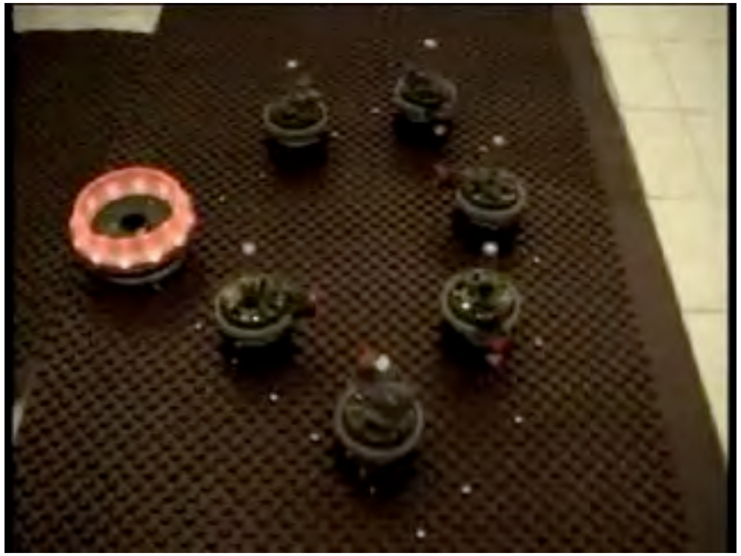
```
activate colour ring in blue
do
    (N1, N2) ← featureExtraction(camera)
    (N3, N4) ← sensorReadings(proximity)
    (N5, N6, N7) ← neuralNetwork(N1, N2, N3, N4)

    if (N7 > 0.5) ∧ (grasping requirements fulfilled)
      then
            grasp
            if (successfully connected)
              then
                      activate colour ring in red
                      activate transport module
              else
                      open gripper
            fi
    fi
    apply (N5, N6) to tracks
while (timeout not reached)
```

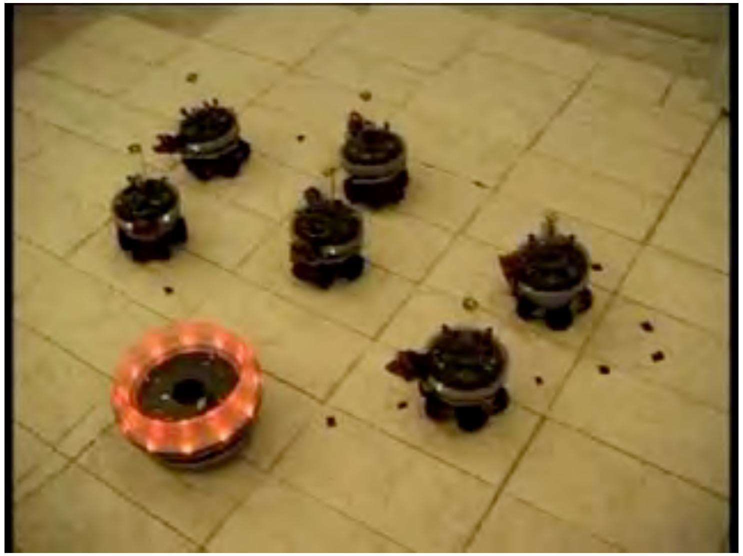




Six s-bots and a prey



flexibility



flexibility

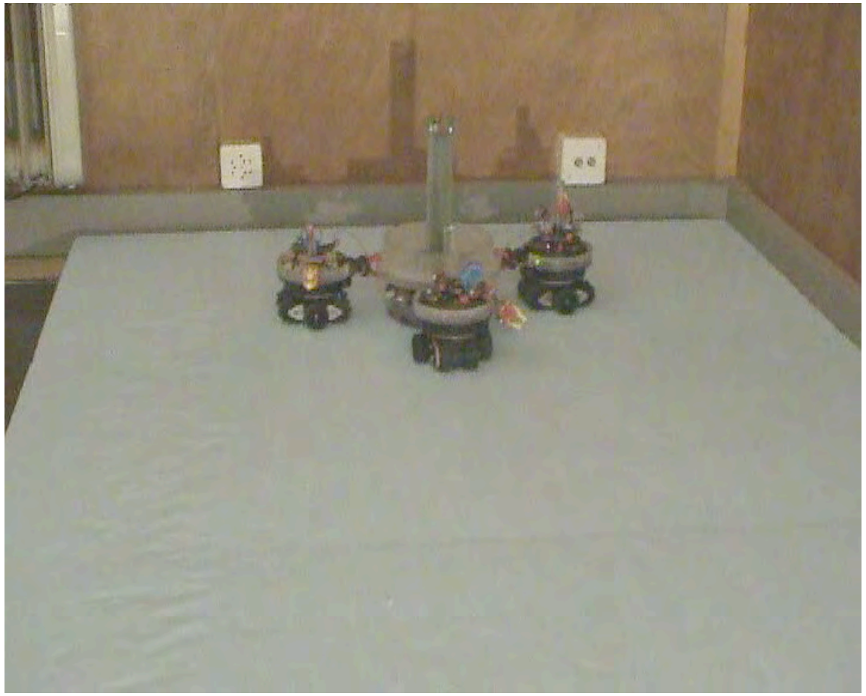
scalability

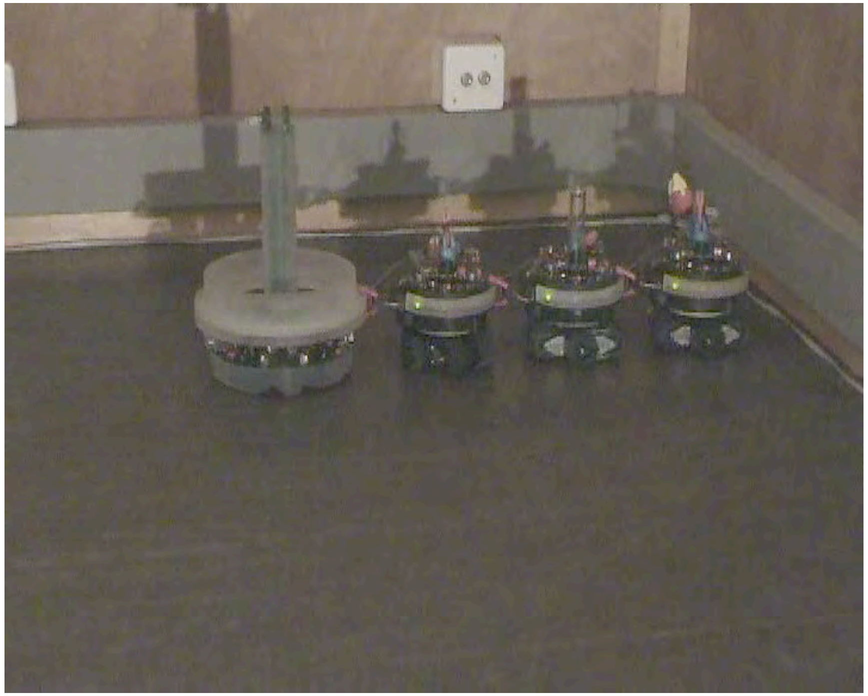
Cooperative transport

- **Goal:**
  - **Let a swarm-bot transport an object to a goal location**
- **Control**
  - **Designed phototaxis behavior**

Swarm-bots: Cooperative transport

Pre-assembled swarm-bots





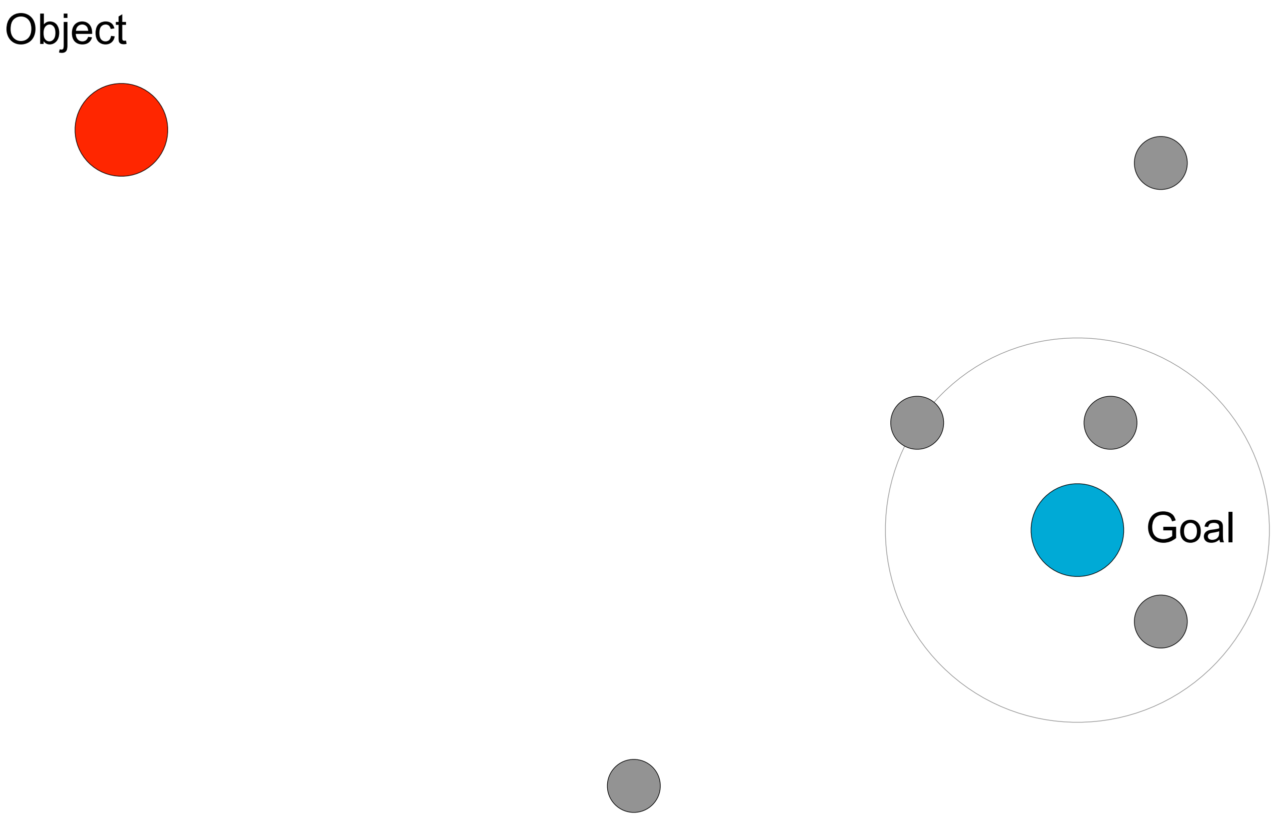
Swarm-bots: Cooperative transport

Self-assembly and transport

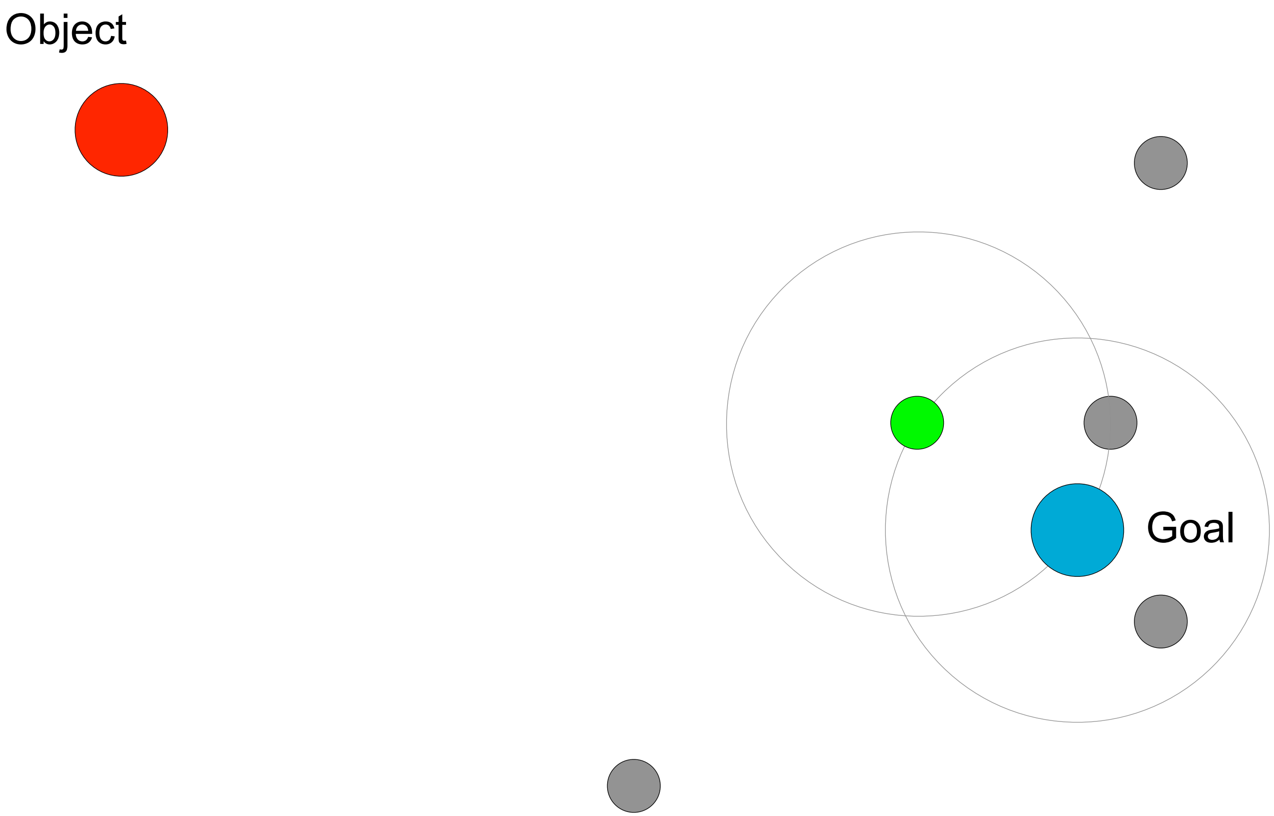
Path formation

- Our robots have limited sensing capabilities:
  - Can distinguish 3 colors (approx up to 30 cm away)
  - Can say which color is closer
- We want to mimic ants trail formation, but ***s-bots*** cannot lay pheromones
- We use ***s-bots*** instead of pheromones

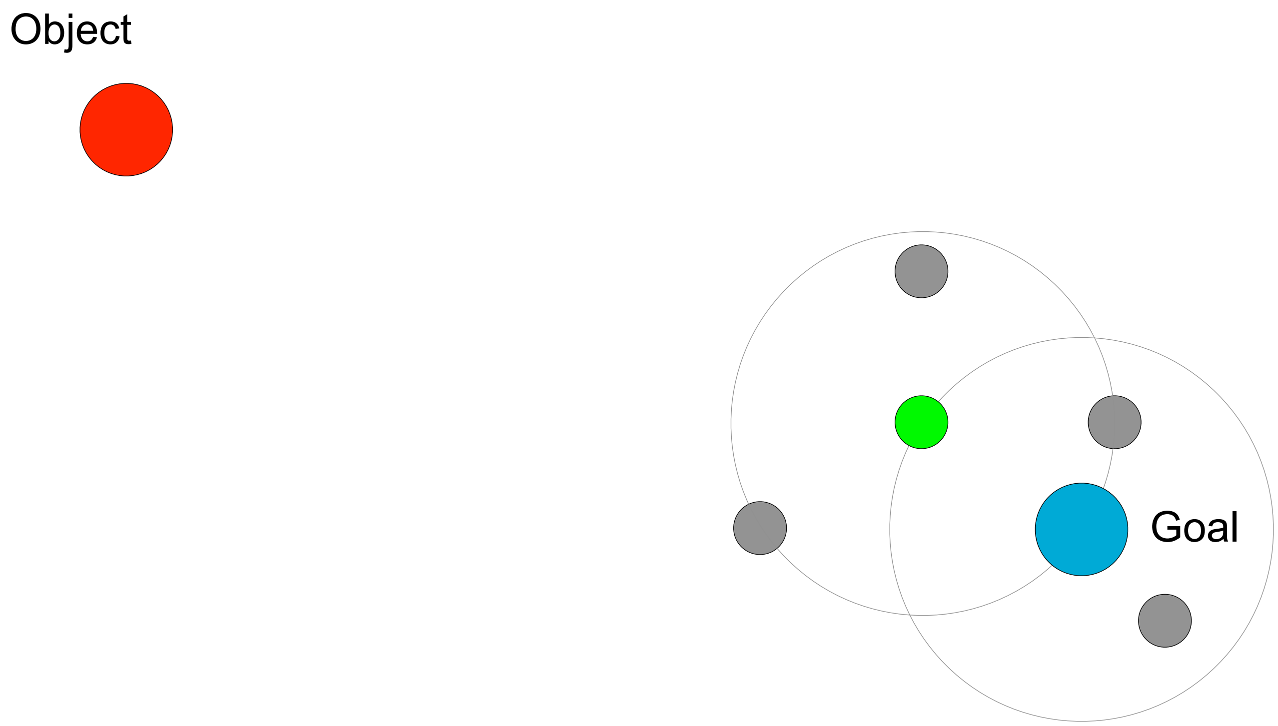
Path formation



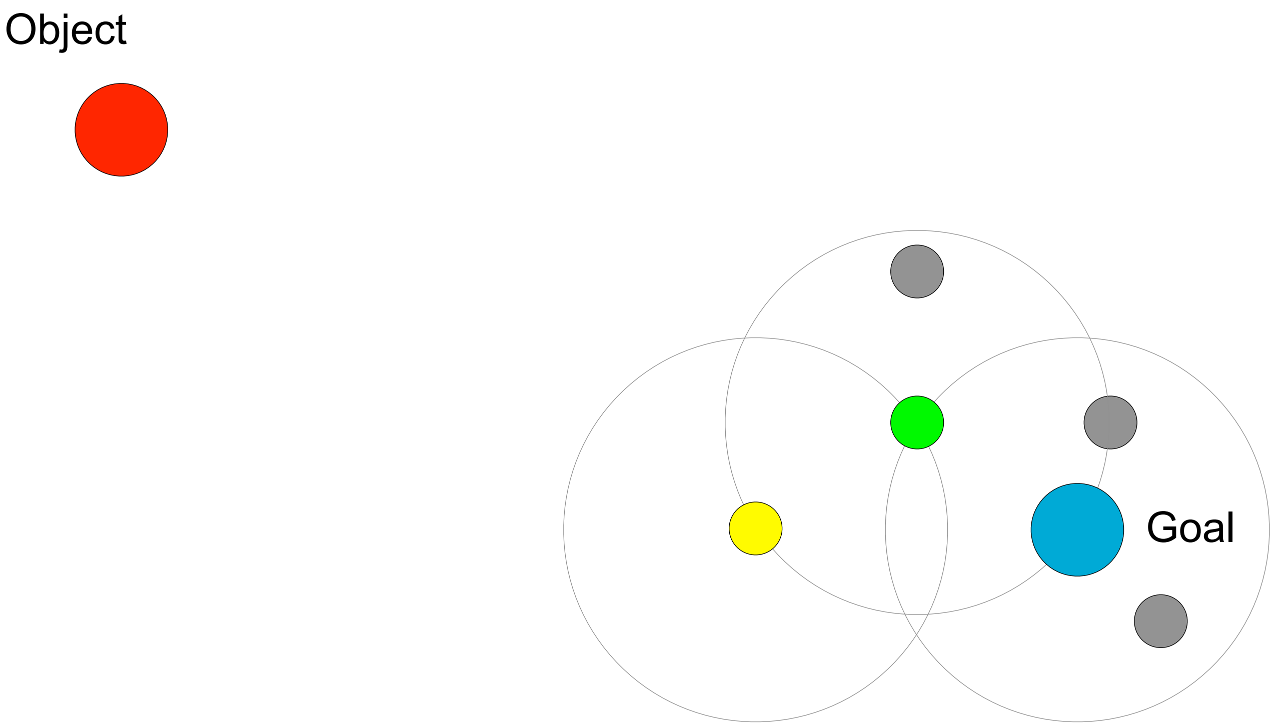
Path formation



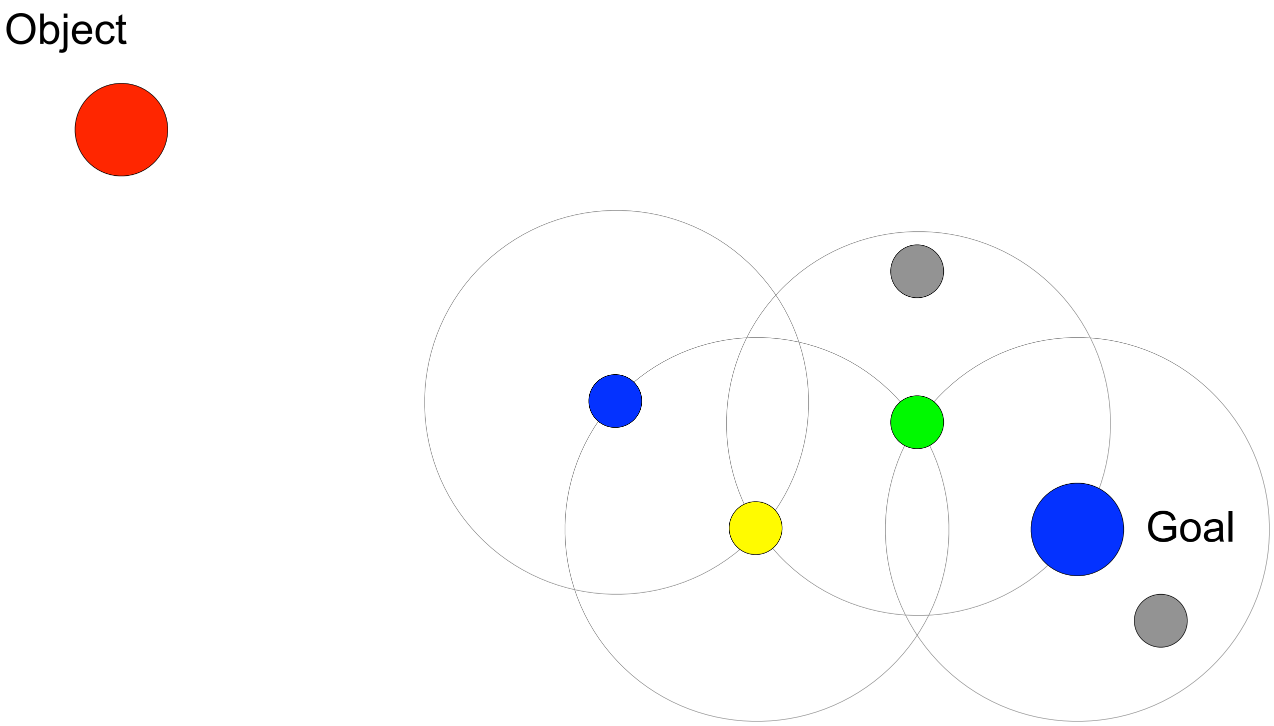
Path formation



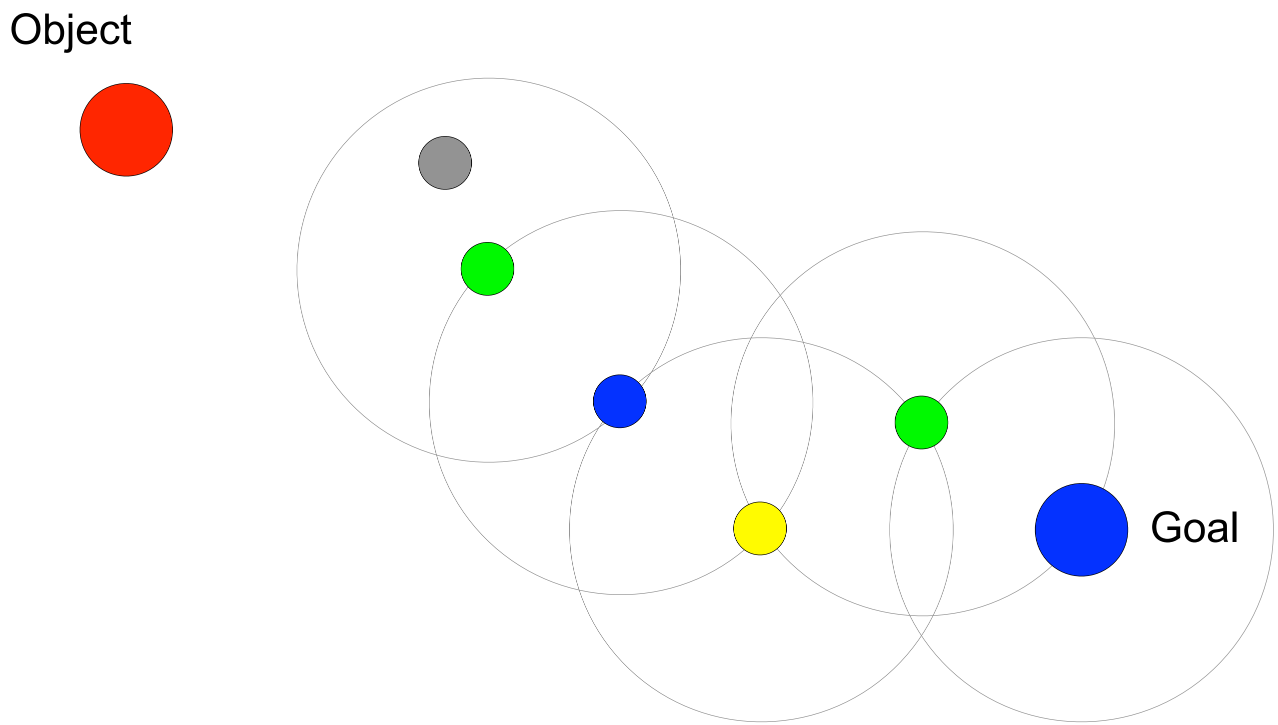
Path formation



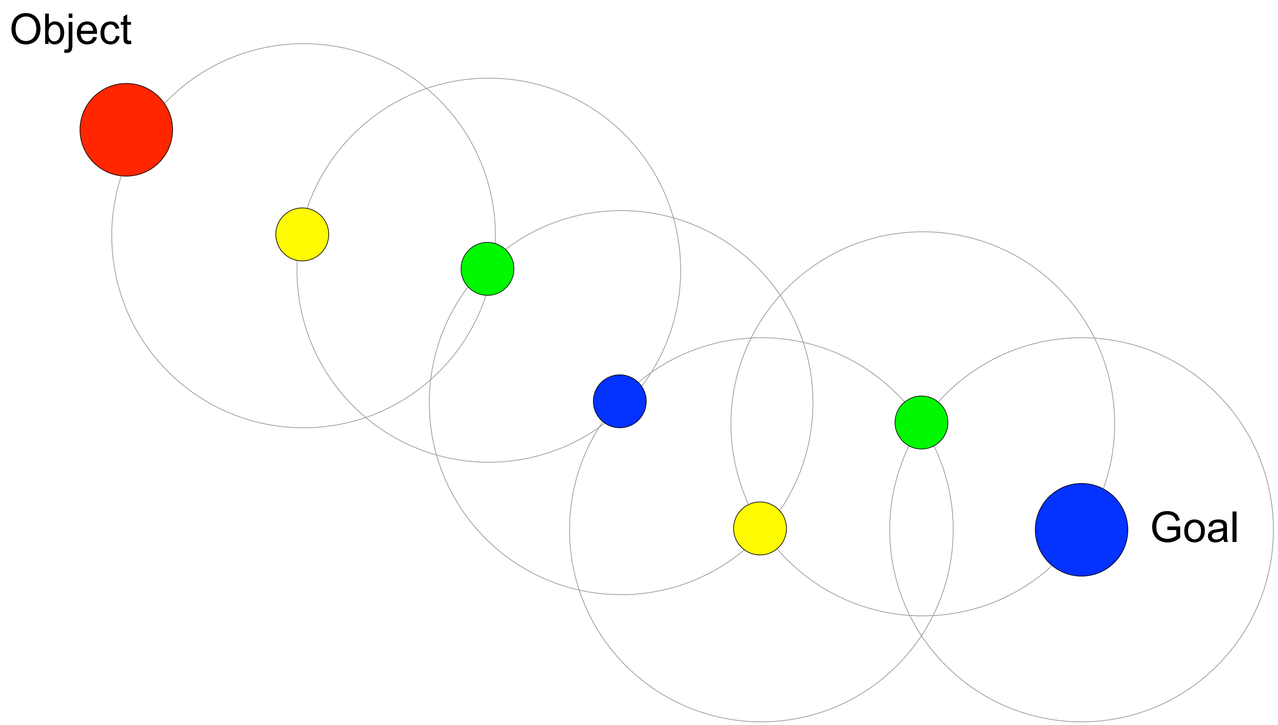
Path formation



Path formation

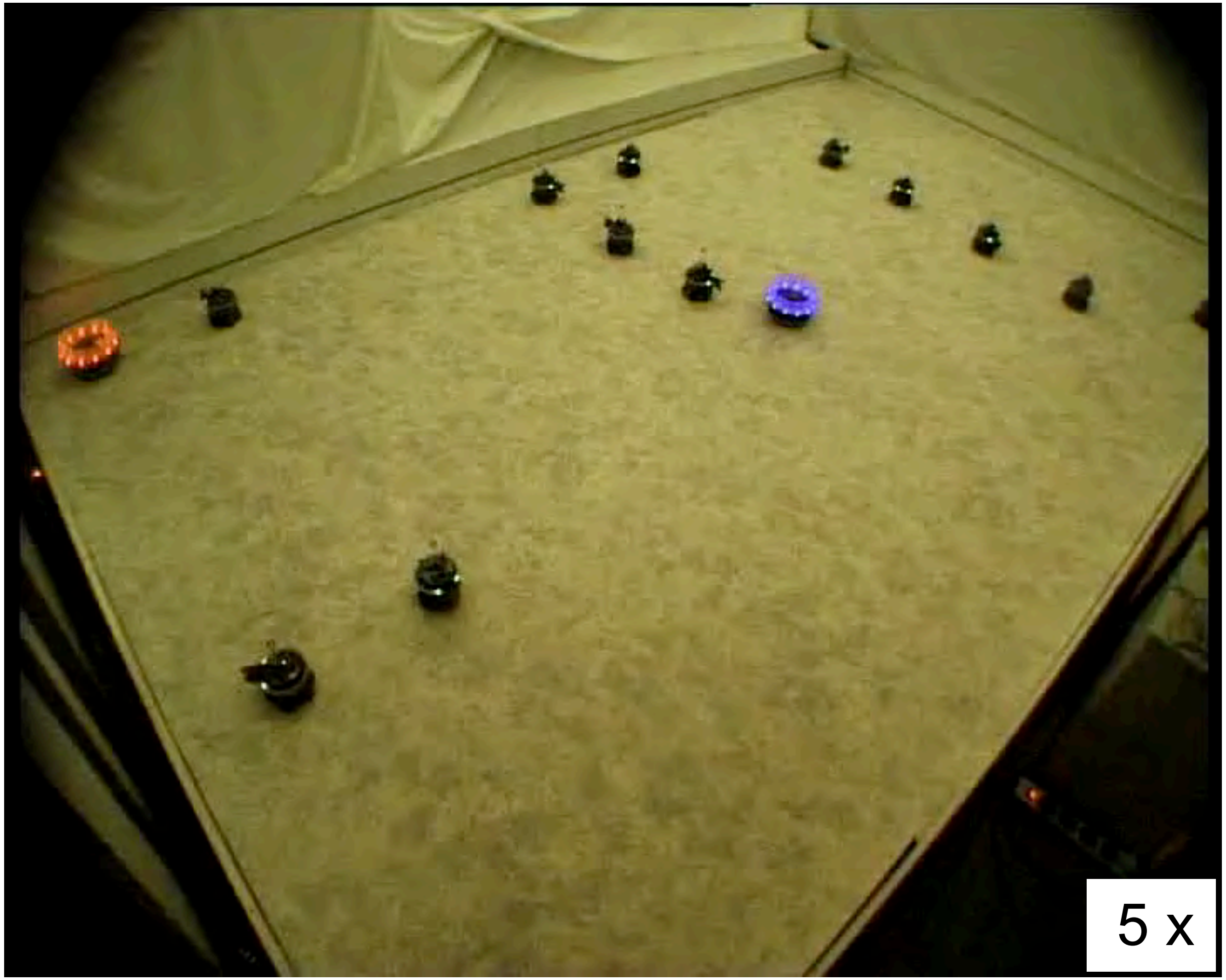


Path formation



Goal search and path formation

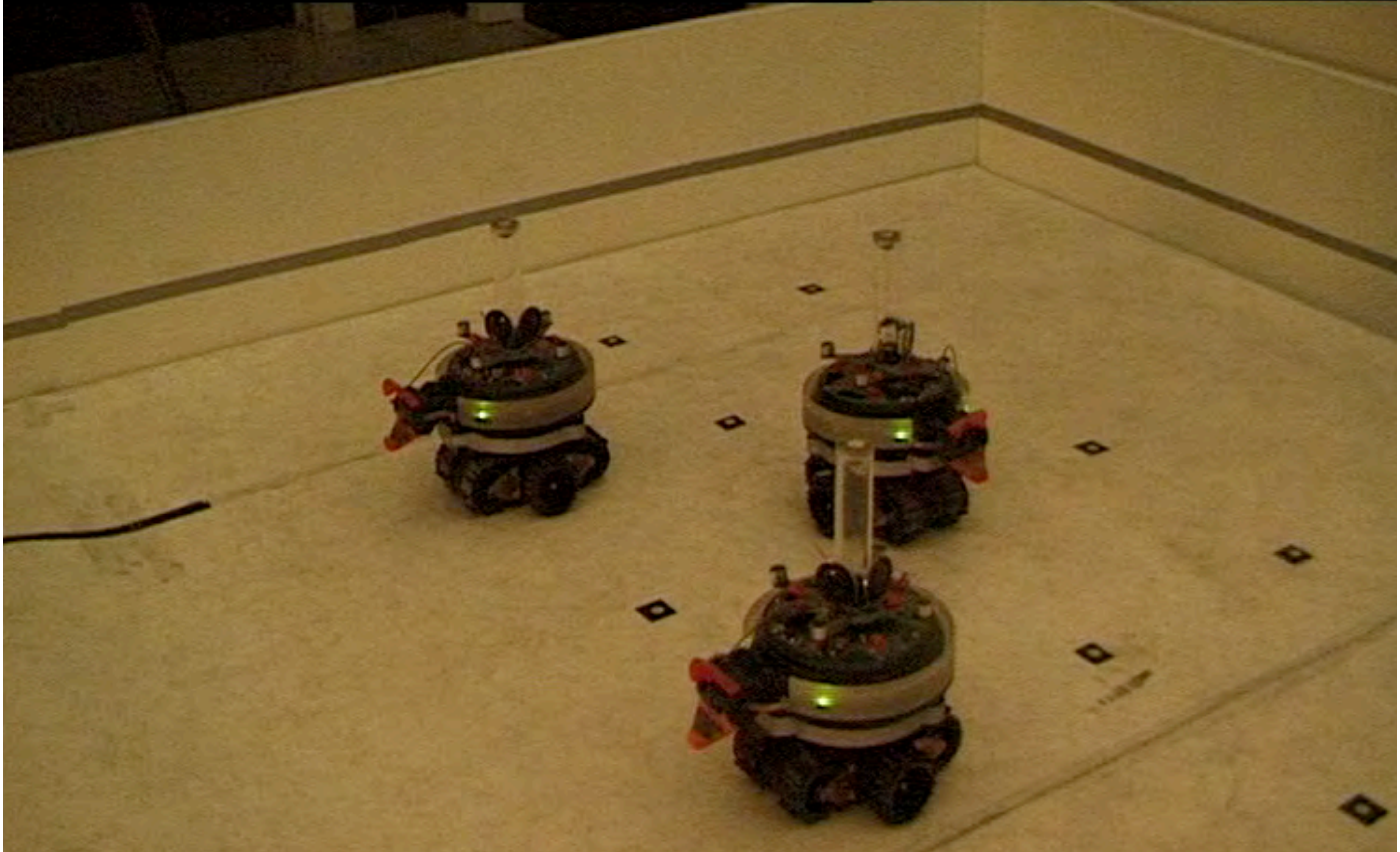
Path formation and retrieval



Ongoing research

- Functional self-assembly
- Morphology formation
- Swarm level fault detection

Functional self-assembly



S-bots can pass a low hill

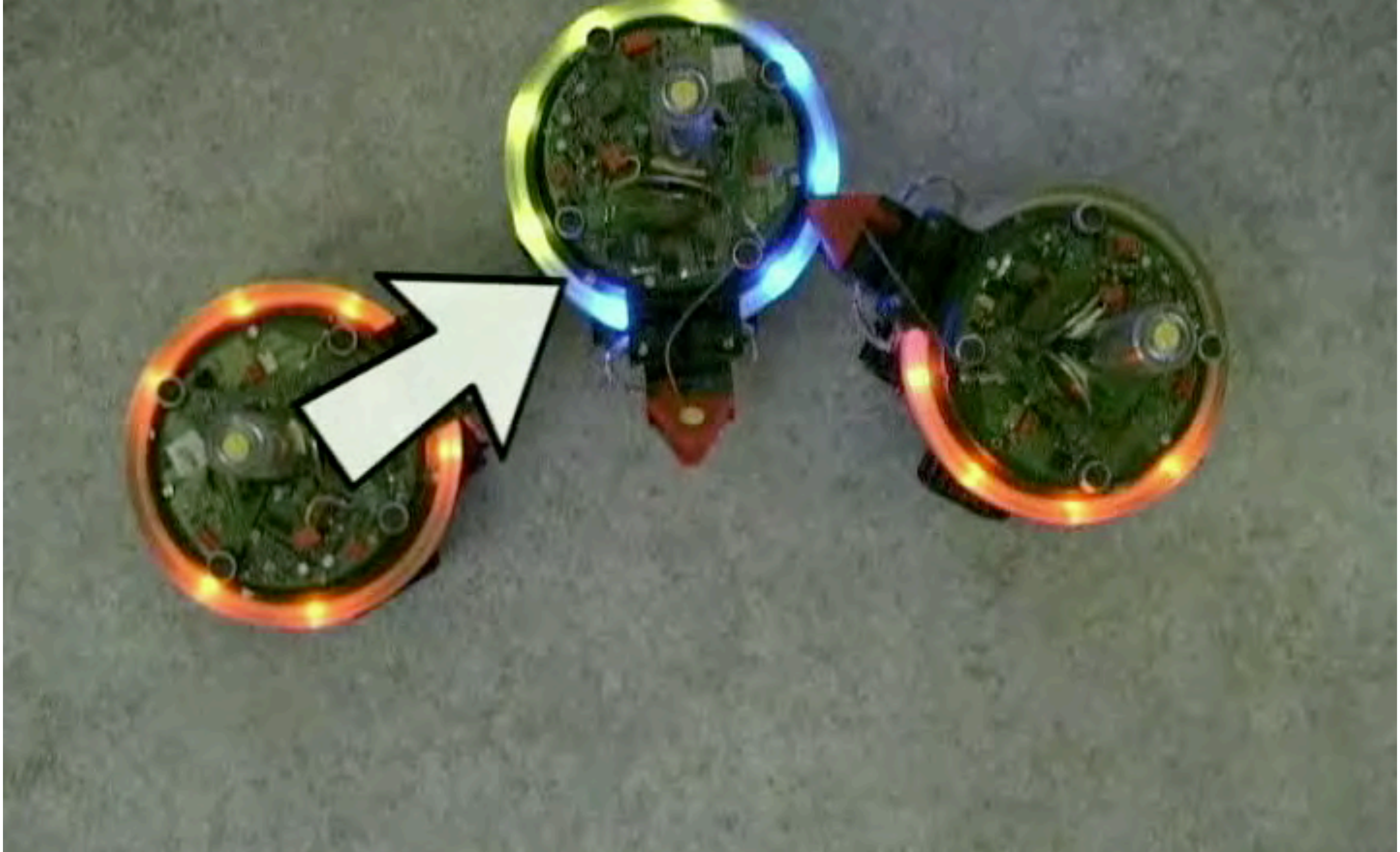
An s-bot cannot pass a high hill

A swarm-bot can pass a high hill

Swarm-bot toppling

5x

Swarm-bot’s adaptive rotation



Swarm-bot's morphogenesys

Example of application of morphology control

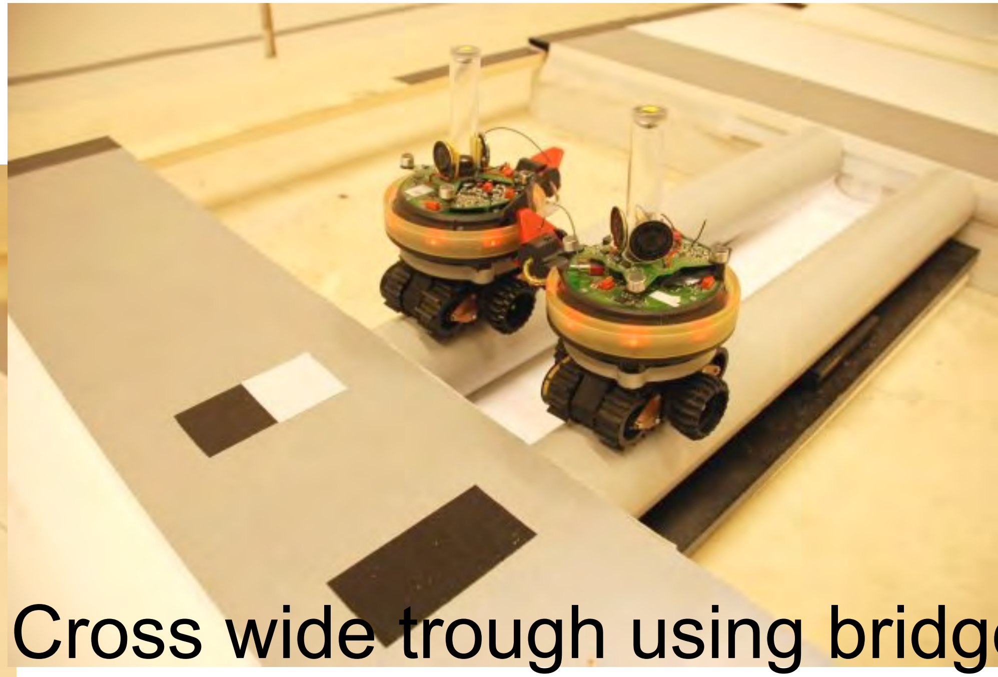
- A same robotic system has to solve different tasks

Our experiment:

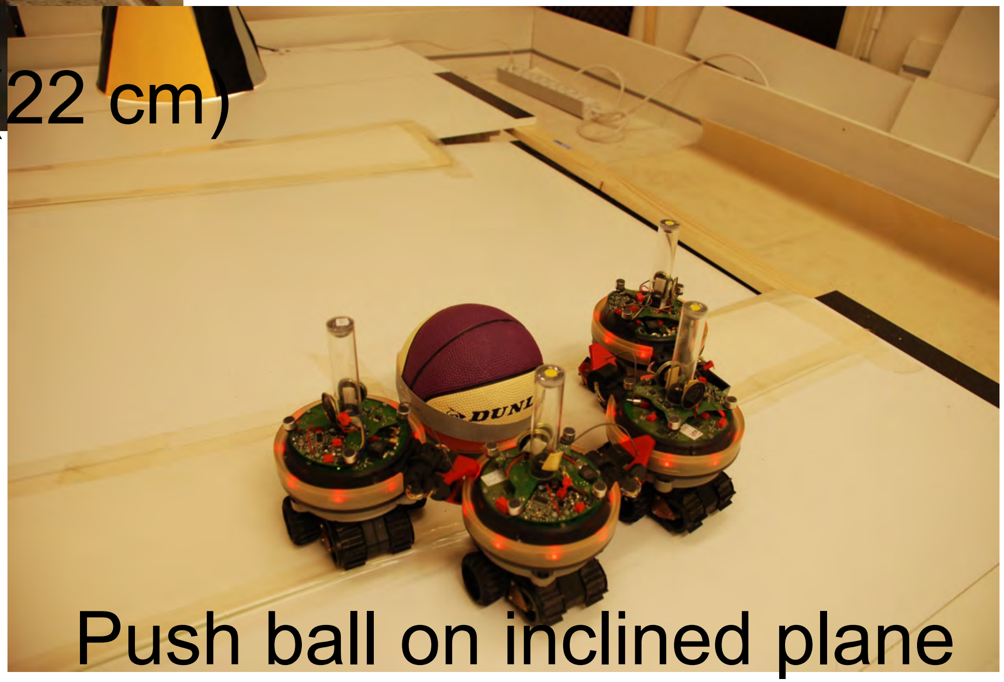
- 3 different tasks to be solved one after the other
  – No a priori knowledge of task sequence
  – Each task only solvable by dedicated morphology

The tasks

Cross narrow trough (22 cm)

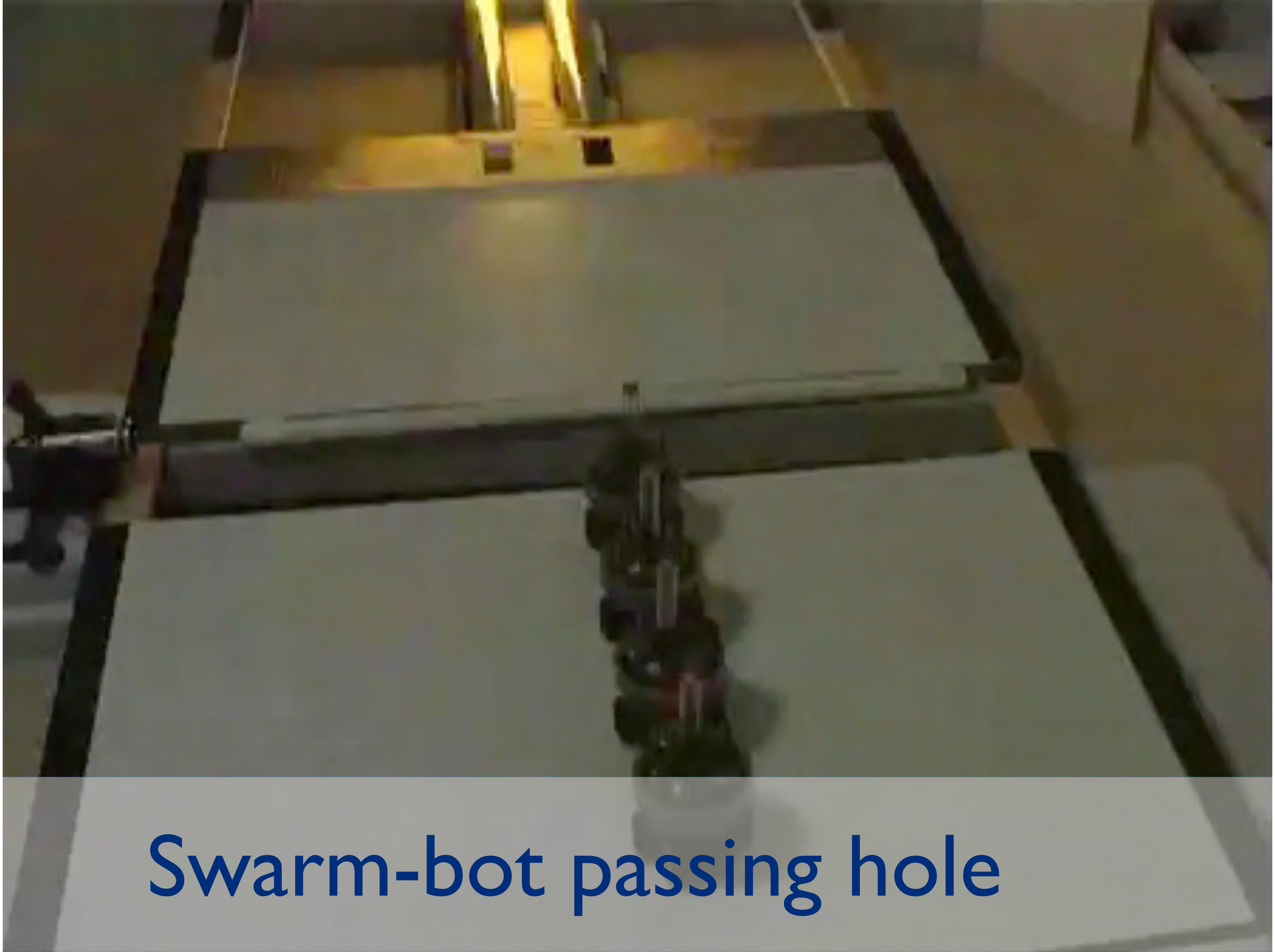


Cross wide trough using bridge

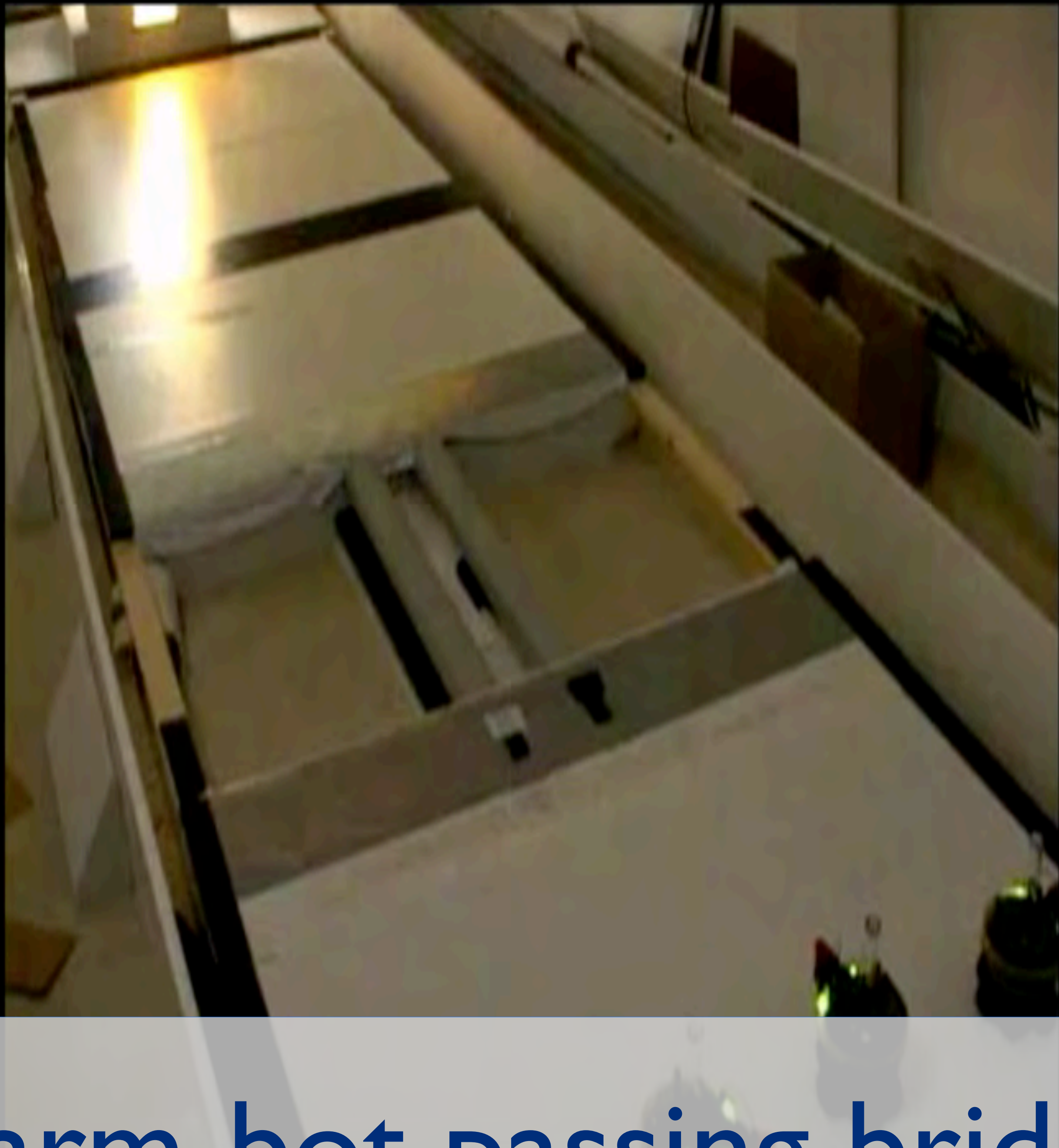


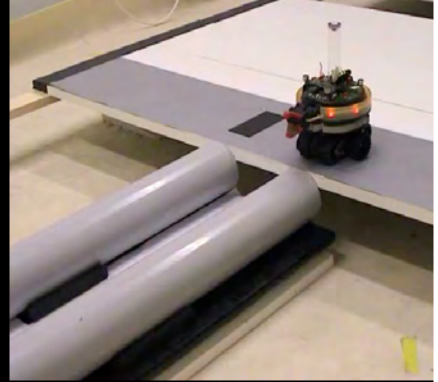
Push ball on inclined plane

Functional self-assembly



Swarm-bot passing hole





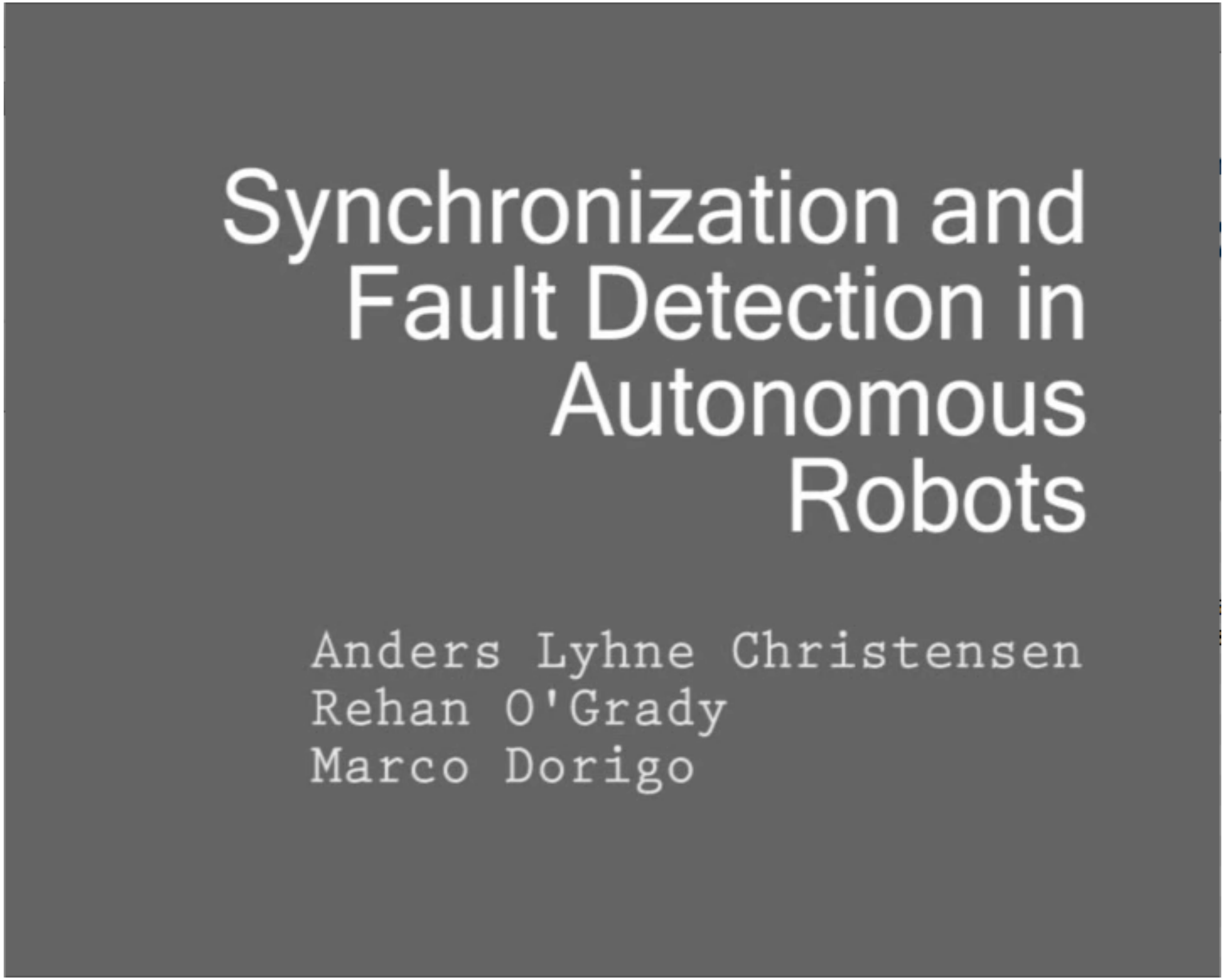
Swarm-bot passing bridge

Swarm-bot passing bridge

Swarm level fault detection inspired by fireflies behavior

- Each robot is given a heartbeat (i.e., robots flash periodically)
- A robot can stop flashing
  - either because it is broken
  - or as a way to signal other robots that it is faulty (it can realize its faultiness using some endogenous fault detection mechanism)
- When a robot stops flashing, other robots consider it as faulty

Firefly inspired fault detection in a swarm of robots



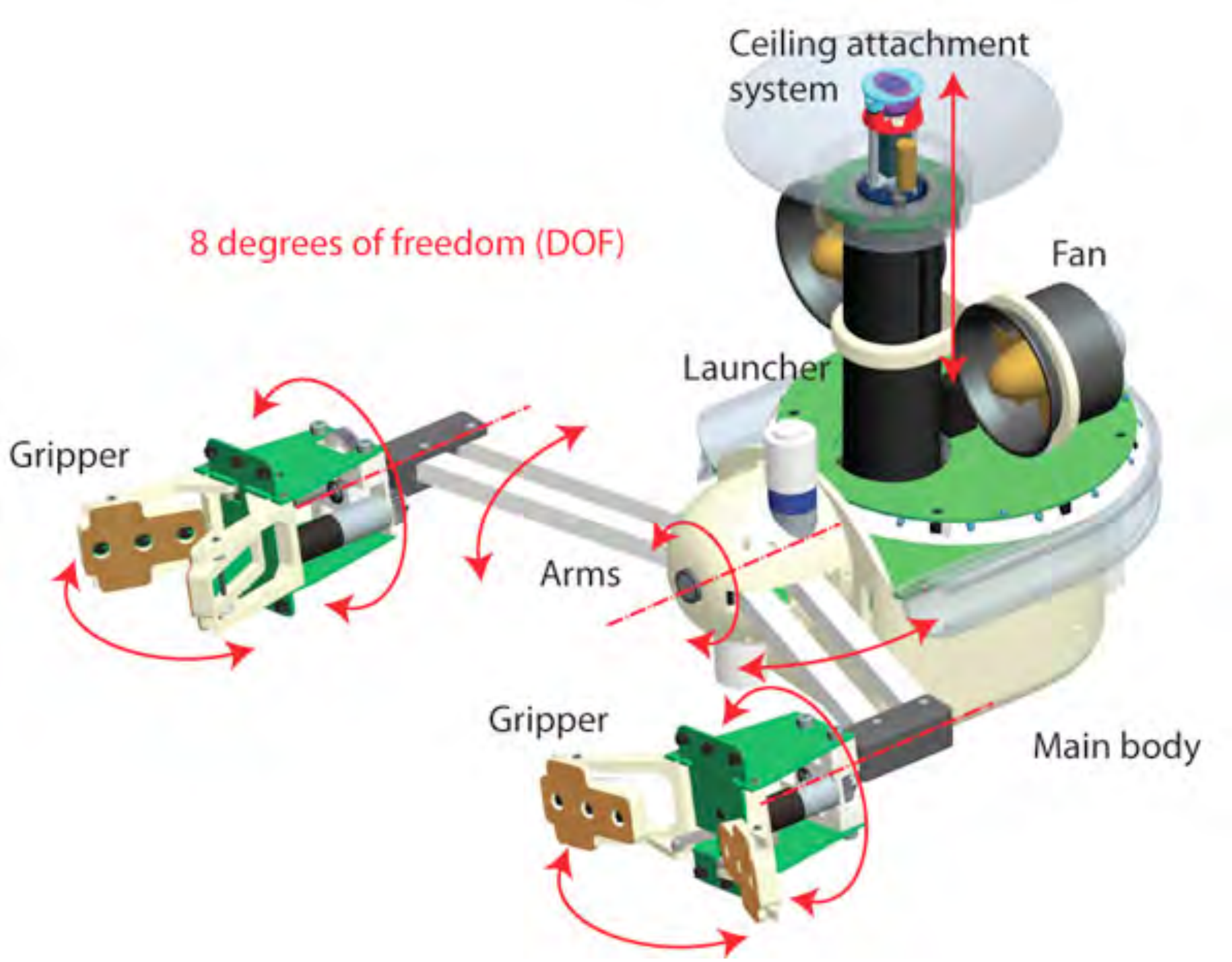
The Swarmanoid experiment

A swarmanoid is composed of:

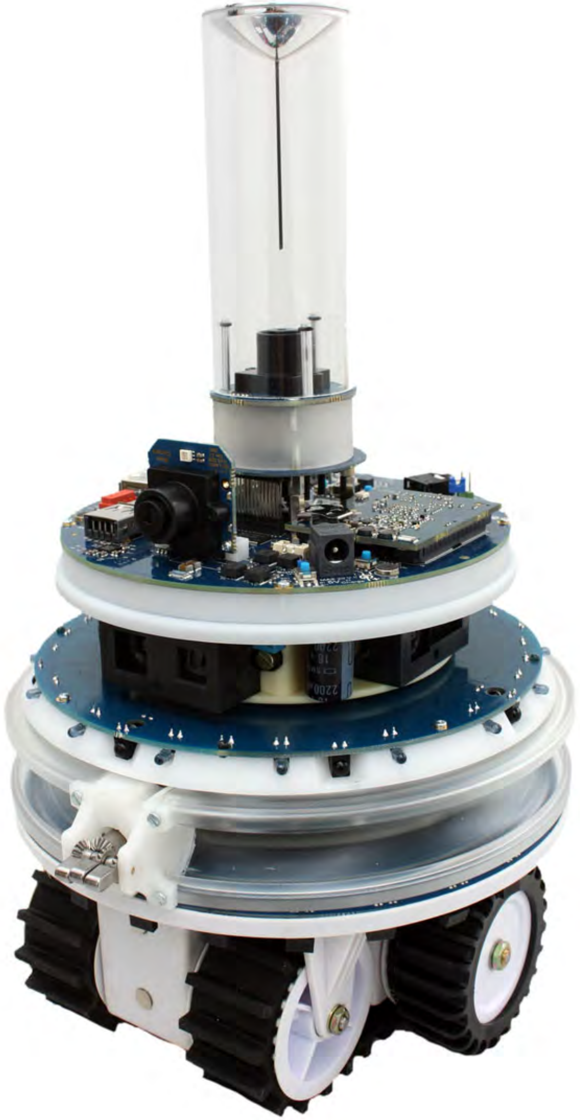
- Hand-bots
- Foot-bots
- Eye-bots

A swarmanoid is a heterogeneous swarm acting in 3D space

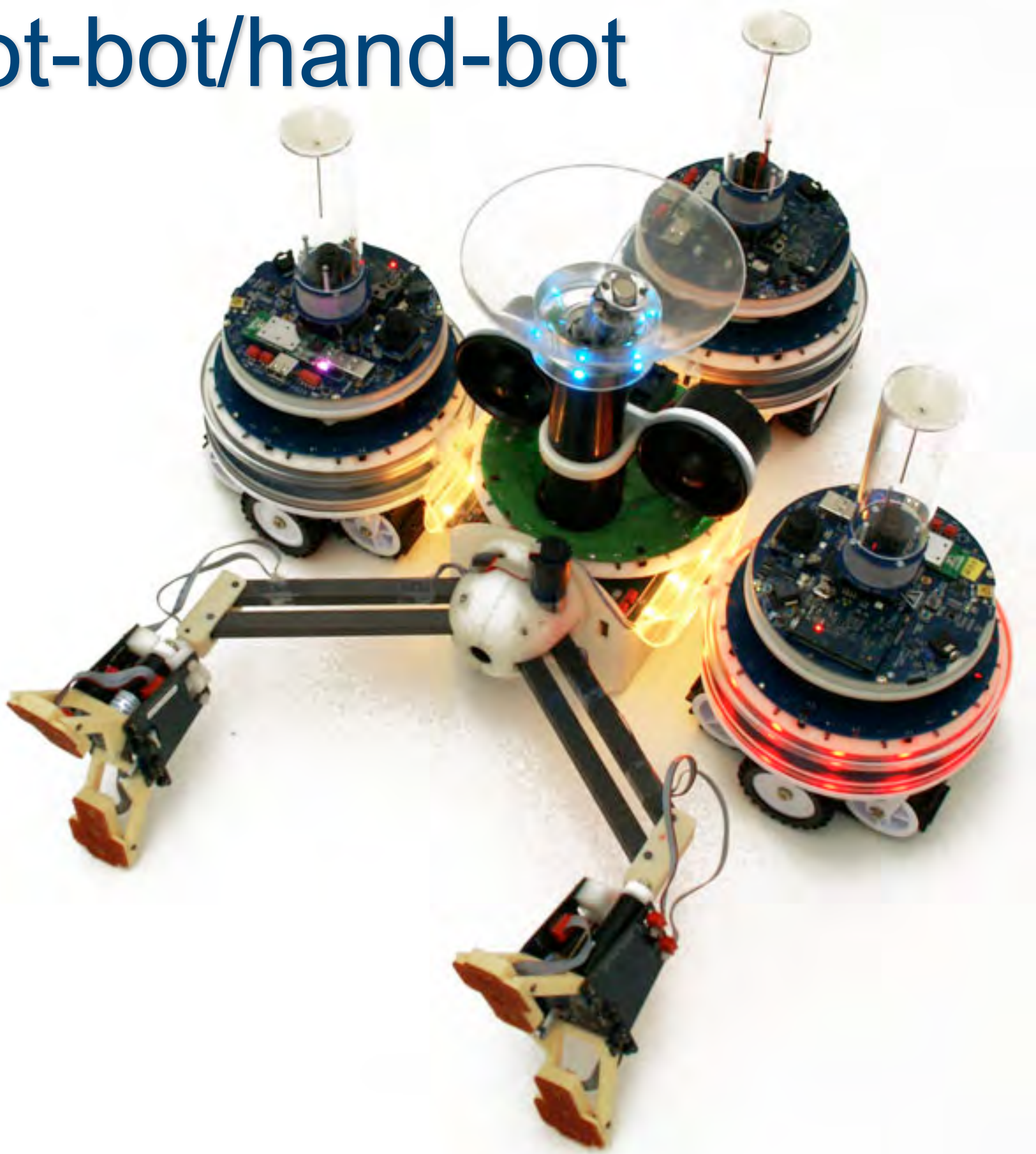
Hand-bot


Ceiling attachment
system
8 degrees of freedom (DOF)
Fan
Launcher
Gripper
Arms
Gripper
Main body

Foot-bot



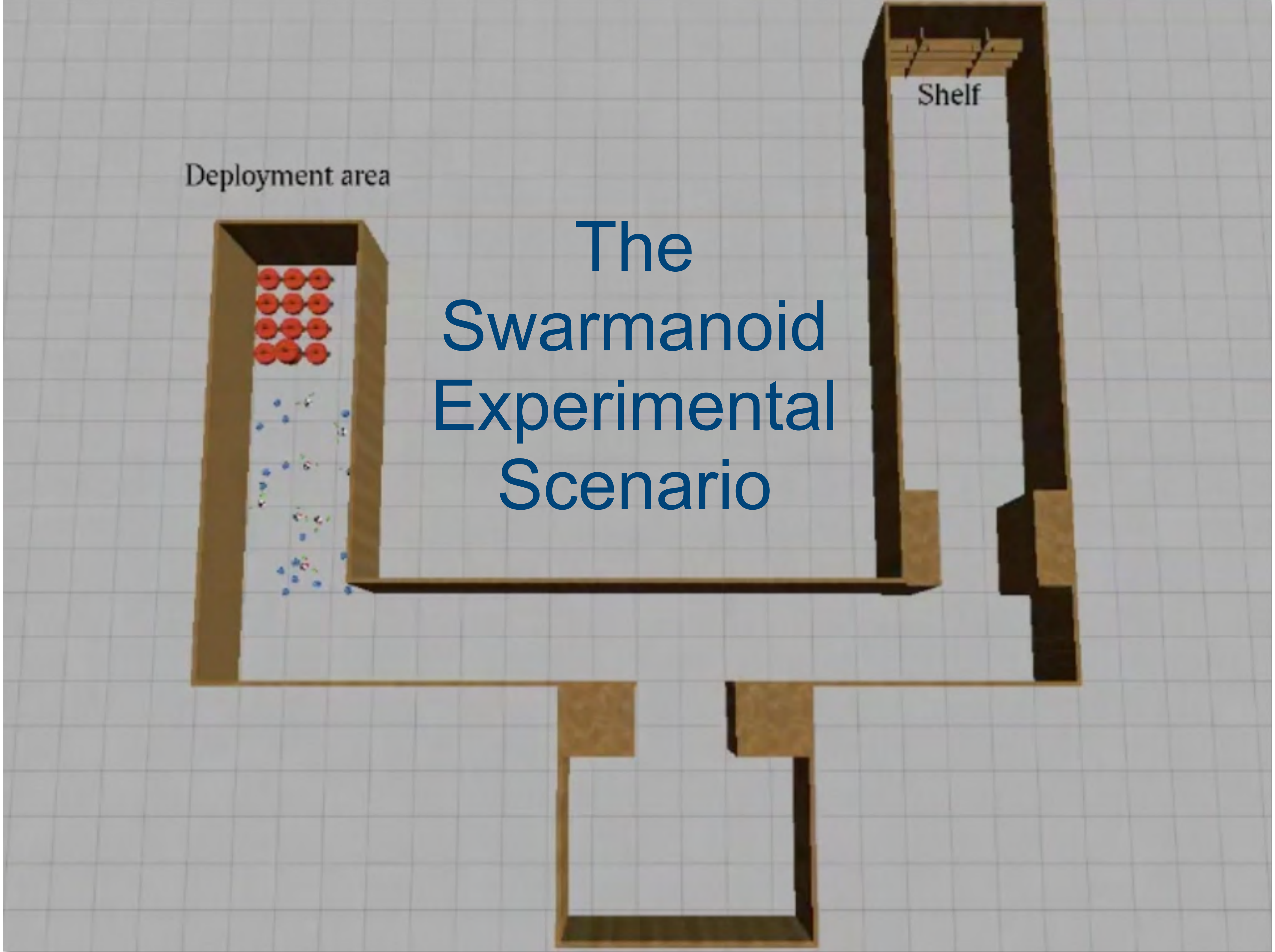
Foot-bot/hand-bot



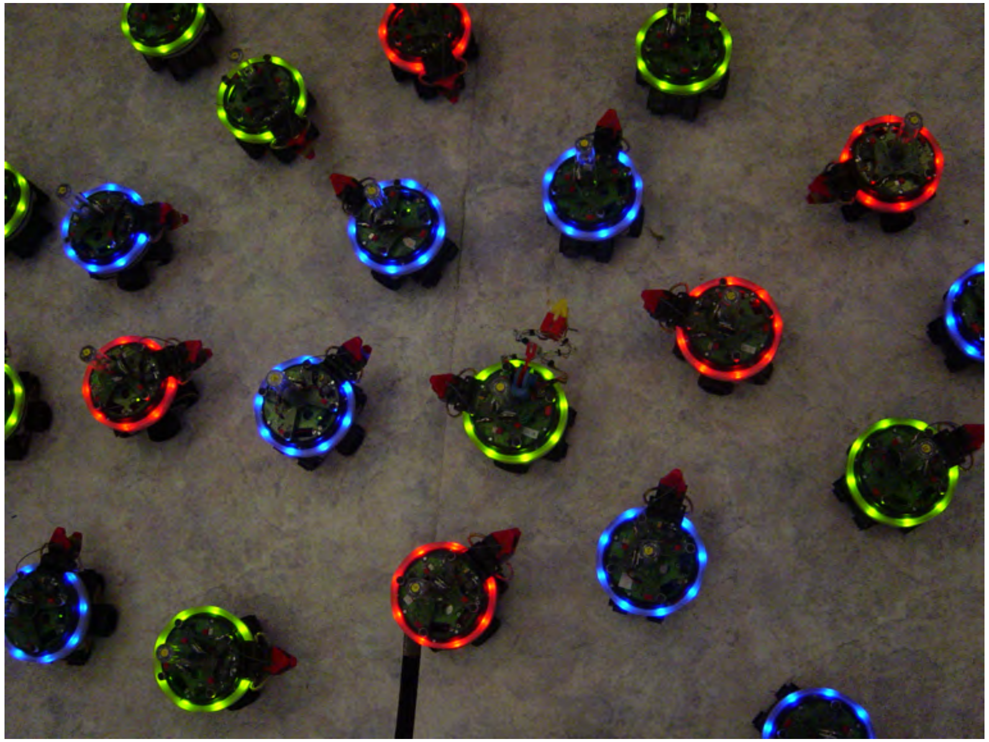
Eye-bot

Spatially targeted communication and self-assembly

The Swarmanoid Experimental Scenario



The end



www.swarm-bots.org
www.swarmanoid.org