Evolutionary Robotics

A practical guide to experiment with Thymio II robots

PPSN 2016, Edinburgh

Jacqueline Heinerman
Julien Hubert
Gusz Eiben
Evert Haasdijk

slides via www.heinerman.nl

Part I (25min)

Evolutionary Robotics: goals, current research and challenges

Part II (35min)

How to start your own experiments with Thymio II robots

Part III (30min-2hour)

Stay and get experience coding the Thymio robot

(Note: laptop required)

Evolutionary Robotics: goals and current challenges

“Evolutionary Robotics aims to apply evolutionary computation techniques to evolve the overall design or controllers, or both, for real and simulated autonomous robots.”

Vargas et al. 2014

Mainstream robotics:

Aims to generate good behavior for a given robot

→ Good design

2[nd] order engineering

Evolutionary robotics:

Aims to create general, robot-generating algorithms

→Good designer

After Bongard, 2013

Why?

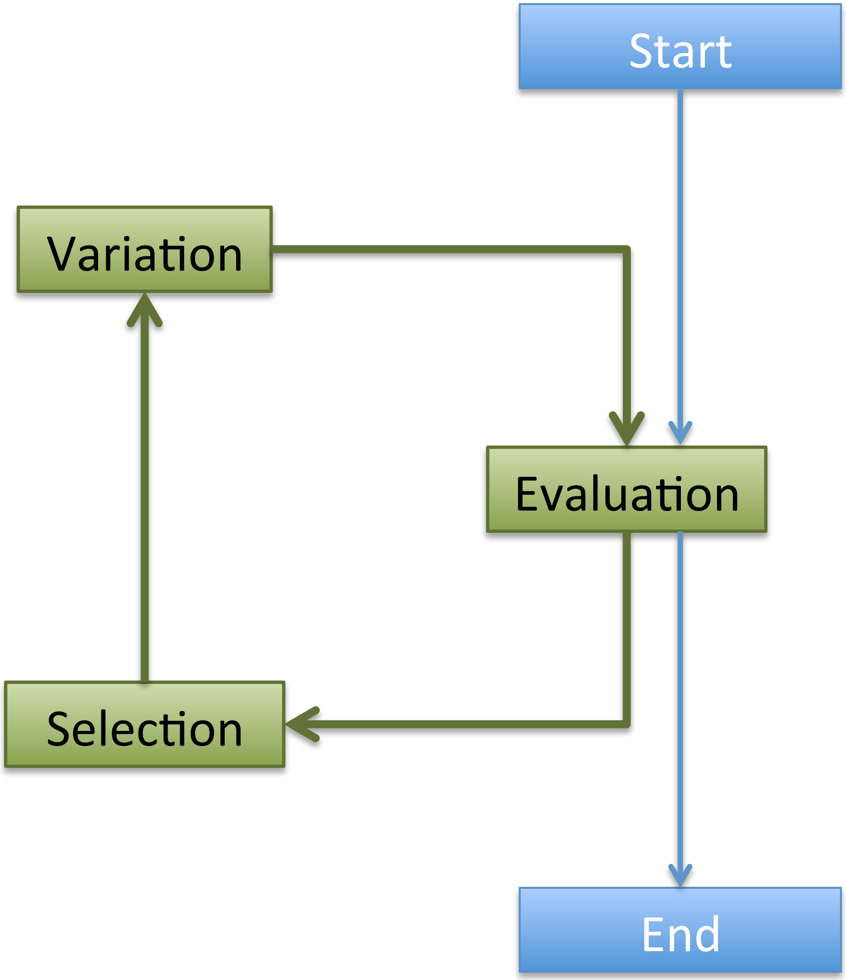
Designing robots is a hard problem because:

- Link between design variables and behaviour is complex and noisy
- Not all situations can be foreseen by designers upfront optimal behaviour is not known upfront
- Situation may change → optimal behaviour may change

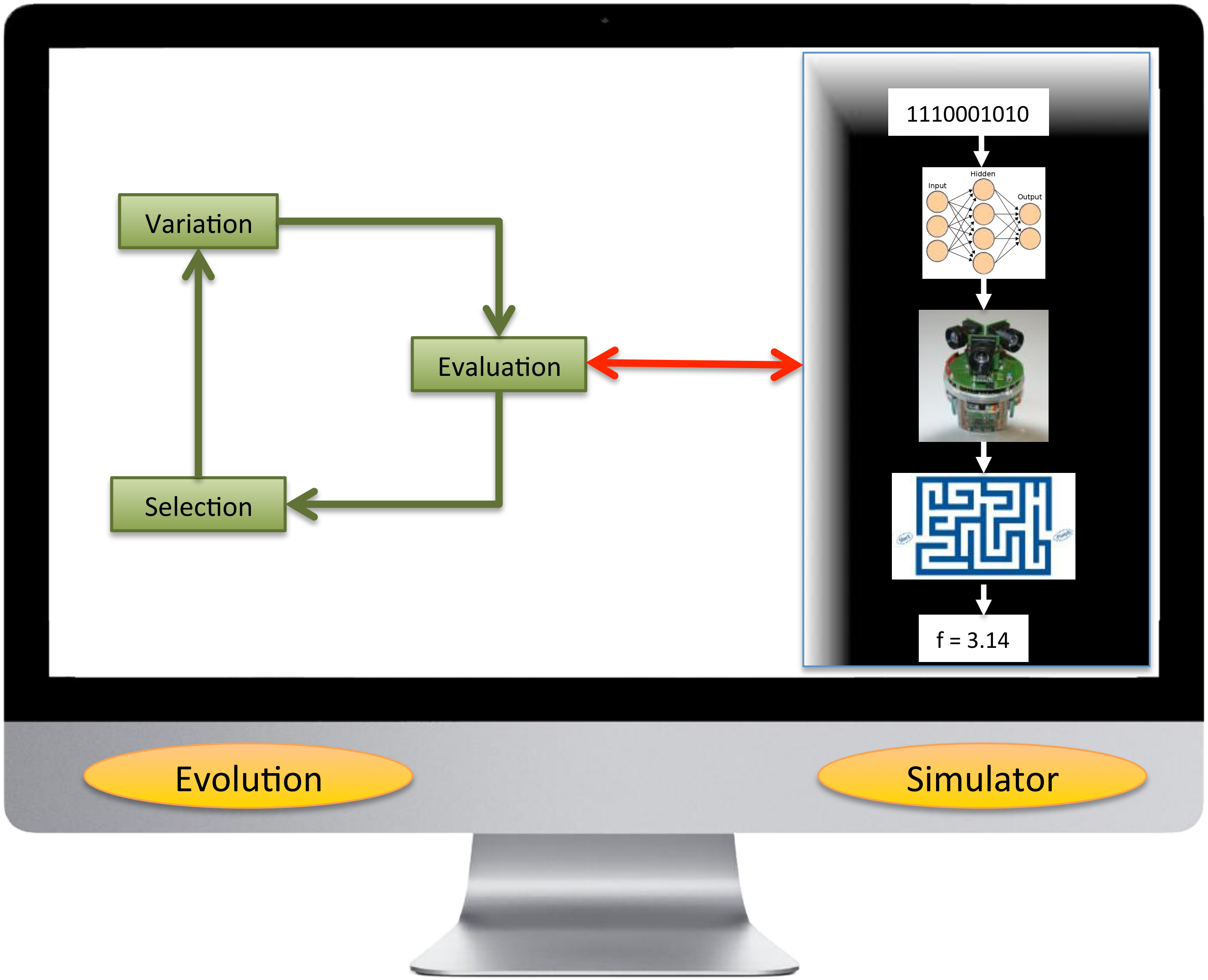
Evolution is a good solution because:

- It can solve hard problems with complex and noisy objective functions
- It can adapt on-the-fly and track a moving optimum

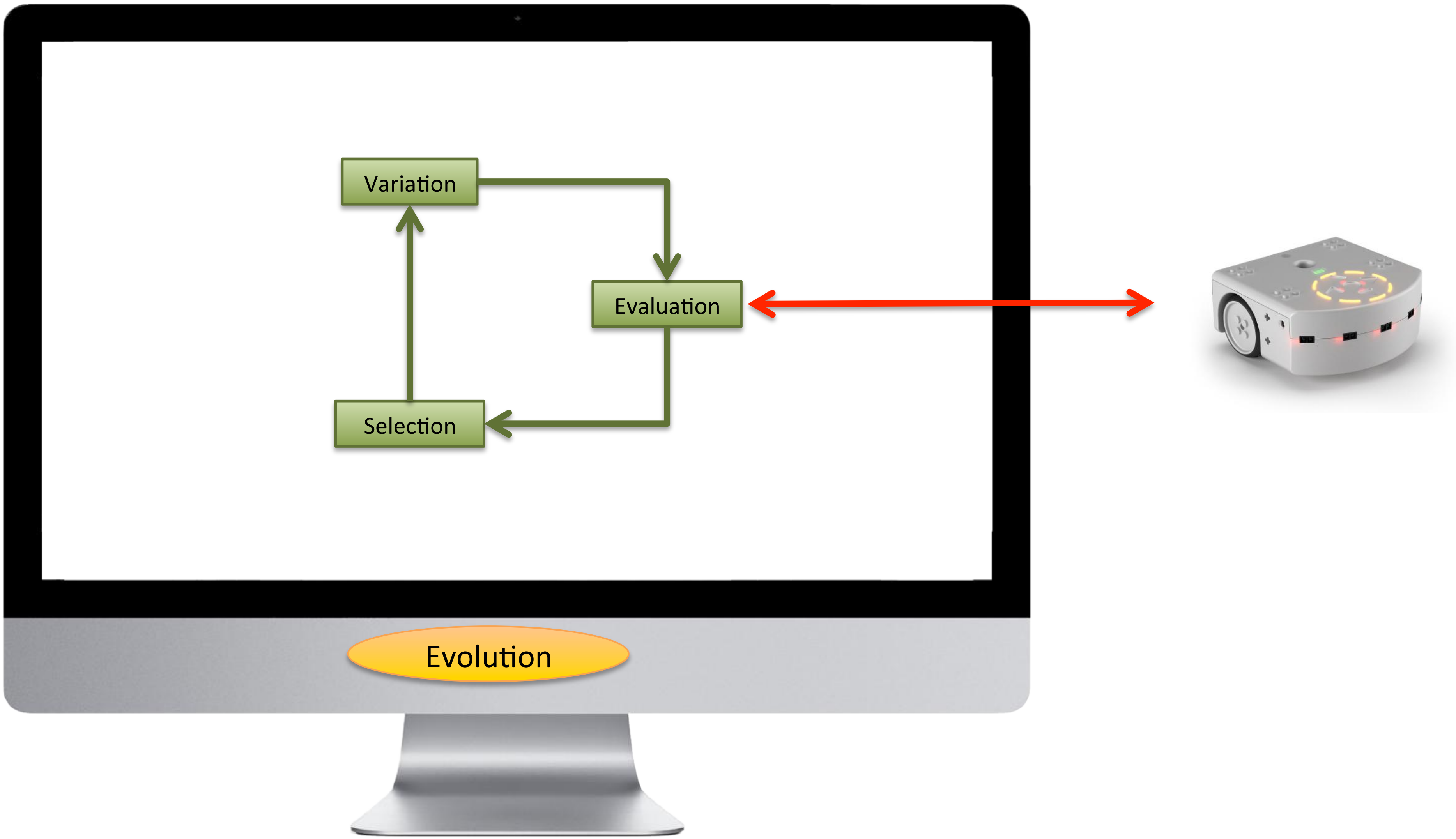
Evolution: diagram 1



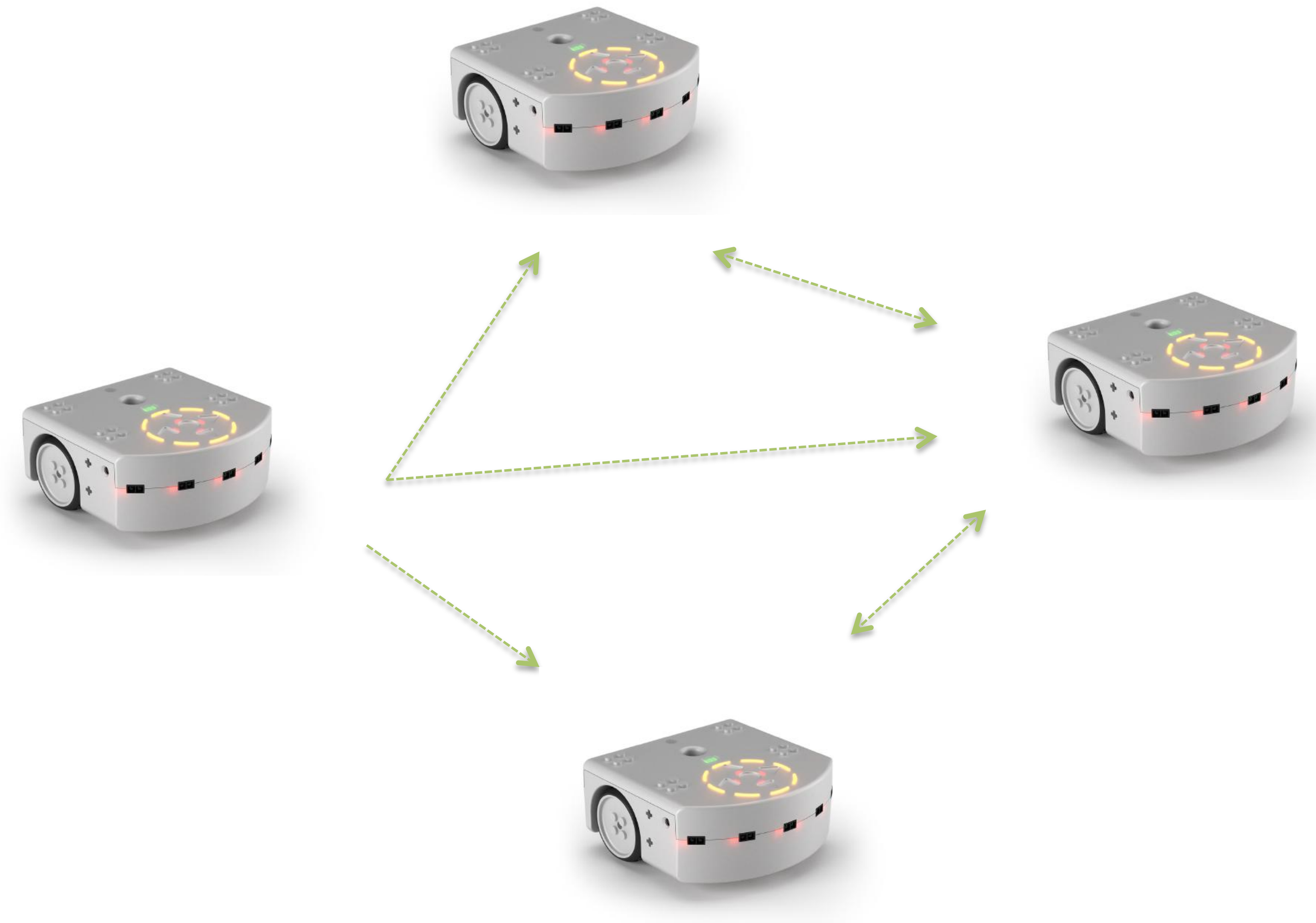
Evolution: diagram 2



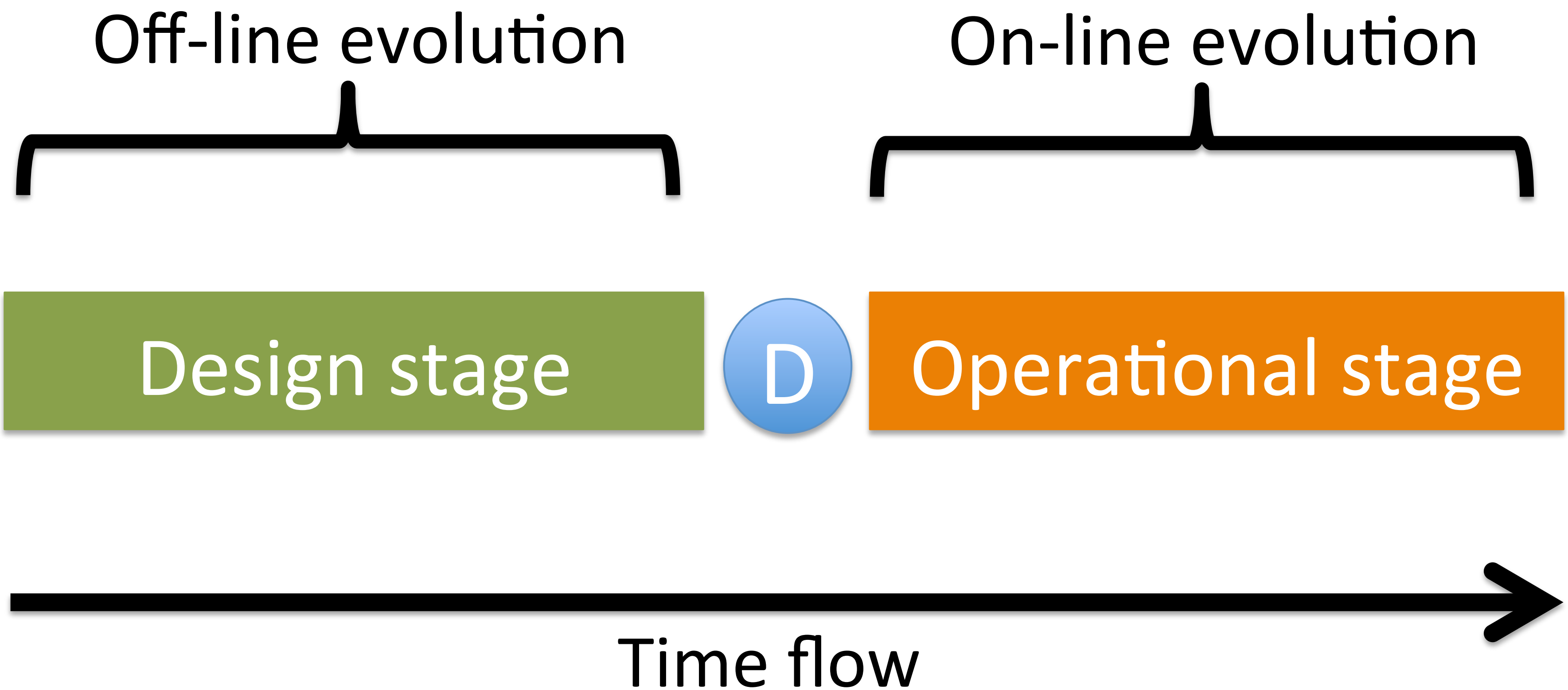
Evolution: diagram 3



Evolution: diagram 4



On-line vs Off-line evolution

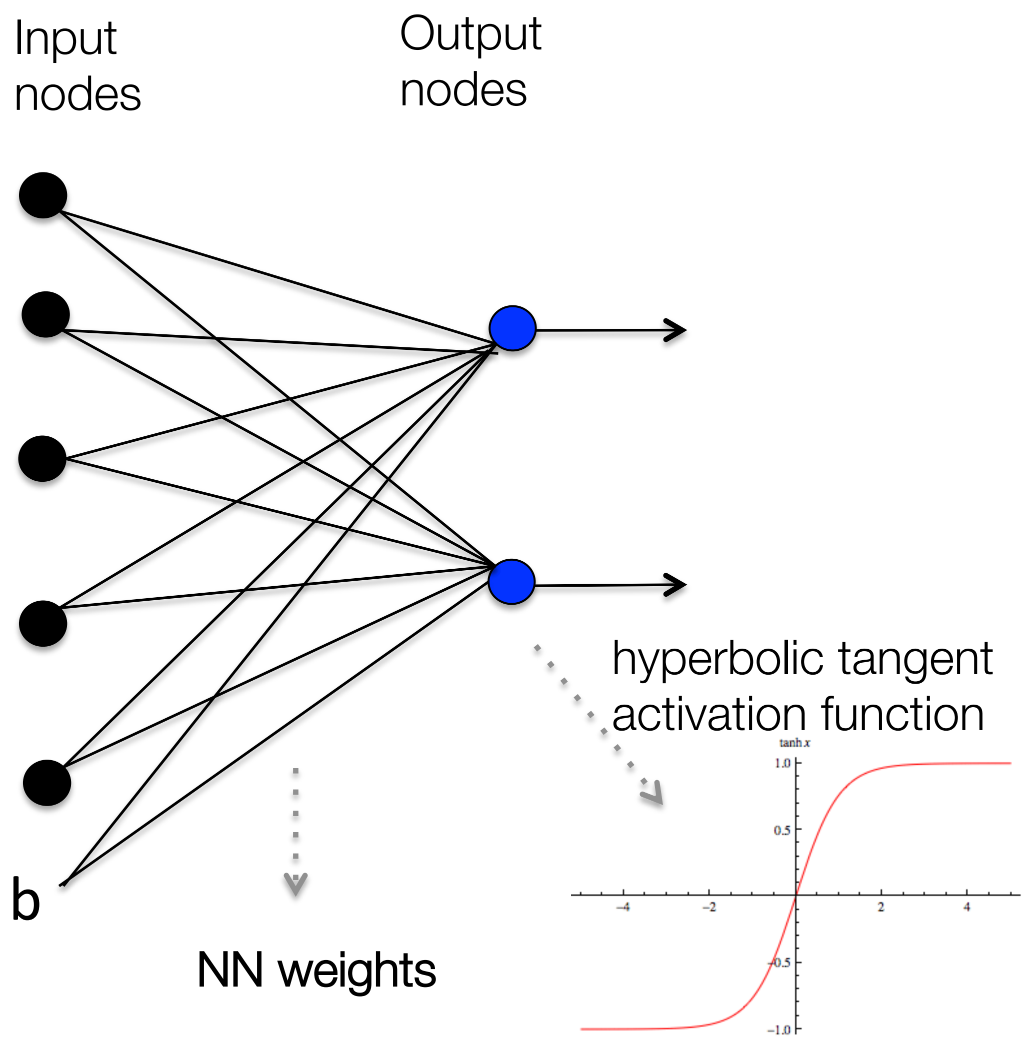


ER problems are different

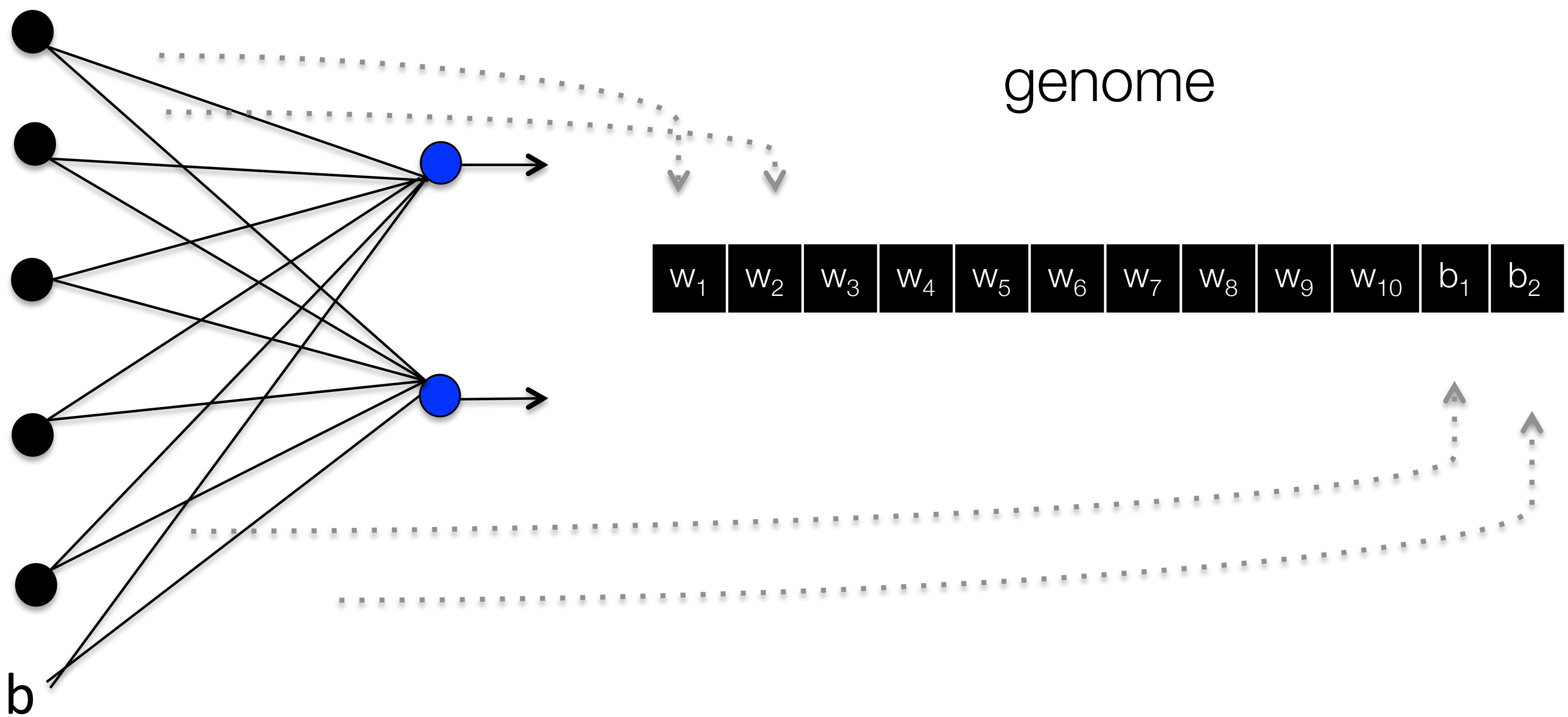
1. Fitness function is very noisy
   - In the real world "the noise is for free", many factors influence working of robots
2. Fitness function is very costly
   - For robust behaviour, several (starting) conditions must be considered
3. Fitness function is very complex
   - EAs for optimization: 3-tuple: genotype-phenotype-fitness
   - Evo Robo: 4-tuple: genotype-phenotype-**behaviour**-fitness
4. Fitness function may be implicit
   - If evolution is not to optimize anything, e.g. Artificial Life
5. Fitness landscape has "no-go areas"
   - Evaluating a bad tour for TSP can waste time, evaluating a bad controller for a robot can waste the robot

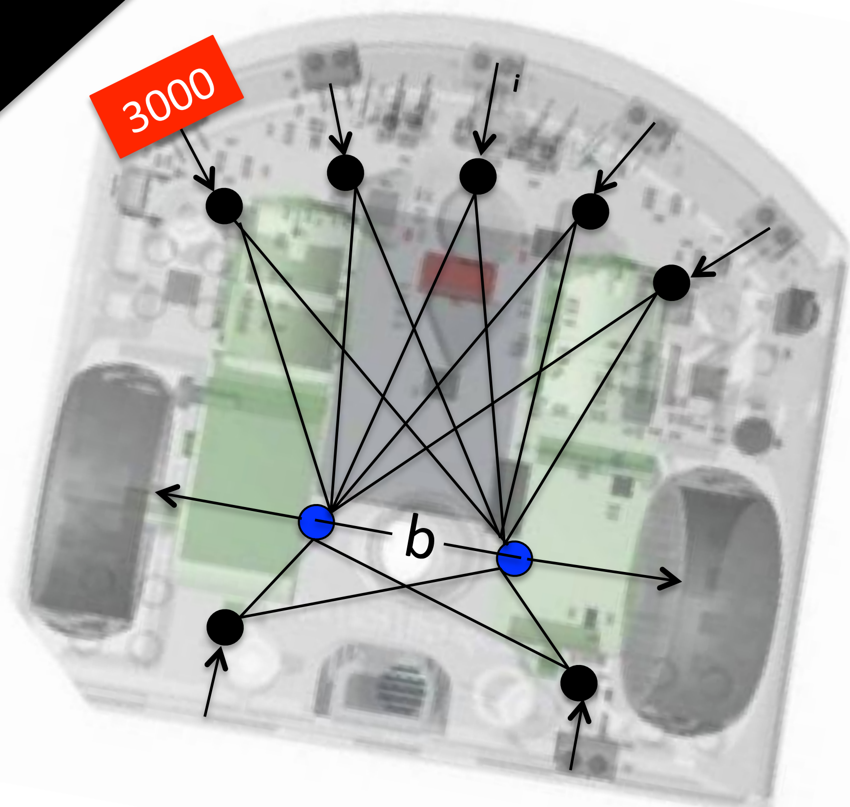
What do we evolve?

We evolve the controller of the robot which is usually a neural network



Controller respresentation





Suppose sensor 1 is activated with value 3000 and

$w_1 = 3$

$w_2 = -1$

$b_1 = b_2 = 0{,}1$

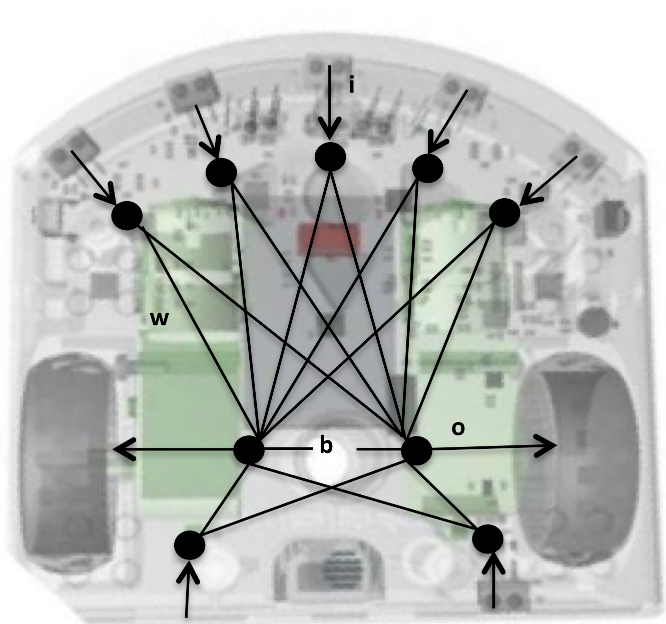
To calculate the wheel speeds do:

1. Normalise sensor input:
3000 – 0,5*4500 / 4500 = 0,1666

2. Calculate normalised output
Left = tanh(0,1666*3+0,1) = 0,54
Right = tanh(0,1666*-1 + 0,1) = -0,0666

3. Calculate wheel speeds
Left = 0,54 * 500 = 270
Right = -33,3

Evaluate the controller



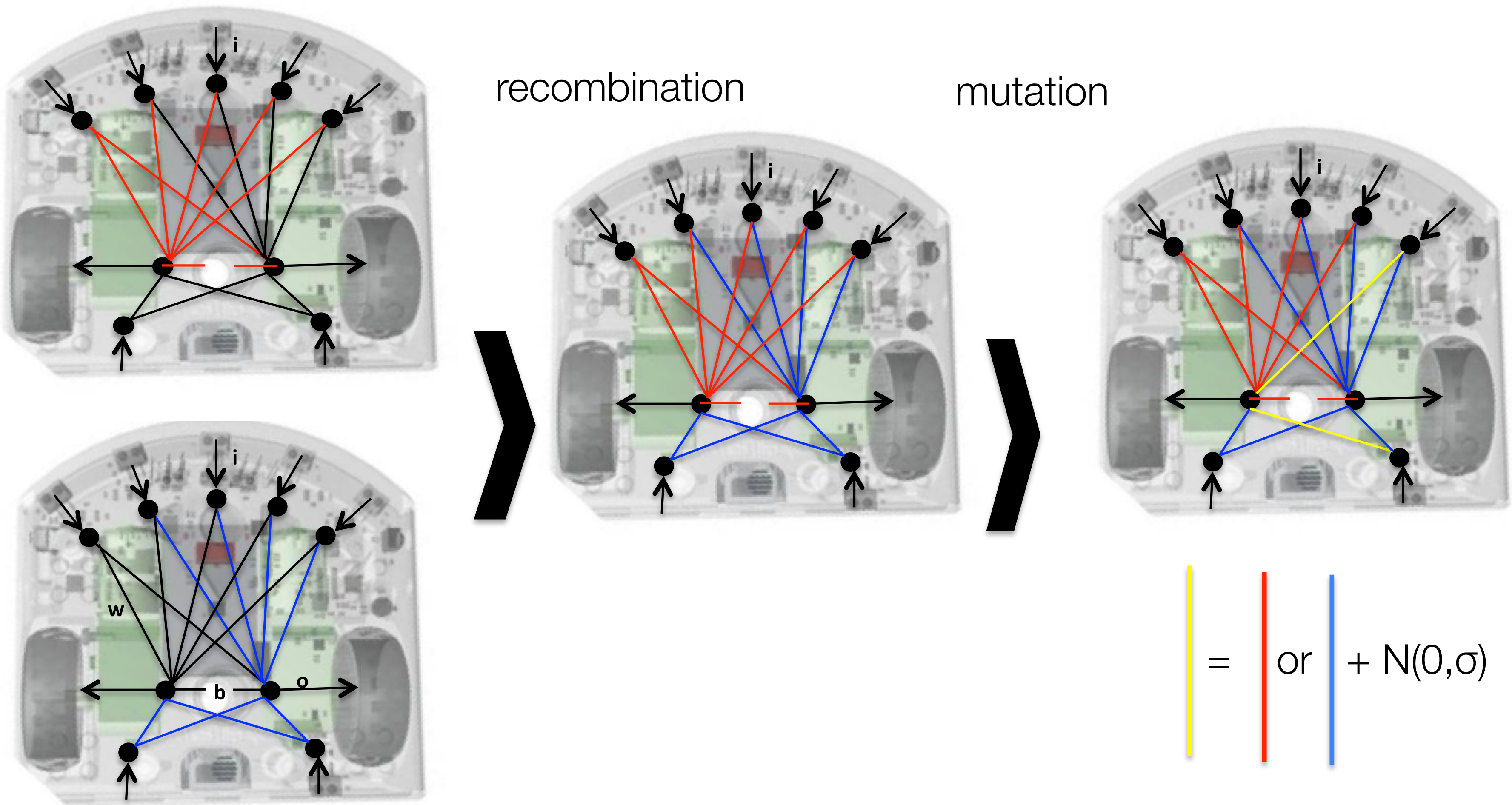
1. Perform one neural network step
2. Calculate the fitness of this step with the following formula

$$f = \sum_{t=0}^{T} \underbrace{s_{trans}}_{\text{forward}} \times \underbrace{(1 - s_{rot})}_{\text{rotational}} \times \underbrace{(1 - v_{sens})}_{\text{sensor activation}},$$

3. Repeat this a number of time steps (minimum of T=300) and you have a controller fitness

Create offspring

1. Select parents to recombine and/or mutate the genomes to create the offspring:



2. Test the controllers of the offspring and select who will form the second generation

Illustrative video

Research challenges in ER

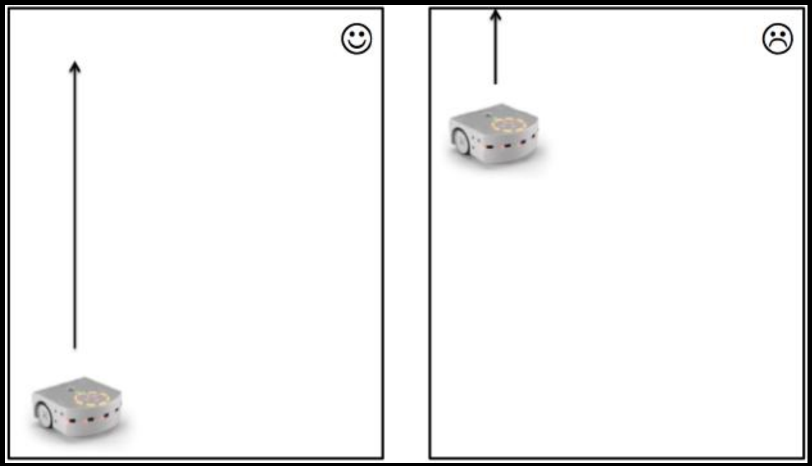
Tasks in hardware are simple

Evolution time is long

Evaluation function is noisy

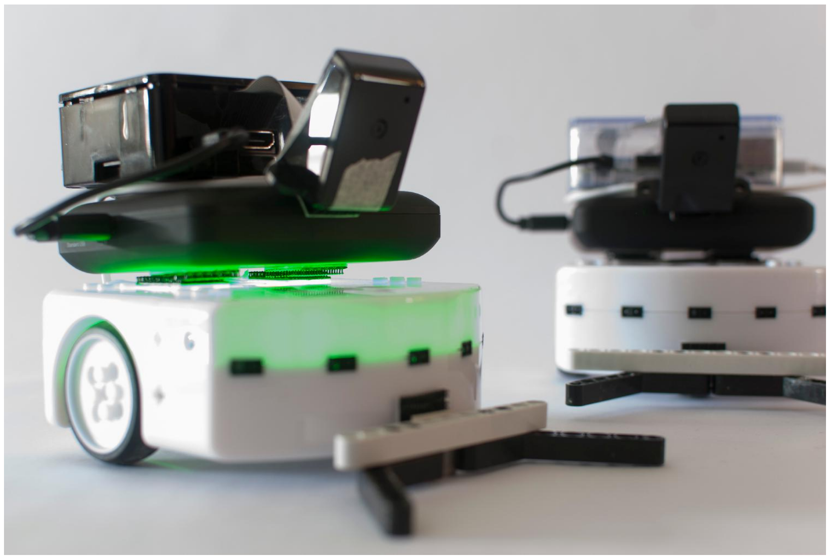
Shaping the fitness function is difficult

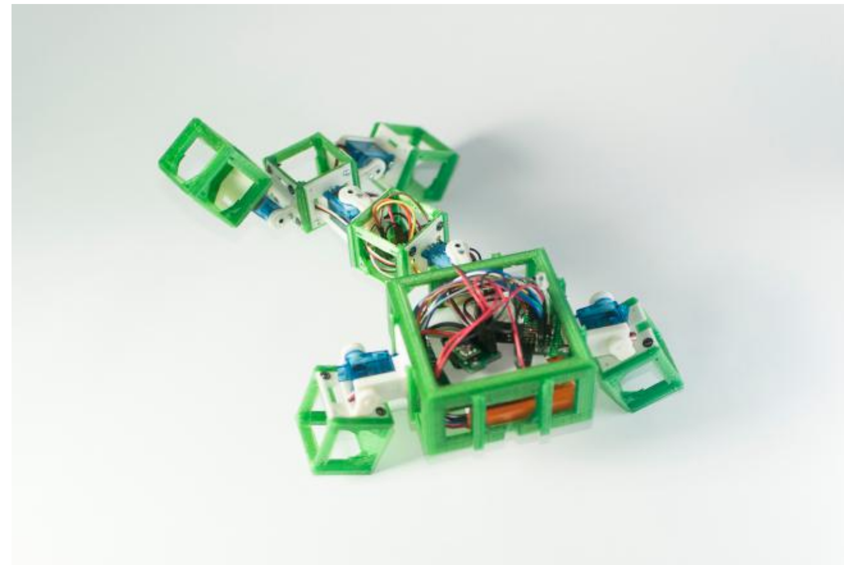
Reality gap



mind

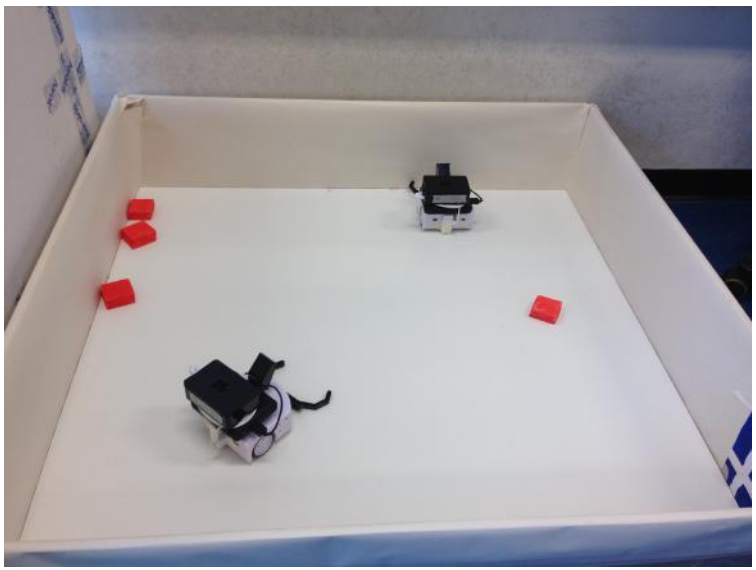
body



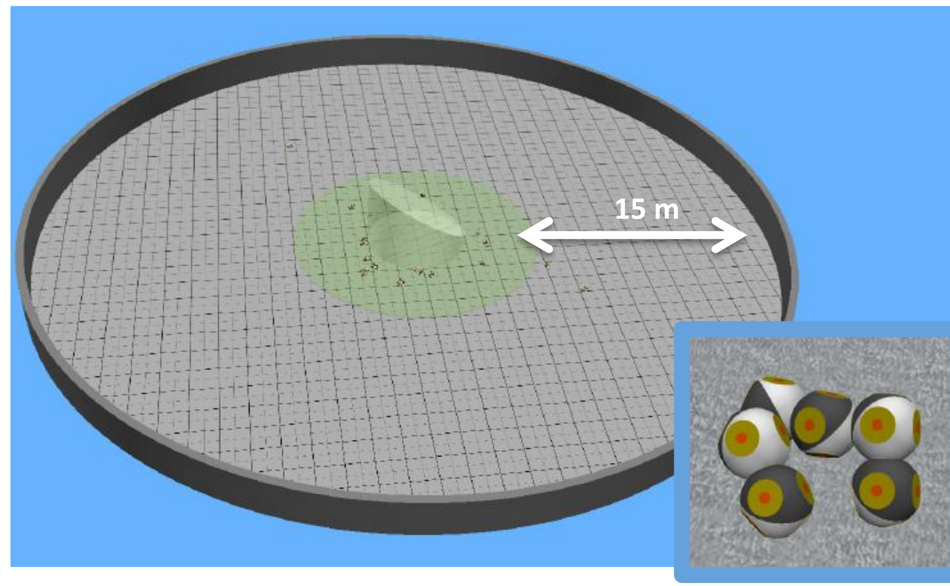


Does the mind shape the body or the other way around?
Can we find a good controller that suits every body type?

objective driven



environment driven



How complex can a system become when the fitness is only environmental?

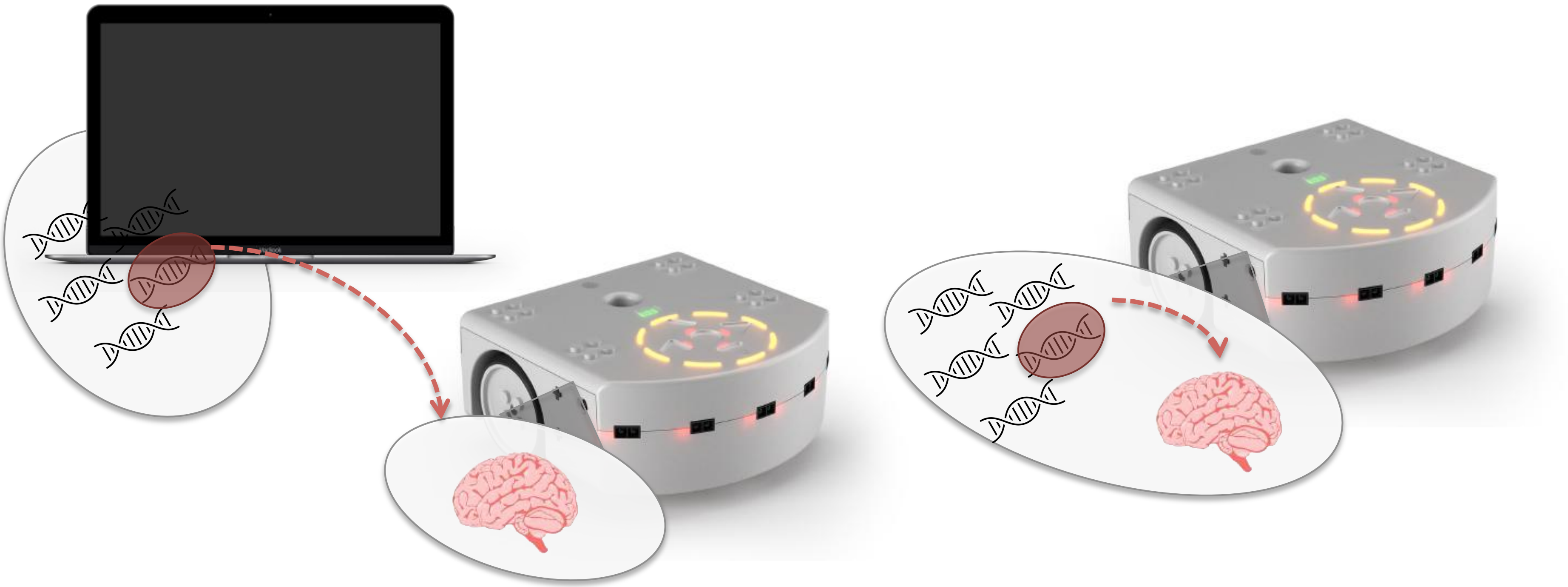
simulated

physical

Can we develop a method to reduce the reality gap?

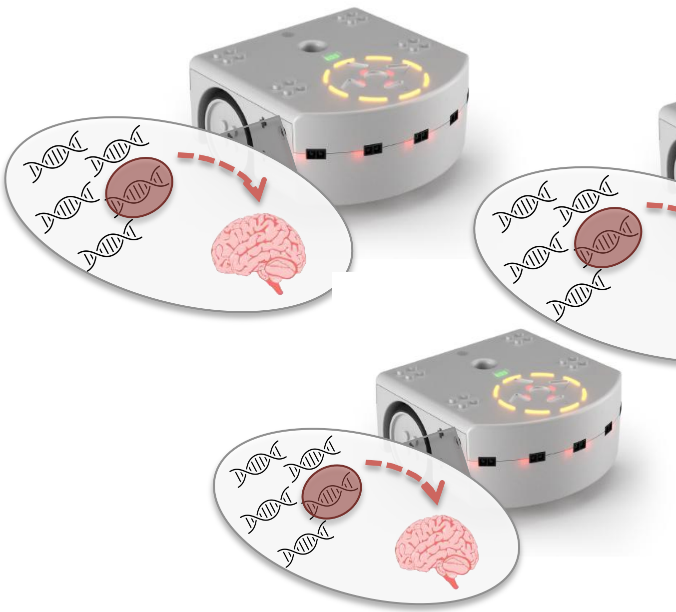
offline evolution

online/embodied evolution

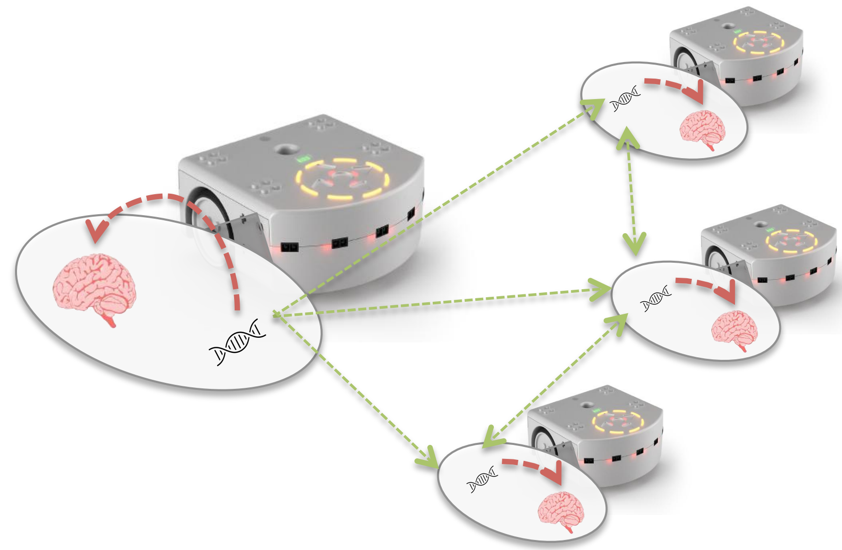


How can the robot perform the desired task well while still trying to improve?
How can a robot quickly adapt to environmental changes?

encapsulated/ individual



distributed/ social



Can a distributed setup result in a performance increase / speed increase / more robust controllers?

Part II

How to start your own experiments with Thymio II robots

Unpack the Thymio to full equipment

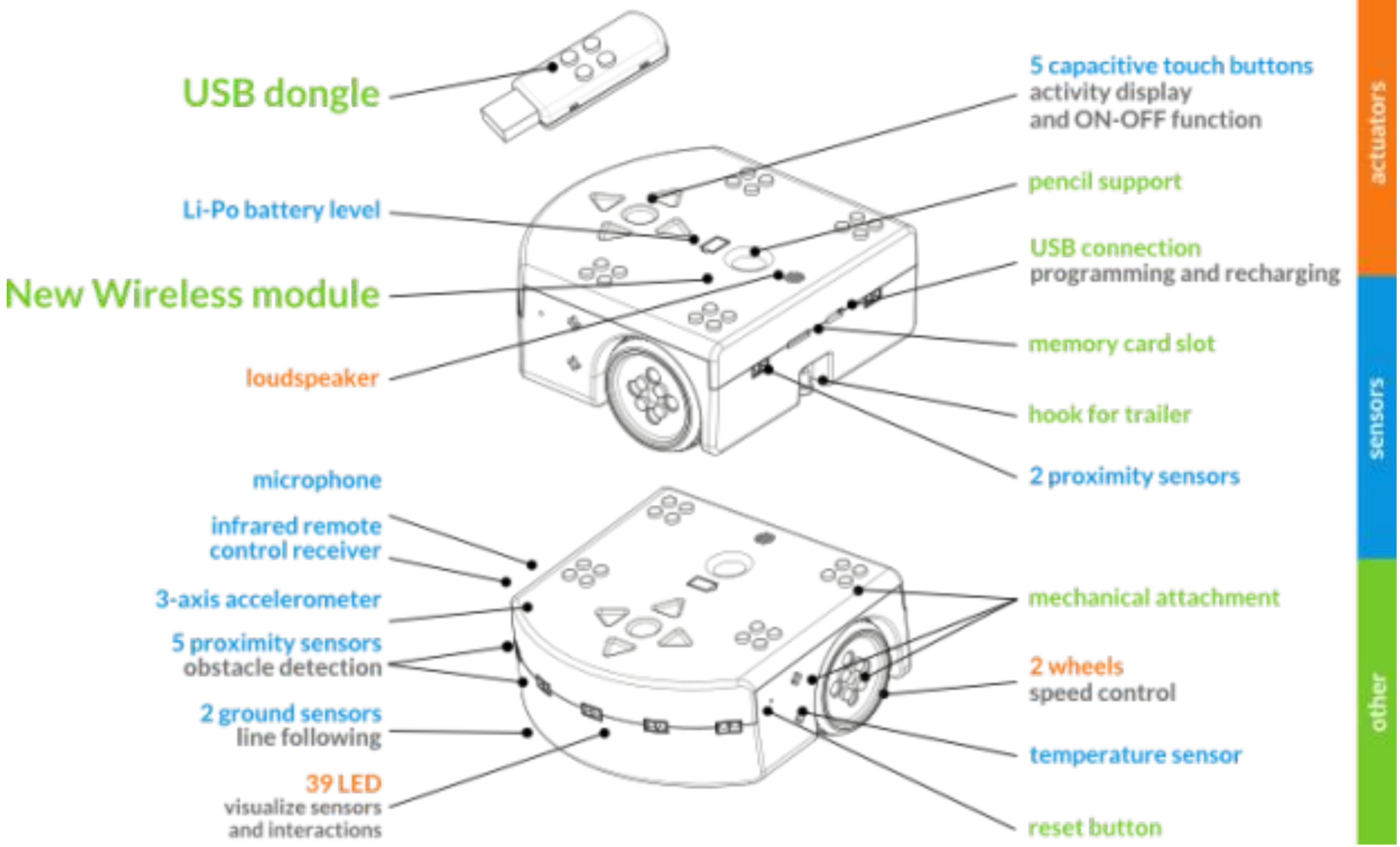
Workflow with do’s and don’ts

Set up experiment with easily changeable tasks and controllers

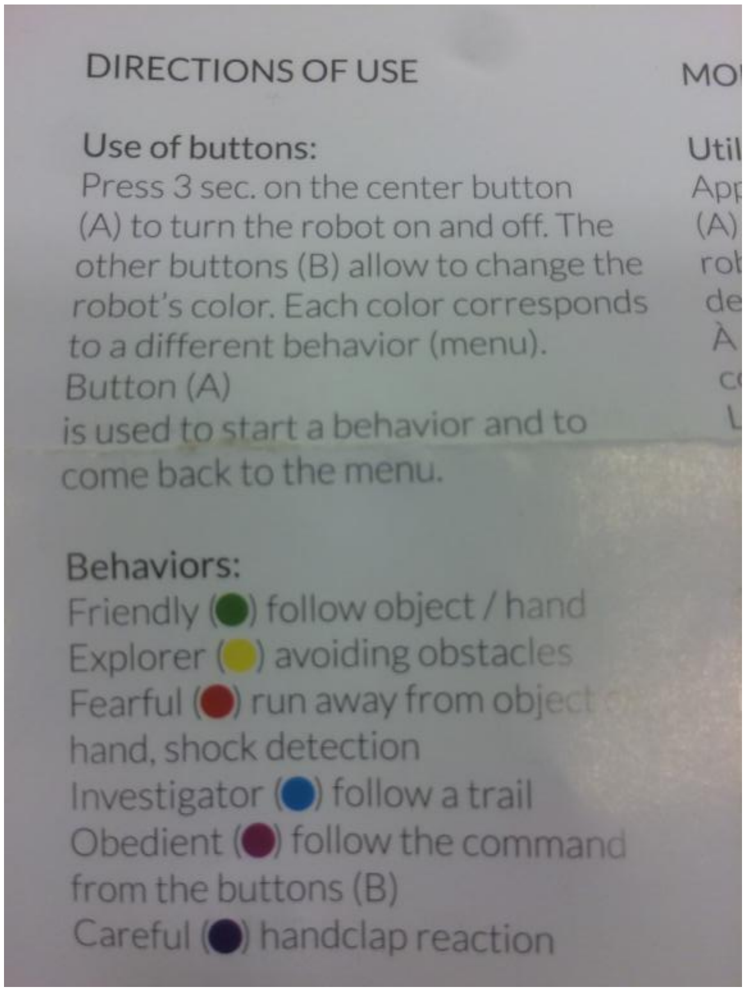
Thymio II robot

This project aims to let a wide public – children in particular – discover and learn computational thinking, robotics, engineering and digital technologies.

Thymio robot: sensors and actuators



Thymio robot: built-in behaviours

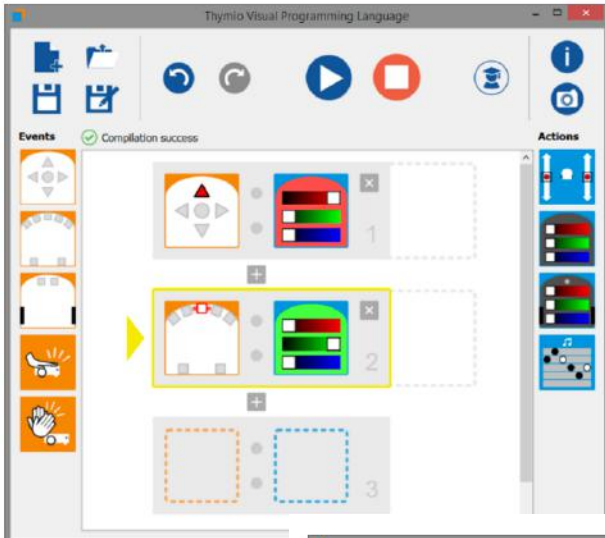


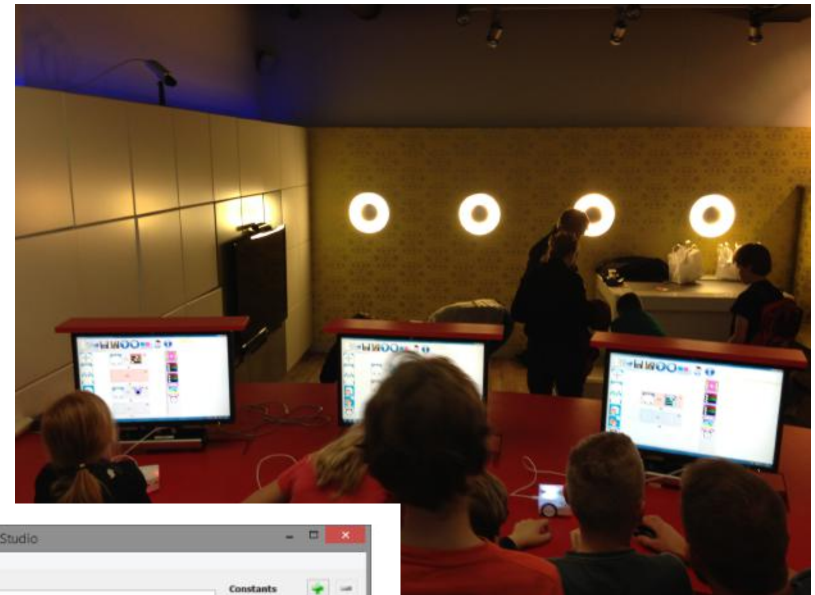
DIRECTIONS OF USE

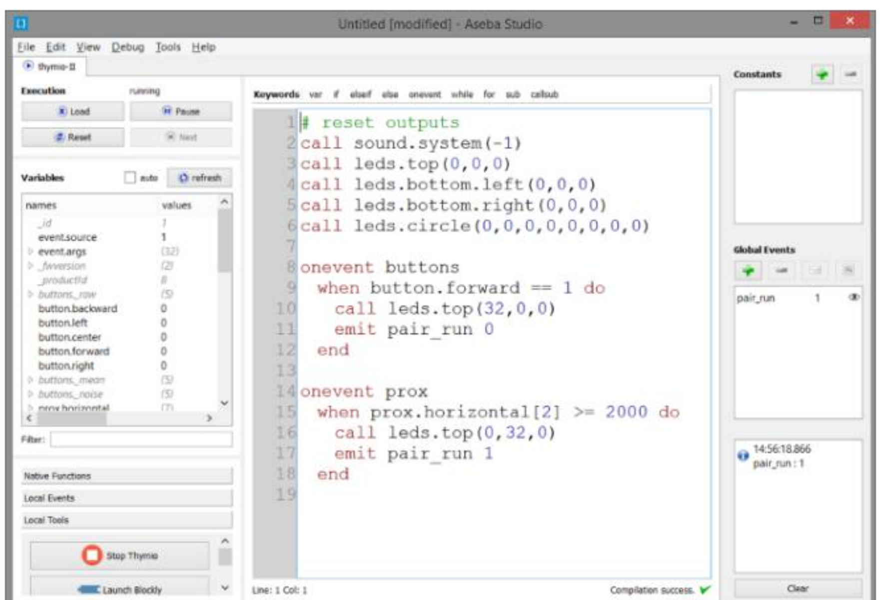
Use of buttons:
Press 3 sec. on the center button (A) to turn the robot on and off. The other buttons (B) allow to change the robot's color. Each color corresponds to a different behavior (menu).
Button (A)
is used to start a behavior and to come back to the menu.

Behaviors:
Friendly (●) follow object / hand
Explorer (●) avoiding obstacles
Fearful (●) run away from object or hand, shock detection
Investigator (●) follow a trail
Obedient (●) follow the command from the buttons (B)
Careful (●) handclap reaction

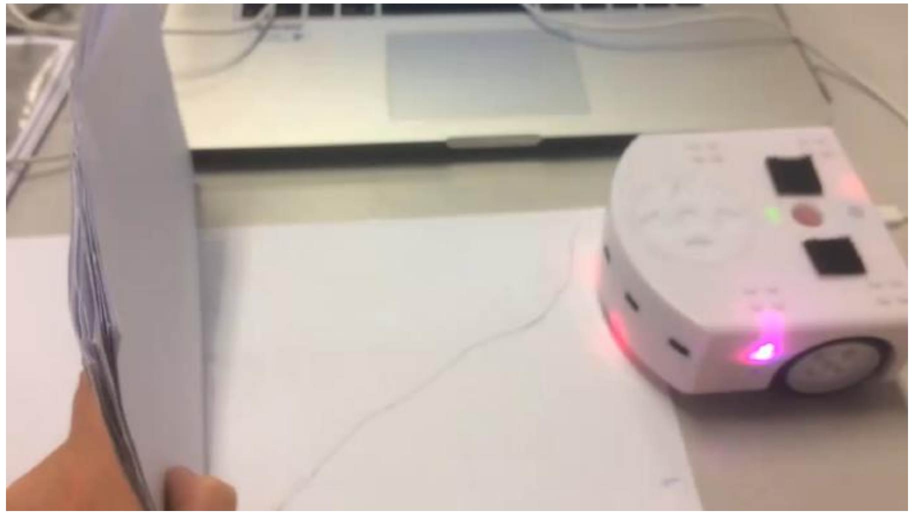
Program own behaviours

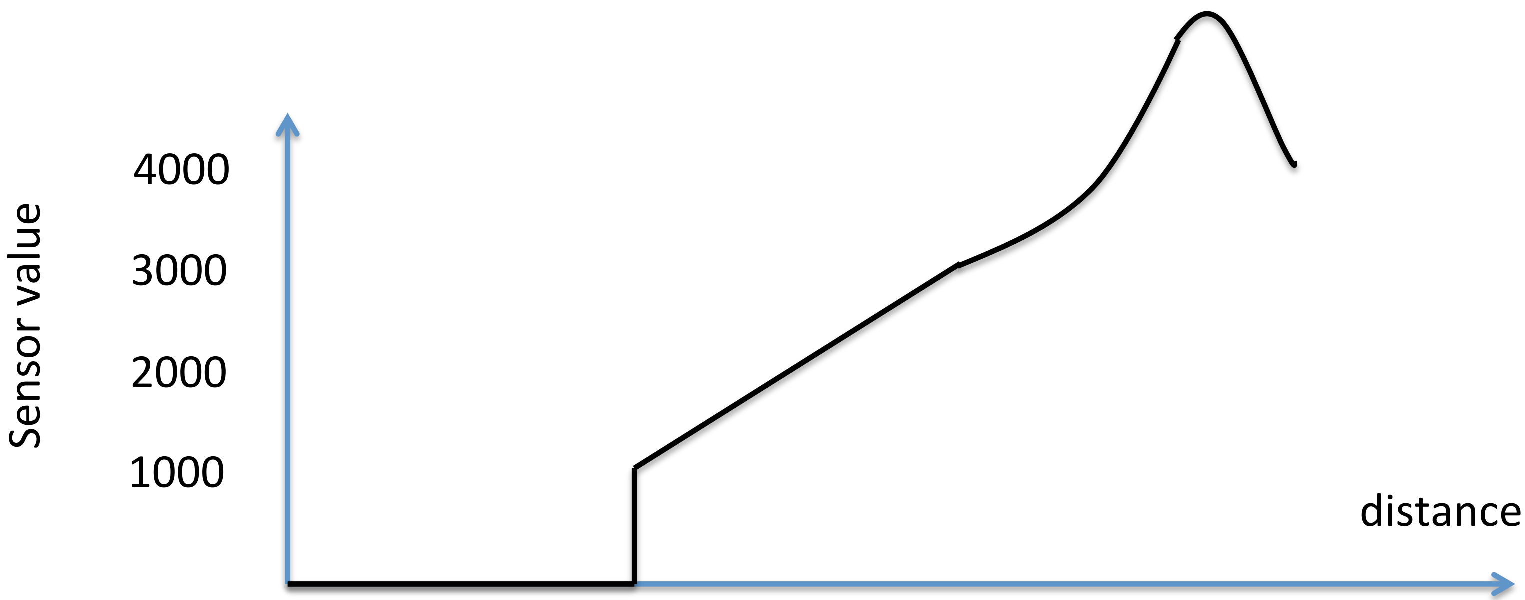






Thymio robot: sensor readings





And be creative

And be creative

But if you want to do research with the Thymio robots …

You need the following hardware:

Some Thymio robots
(€130 pp)

Raspberry Pi 2
(€45) (or Pi3)

Pi camera
(€25)

WiFi
dongle
(€10)

Micro SD
(€10)

External
battery
(€50)

2x usb-micro
usb

Router
(€40)

Total costs:
€270 per robot
(e-puck no extensions
= €780)

First steps: Assemble hardware

Next steps (I/II):

Raspian is a possible operating system for raspberry pi

← Aseba →

Set of tools to allow you to program the robot

Install necessary packages on PI including openCV, piCamera etc through the command line

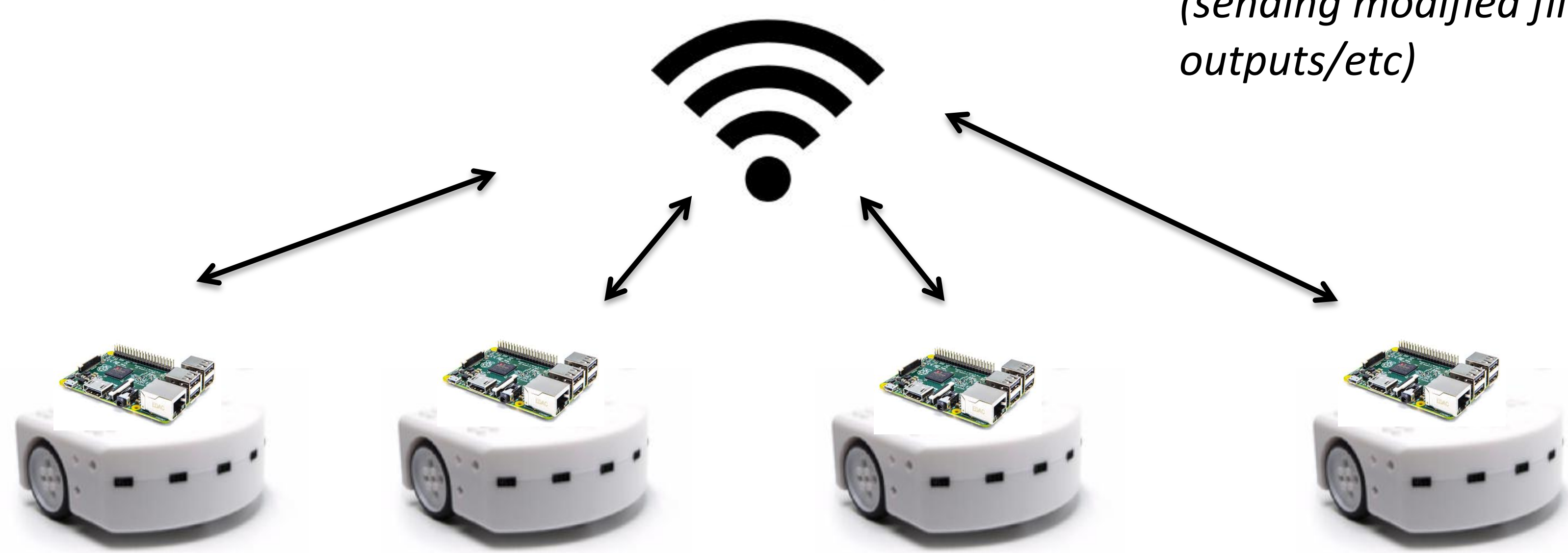
Modify the WiFi settings to the robot network by modifying two files

Next steps (II/II):

GitHub

Get our algorithms online at https://github.com/ci-group/NEATThymio

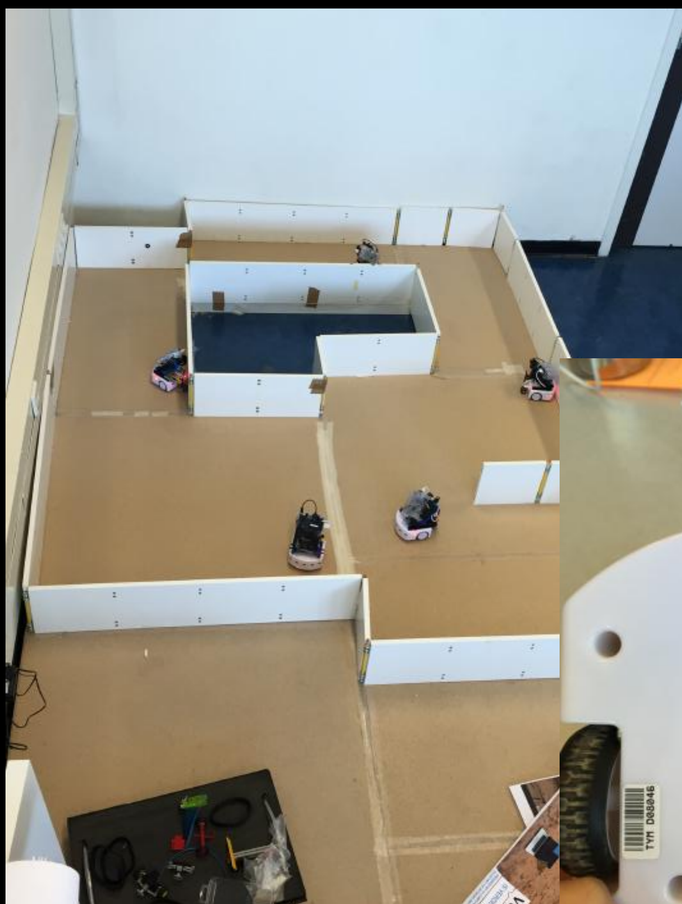
Communicate with the robot (sending modified files/receiving outputs/etc)



Before starting: do’s and don’ts

☹ Floor with to much friction

☺ Smooth floor



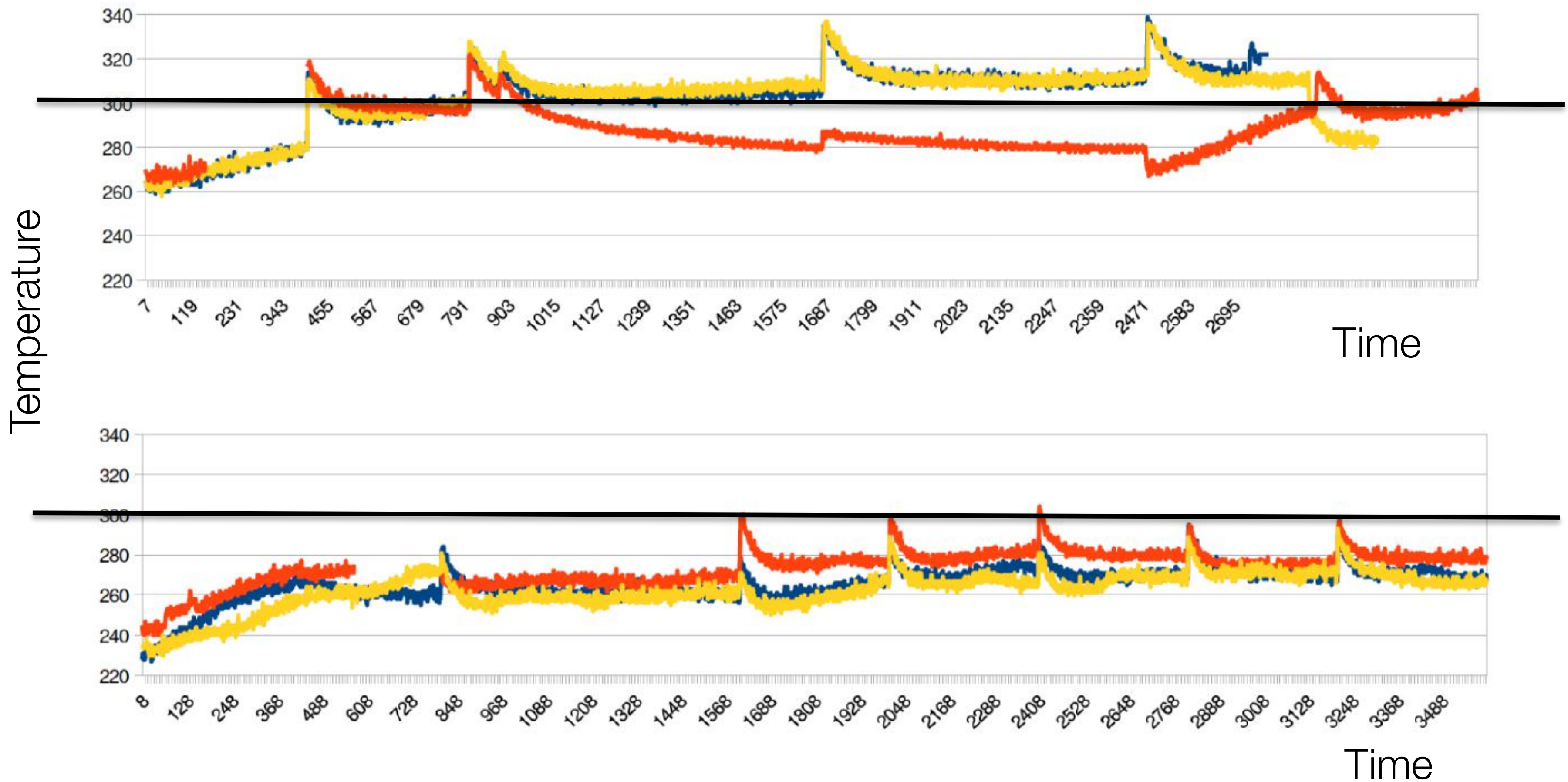
Before starting: do’s and don’ts

☹ Weight not equally distributed

☺ Hook to hook or cover

Before starting: do’s and don’ts

☺ Have a cold room (16 degrees or colder) and set maximum speed to 60% (no camera) and 30% (camera) to process the images.



Before starting: do's and don'ts

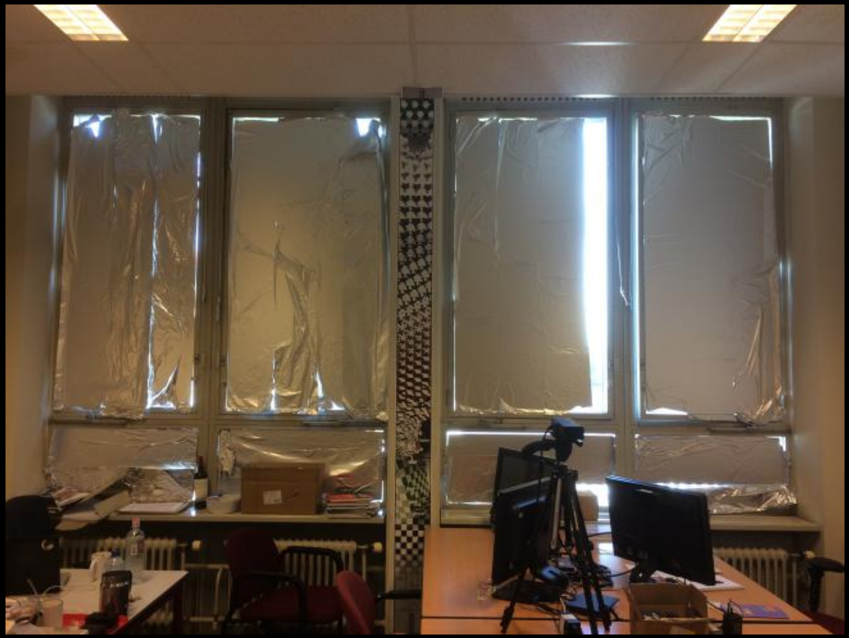
☺ Decide if and when to intervene upfront

Before starting: do’s and don’ts

☺ Check if the robot is still driving straight

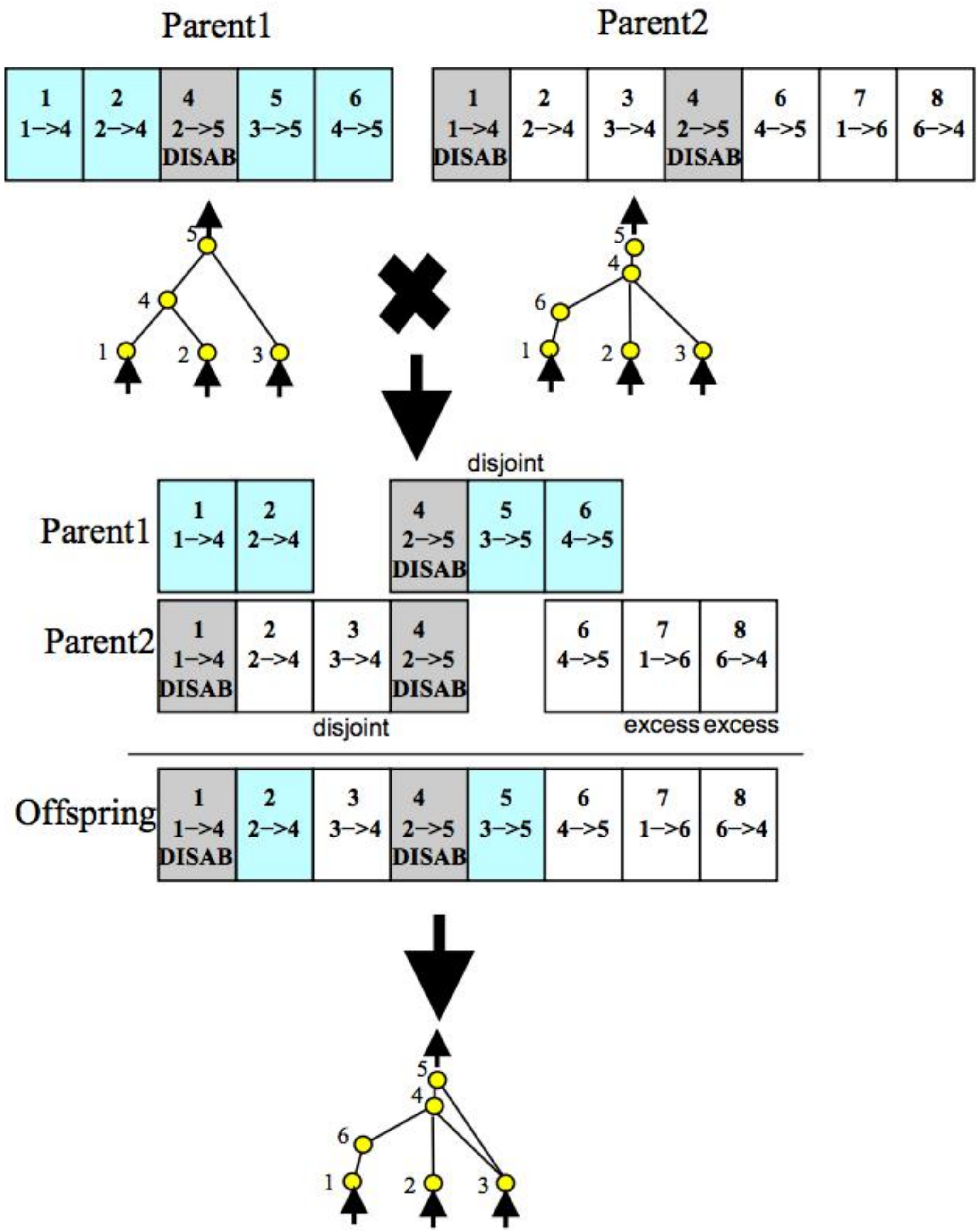
Before starting: do’s and don’ts

☺ Daylight heavily effects the Raspberry Pi camera input. Options: dark room / big led lights / black cover over arena with lamp (tracking required then).



Current controllers and tasks

Controllers	Learning Algorithm	Tasks
Neural Network	NEAT	Obstacle Avoidance
	distributed NEAT	Foraging



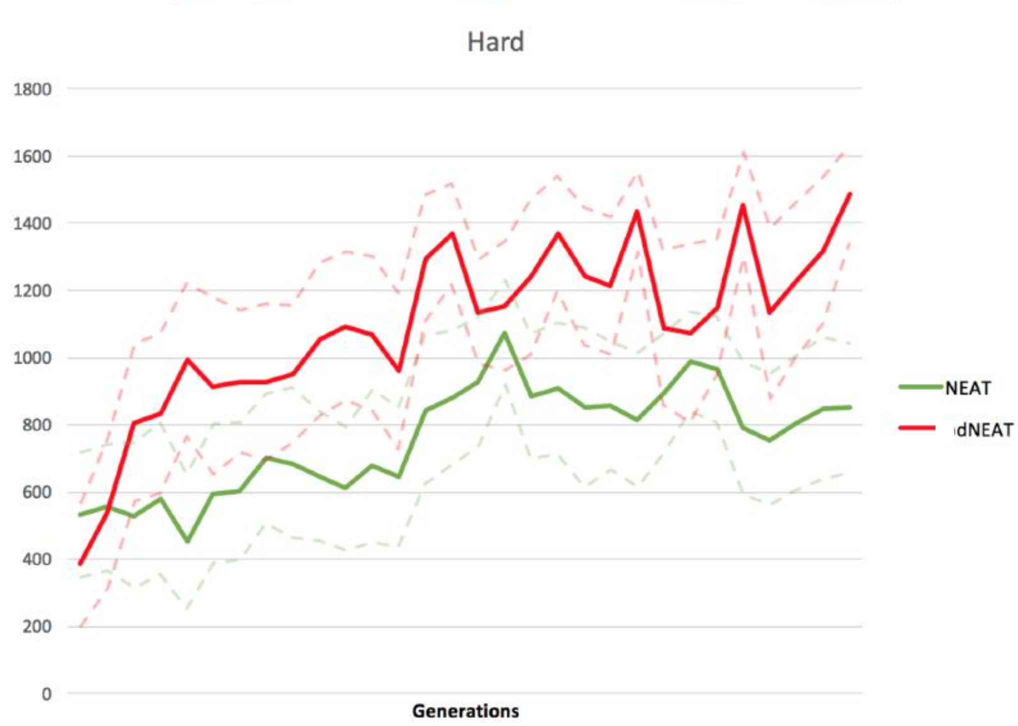
The population consist of species with similar network typology.

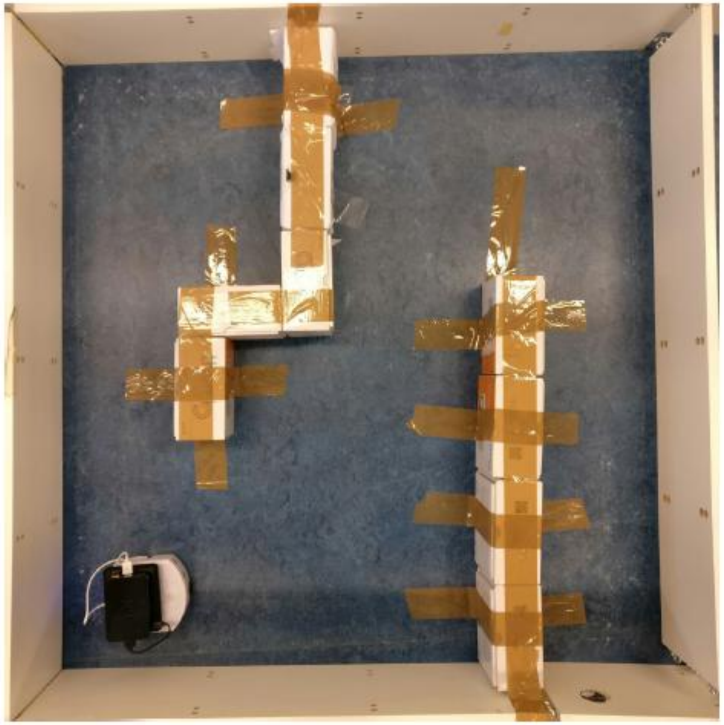
Distributed NEAT:

- Requires clocks instead of innovation numbers
- Sends the best controller to be accepted as a species

Current Research

Is there an increase in learning speed and/or performance when robots can share their best controller?





First results show a higher end performance and a faster learning curve.

Questions?

Part III

Stay and start experimenting with the Thymio robot yourself for your level of experience

1st option:

start obstacle avoidance and change the fitness function

2nd option:

start foraging task and stream the Thymio camera live

3th option:

join forces and create a cooperating task with multiple robots

Start Experiments

1. Connect to the conference WiFi
2. Open a terminal
3. git clone https://github.com/ci-group/NEATThymio.git
   (for mac, install xcode-select --install (error: xcrun error))
4. Connect to ThymioNet (password: 172luckytulip75B)
5. nano NEATThymio/src/bots.txt and put you robots IP address in this file
6. Go to src folder (cd NEATThymio/src)
7. sh ./sync_all.sh (to send your files to the robot), type “yes” and use the password raspberry (multiple times)
8. Open a new terminal
9. Login to the robot: ssh pi@192.168.1.y (where y= pi address) and use the password raspberry
10. Start an experiment with the command:
    sh ./start_one.sh obstacle_avoidance.py 192.168.1.y GO
11. To stop the robot use the other terminal and type ./stop_all.sh

For passwordless access to the raspberry do:

```
ssh-keygen -t rsa
ssh-copy-id pi@192.168.1.y
```

References

- J. Bongard, Evolutionary robotics. Commun. ACM 56, 74–85, 2013.
- Bredeche, N., Haasdijk, E., & Eiben, A. E. (2009, October). On-line, on-board evolution of robot controllers. In International Conference on Artificial Evolution (Evolution Artificielle) (pp. 110-121). Springer Berlin Heidelberg.
- Bredeche, N., Montanier, J. M., Liu, W., & Winfield, A. F. (2012). Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. Mathematical and Computer Modelling of Dynamical Systems, 18(1), 101-129.
- D'Angelo, M., Weel, B., & Eiben, A. E. (2013, December). Online gait learning for modular robots with arbitrary shapes and sizes. In International Conference on Theory and Practice of Natural Computing (pp. 45-56). Springer Berlin Heidelberg.
- S. Doncieux and N. Bredeche and J.-B. Mouret and A.E. Eiben, Evolutionary robotics: what, why, and where to, Frontiers in Robotics and AI, 2(4), doi:10.3389/frobt.2015.00004, 2015
- A.E. Eiben, Grand challenges for evolutionary robotics, Frontiers in Robotics and AI , 1(4), doi: 10.3389/frobt. 2014
- Koos, S., Mouret, J. B., & Doncieux, S. (2013). The transferability approach: Crossing the reality gap in evolutionary robotics. IEEE Transactions on Evolutionary Computation, 17(1), 122-145.
- Pfeifer, R., & Bongard, J. (2006). How the body shapes the way we think: a new view of intelligence. MIT press.
- Silva, F., Correia, L., & Christensen, A. L. (2015, September). A case study on the scalability of online evolution of robotic controllers. In Portuguese Conference on Artificial Intelligence (pp. 189-200). Springer International Publishing.