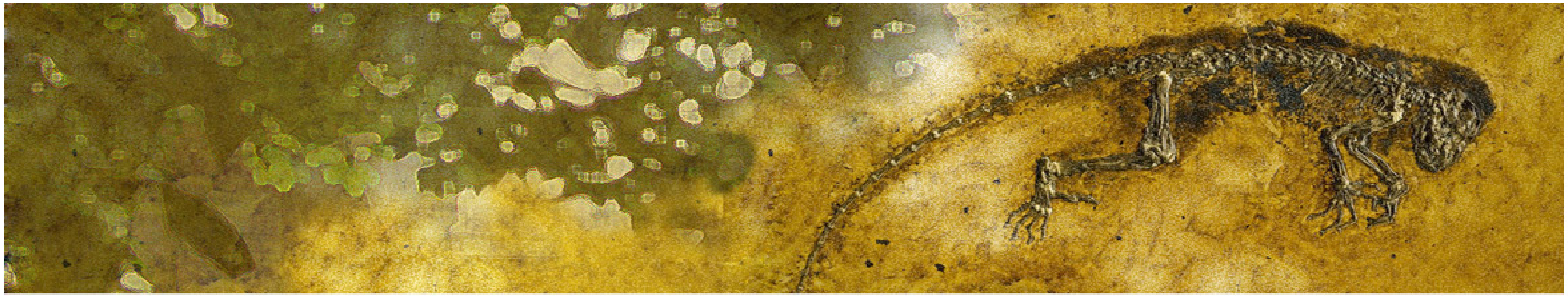


UiO : University of Oslo

INF3480

Evolutionary robotics

Kyrre Glette

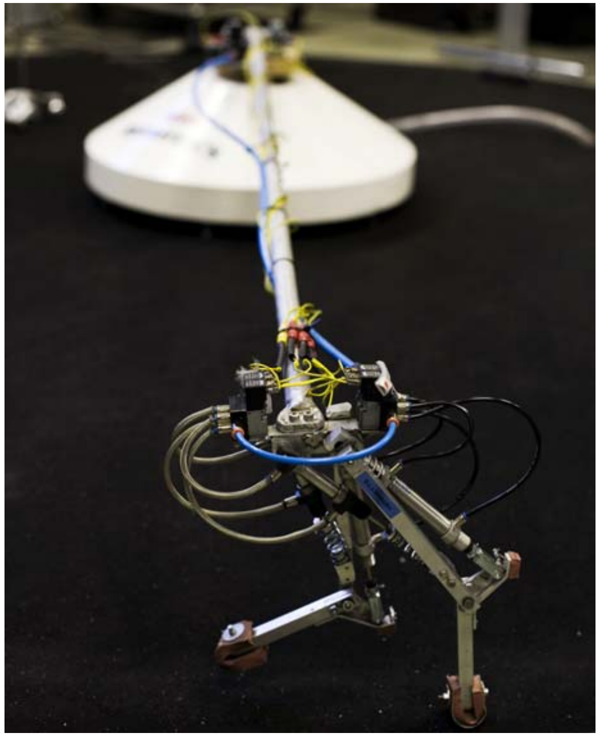
Today: Evolutionary robotics

- Why evolutionary robotics
- Basics of evolutionary optimization
  - INF3490 will discuss algorithms in detail
- Illustrating examples
  - ROBIN in-house robotic platforms
- Research challenges
  - Reality gap

Machine intelligence in robotics

- Sensing, vision
  - Gather information about the world and represent it internally for further processing
- Control and planning
  - Low-level control
  - Path planning (arms and mobile robots)
  - Task planning
- Design
  - Robot body shape / structure

Example: Henriette



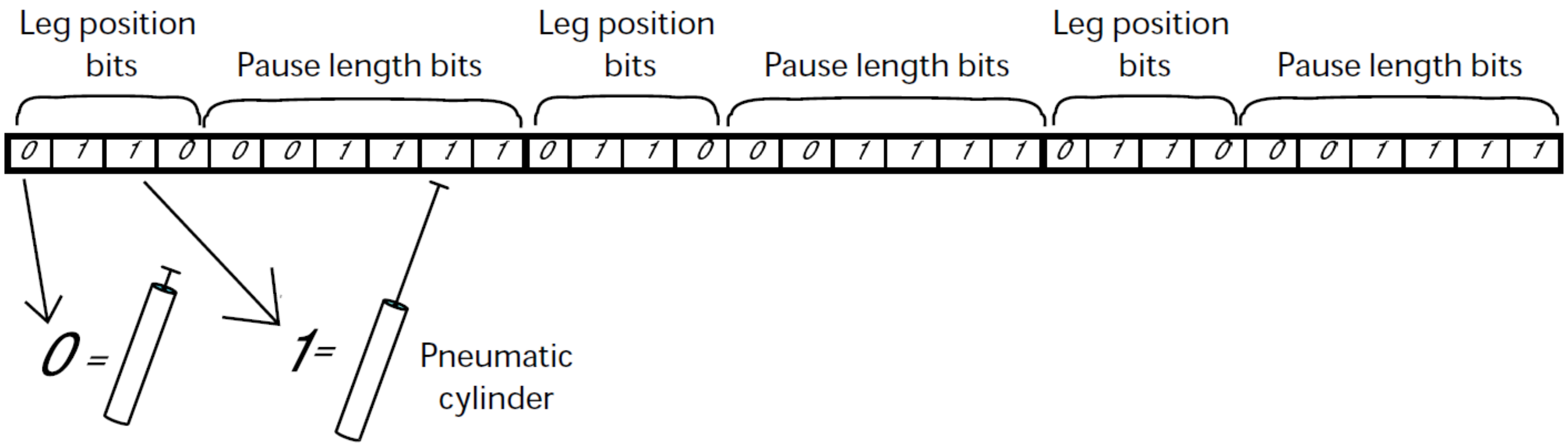
http://www.youtube.com/watch?v=mXpz5khMY2c

Why evolutionary robotics?

- Adaptation to changes in environment or robot
  - Robot may break or deteriorate
  - Environment may change unexpectedly
- Optimizing for efficiency
  - Energy, speed weight, actuators
- Unconventional, complex designs
  - New materials and actuators make it more challenging with conventional design approaches

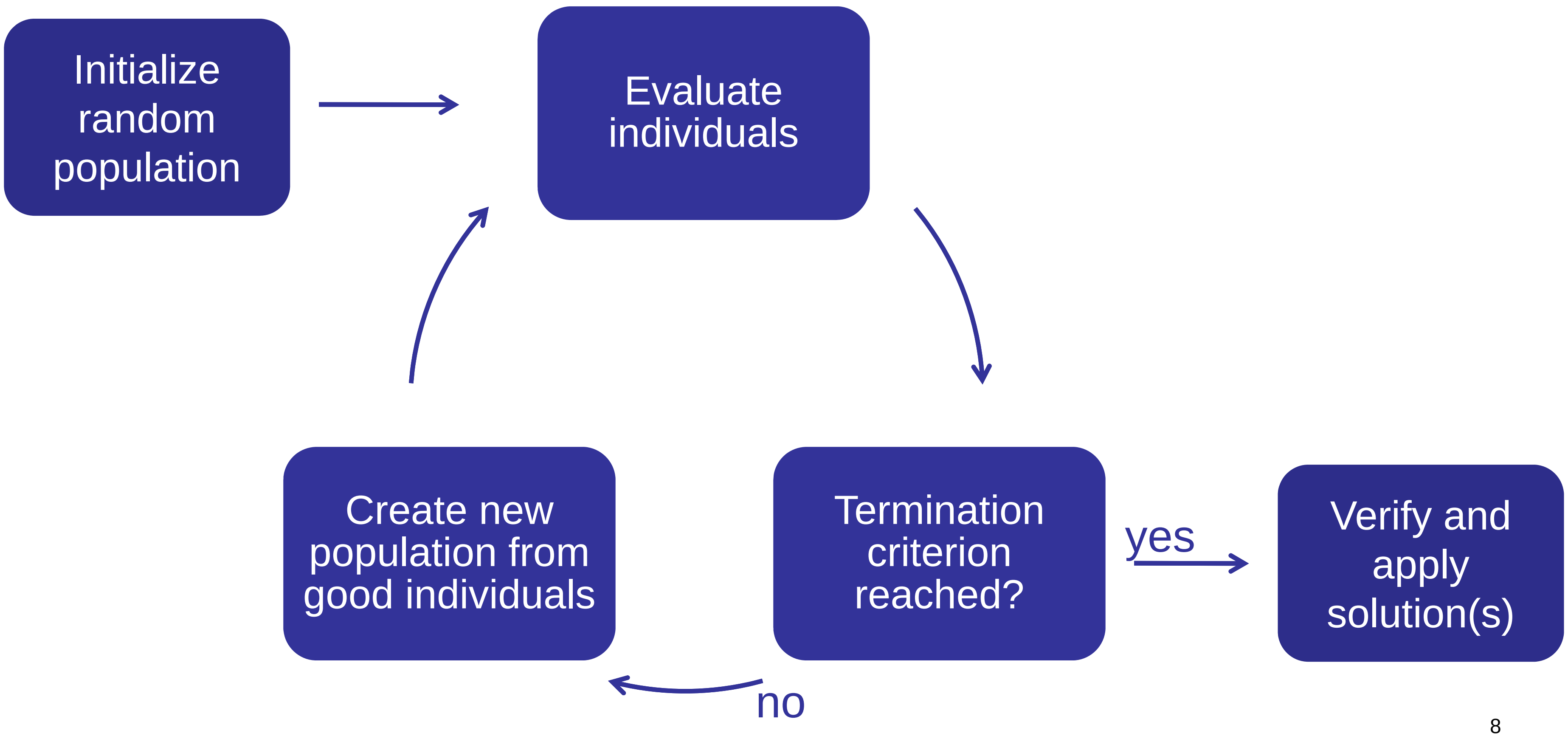
Adaptation, optimization, exploration

Henriette: Parameterized control



- Walking pattern coded into bit strings.
- 3 “states” consisting of leg configuration and pause length
- An evolutionary algorithm was used to evolve the leg configurations and the pause length.
- For each leg configuration, 4 bits denote the position of 4 actuators, 6 bits denote the length of the pause.
- Total bit string / *genome* length: 30 bits

Evolutionary Algorithm (EA)



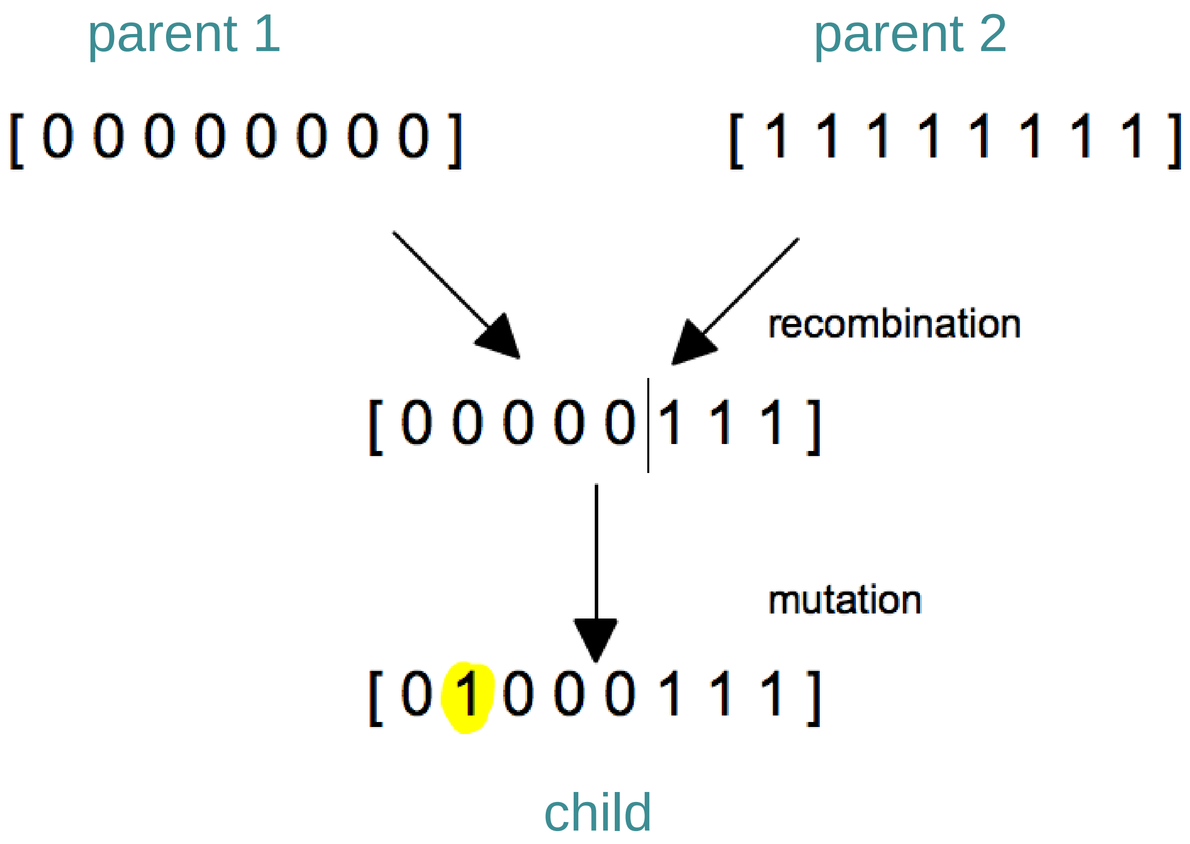
Evolutionary mechanisms

- Selection
  - Good / fit individuals have a higher chance of reproducing
- Inheritance
  - Properties from parents are transferred to offspring
- Variation
  - Changes in the genome adjust the behavior of the offspring, sometimes to the better

Selection

- Each *individual* in a population is evaluated and assigned a *fitness* value, ie. a measure of how a solution performs a given task
  - Example: The forward speed of a robot
  - Henriette: measured by the angular difference from the rotation encoder over 3 repetitions of the sequence
- The probability of an individual being selected for reproduction is proportional to its fitness value (randomness is present)

Inheritance + variation



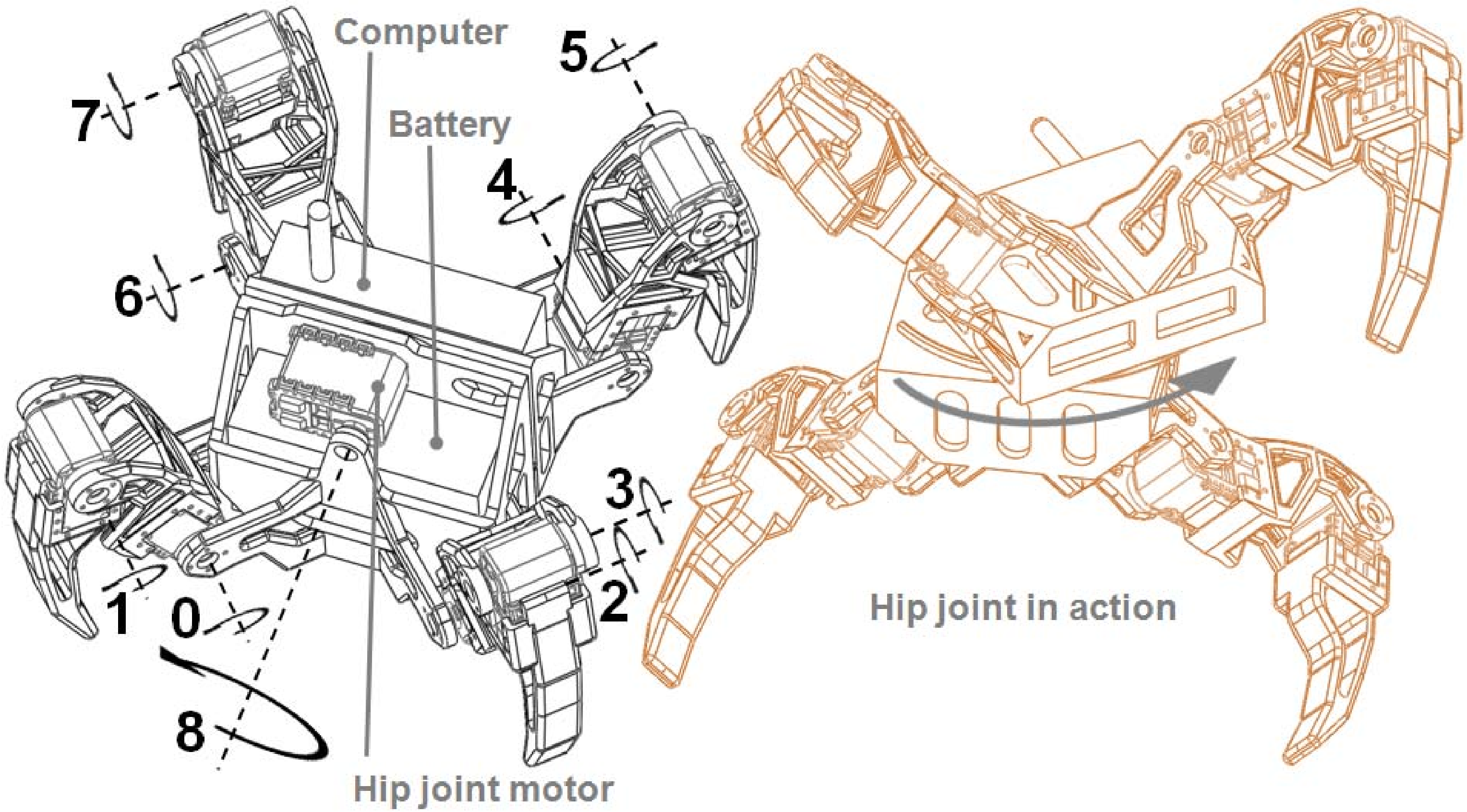
Without bio-terminology, what is an EA?

- A population-based stochastic search algorithm
  - Searching for ***satisfactory*** solutions in a solution space of all possible solutions
  - Searches in «parallel» on a population of solutions
  - Black-box: does not assume knowledge about the problem (but the results depend on the mapping and fitness function)
- Can handle large search spaces with complex fitness landscape
  - Less chance of being stuck in local optima
- Can give unexpected results →

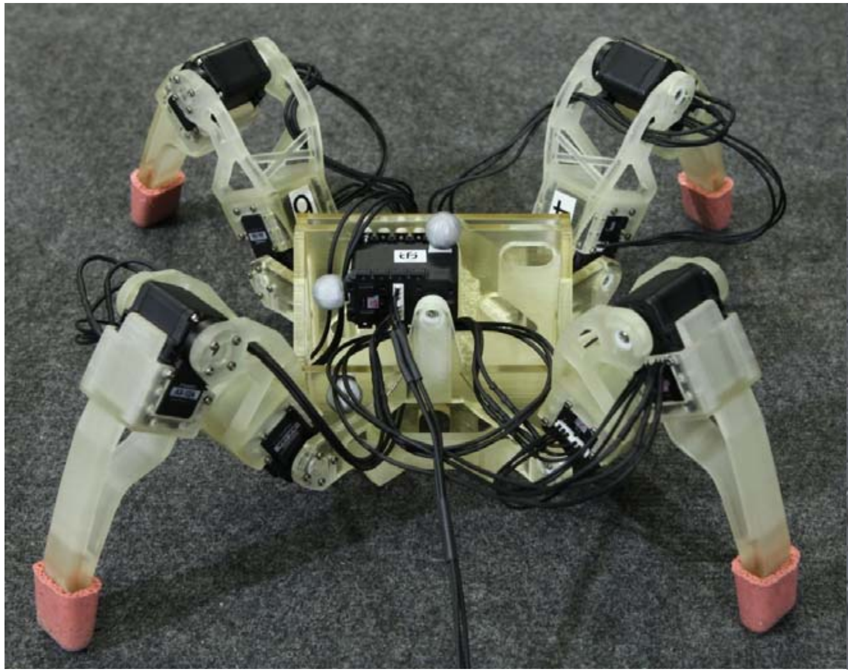
Simulation

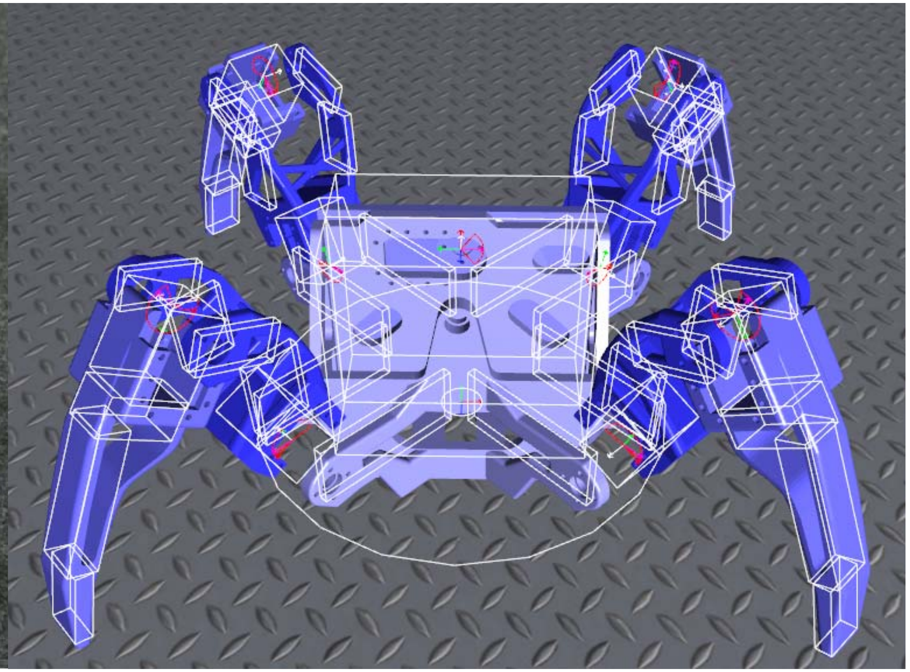
- Evolution on a real robot is impractical
  - Time consuming
  - Requires supervision: can get stuck, fall over
  - Mechanical wear
- Simulation should help
  - Allows automated evaluation
  - Can be much faster
    - especially with parallel computation

Example: Quadratot



Quadratot: Hardware and model (demo)

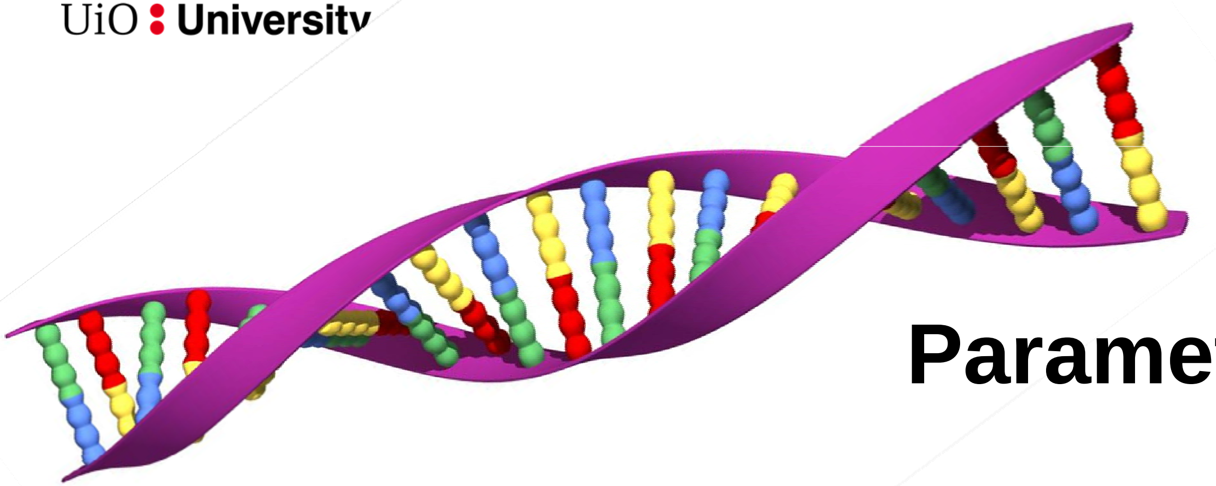




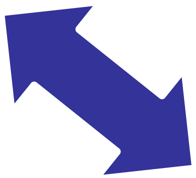
3D printed parts
AX12/18 servos
Silicone rubber socks

NVIDIA PhysX
Revolute motor joints
Rigid bodies (boxes)

UiO : University

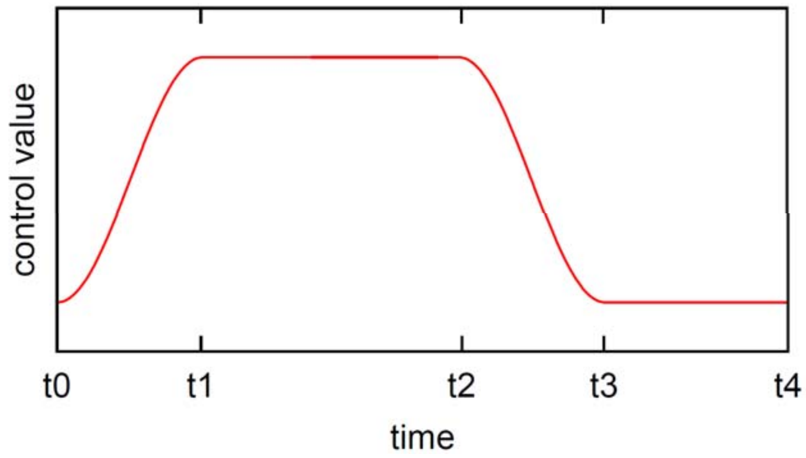


Quadratot: Parameterized control (mapping)



For each joint:

- Curve shape parameters (4)
- Phase
- Amplitude
- Center angle



Quadratot: Genetic algorithm (GA)

Genome length	314 bits
Population size	200
Number of generations	300
Mutation rate	1/314
Crossover rate	0.2

Initialize random population

Evaluate individuals (simulator)

Termination criterion reached?

yes

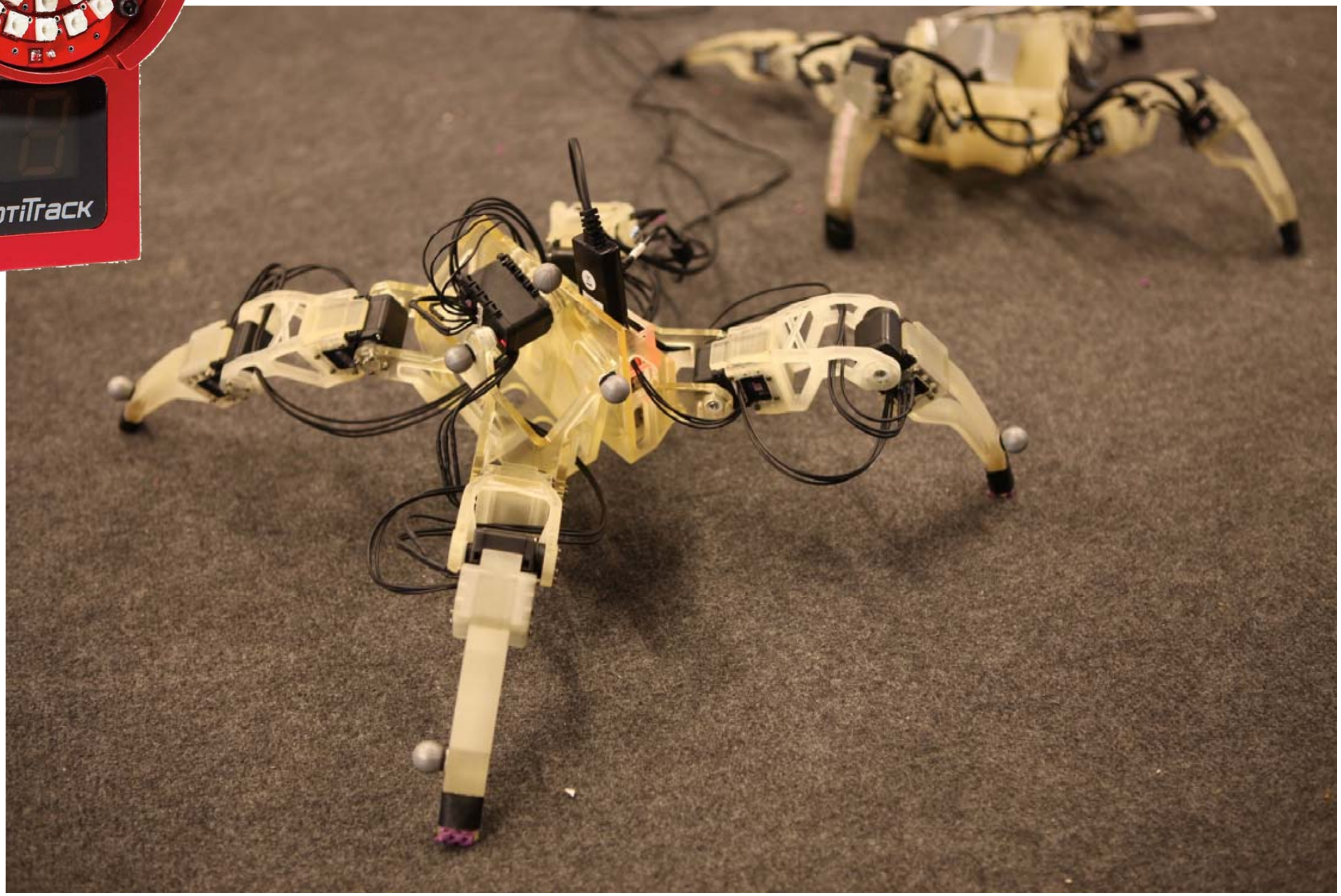
Verify solution(s) on robot

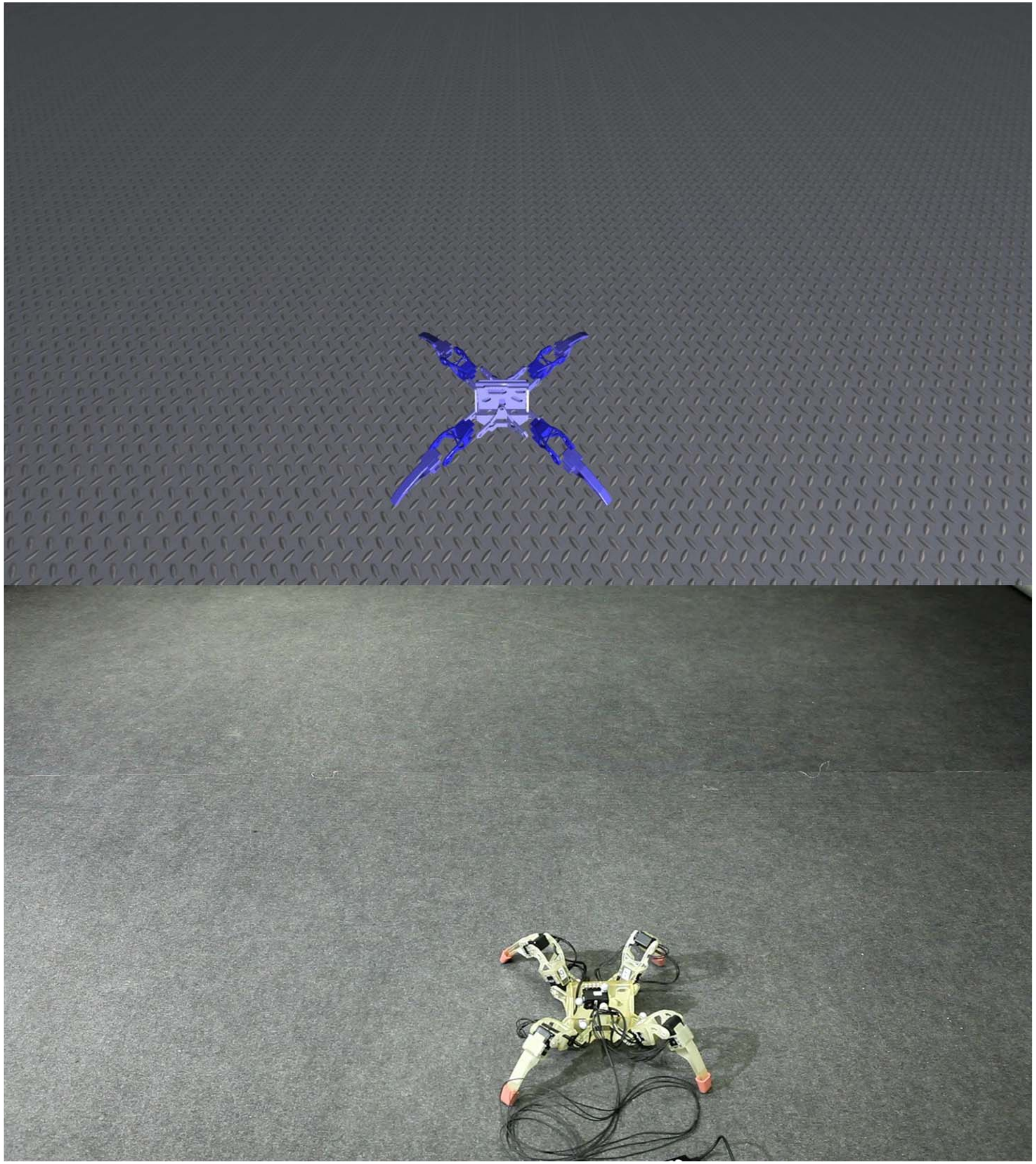
no

Create new population from good individuals

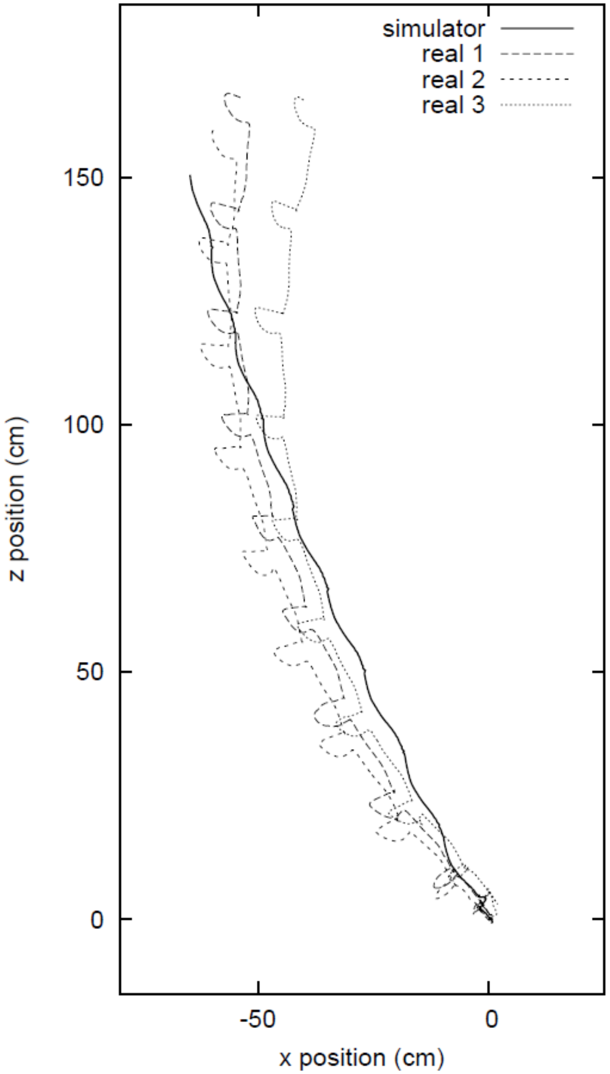
200 x 300 = 60 000 tests per evolutionary run!

Motion capture

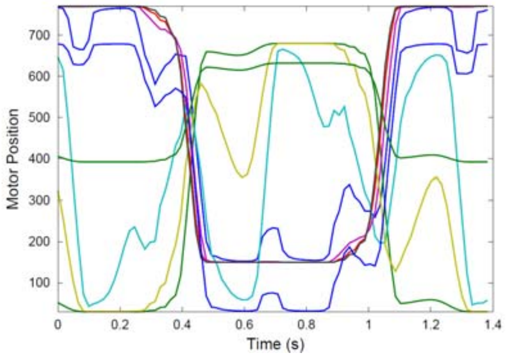


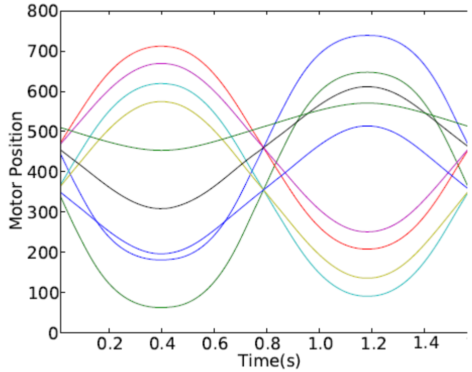


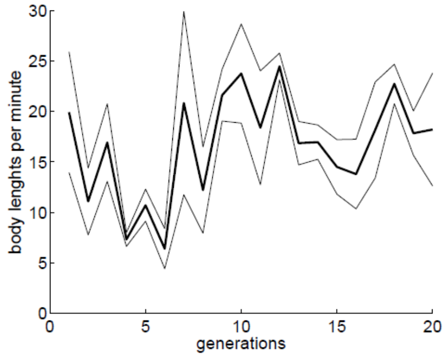
Quadratot: Evolved gait

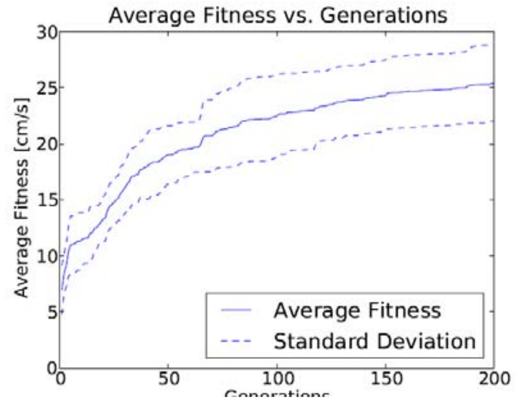


Benefits of simulation









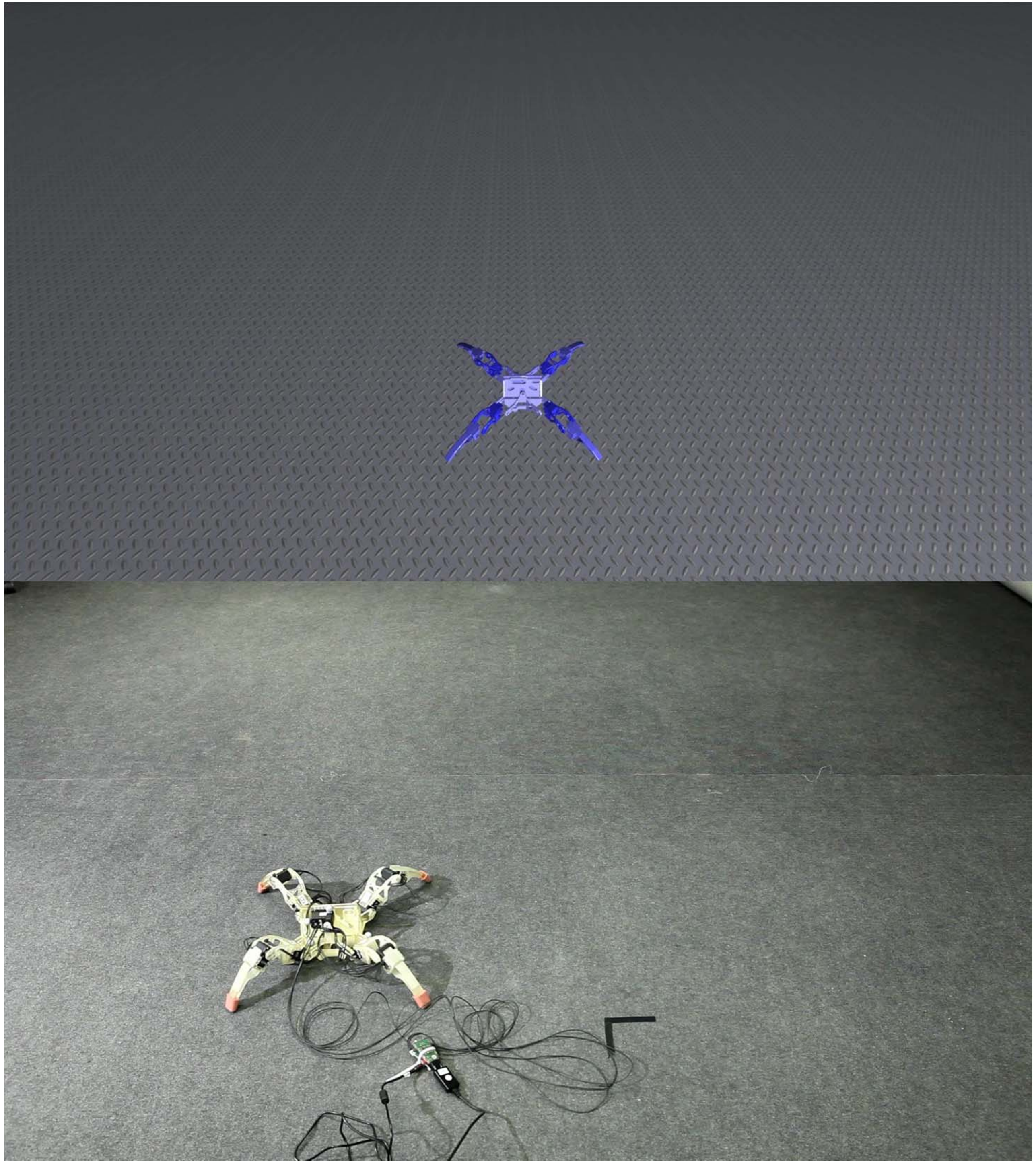
	Evaluations	Simulated Velocity	Real Vel. (CCML)	Real Vel. (ROBIN)
Parameterized gaits + optimization [25]	153	–	5.8	–
HyperNEAT in hardware [25]	180	–	9.7	–
RL PoWER Spline [18]	300	–	11.1	–
GA + simulator [9]	60000	*16.7	13.8	**17.8**
HyperNEAT + simulator [this paper]	40000	****25.4**	**14.5**	–

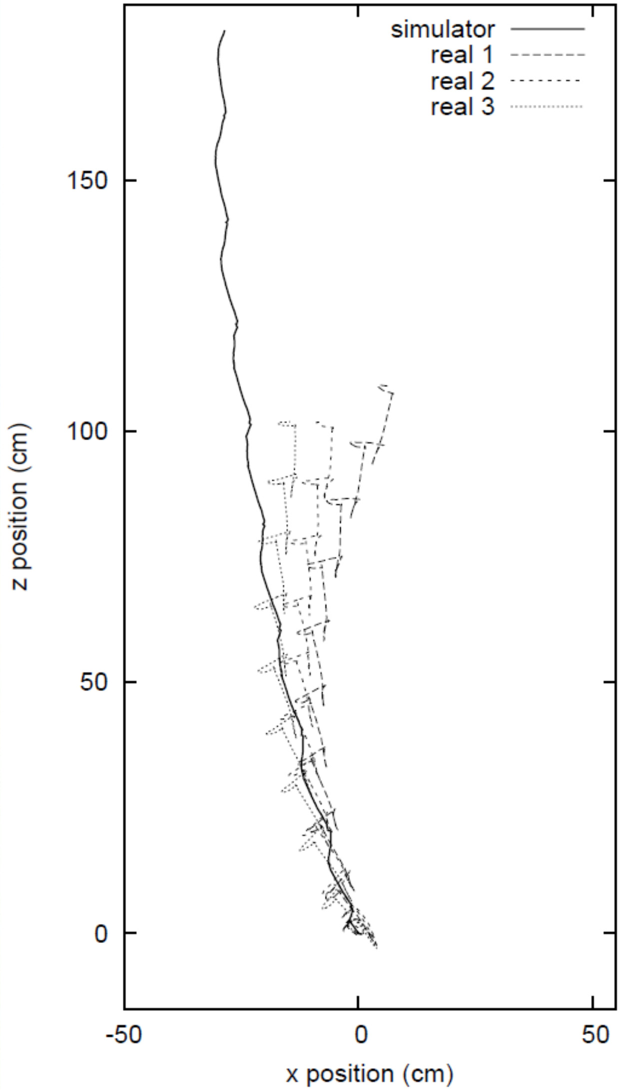
Reality gap

- A simulator cannot capture all aspects of reality
- Evolved solutions may exploit features of the simulator not present in reality

→ The solutions evolved in simulation behave differently when applied to the real robot!

Quadratot: Reality gap





How to deal with the reality gap?

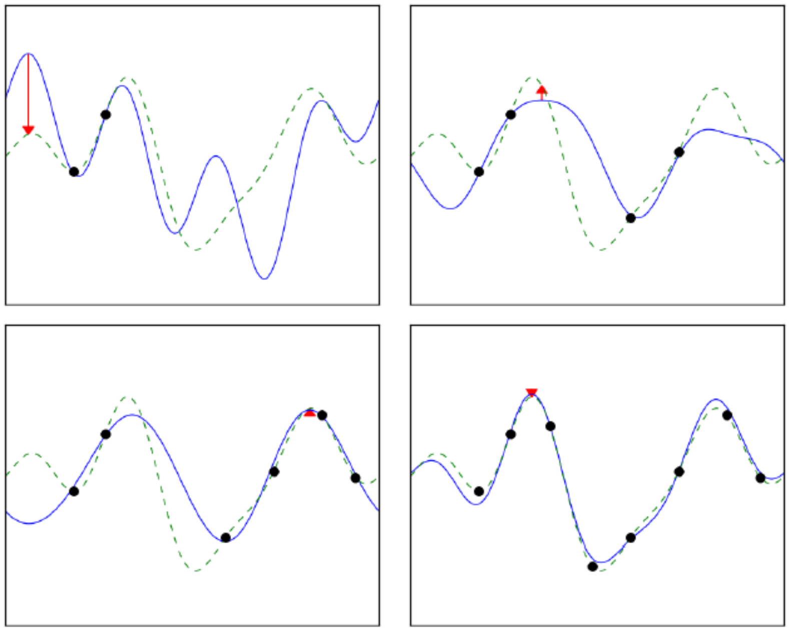
- Ideas?

How to deal with the reality gap

1. Increase simulation fidelity
   – Manually: do more precise measurements, increase solver accuracy
   – Automatically: measure deviation simulation-reality, auto-tune simulator for smaller deviation
2. Do not allow for solutions using badly simulated behaviour
   – Manually: E.g. Encourage slow, static movements
   – Automatically: Avoid solution types that transfer poorly
3. Online learning after deployment on real robot
   – Can use more evolution, reinforcement learning, or other method

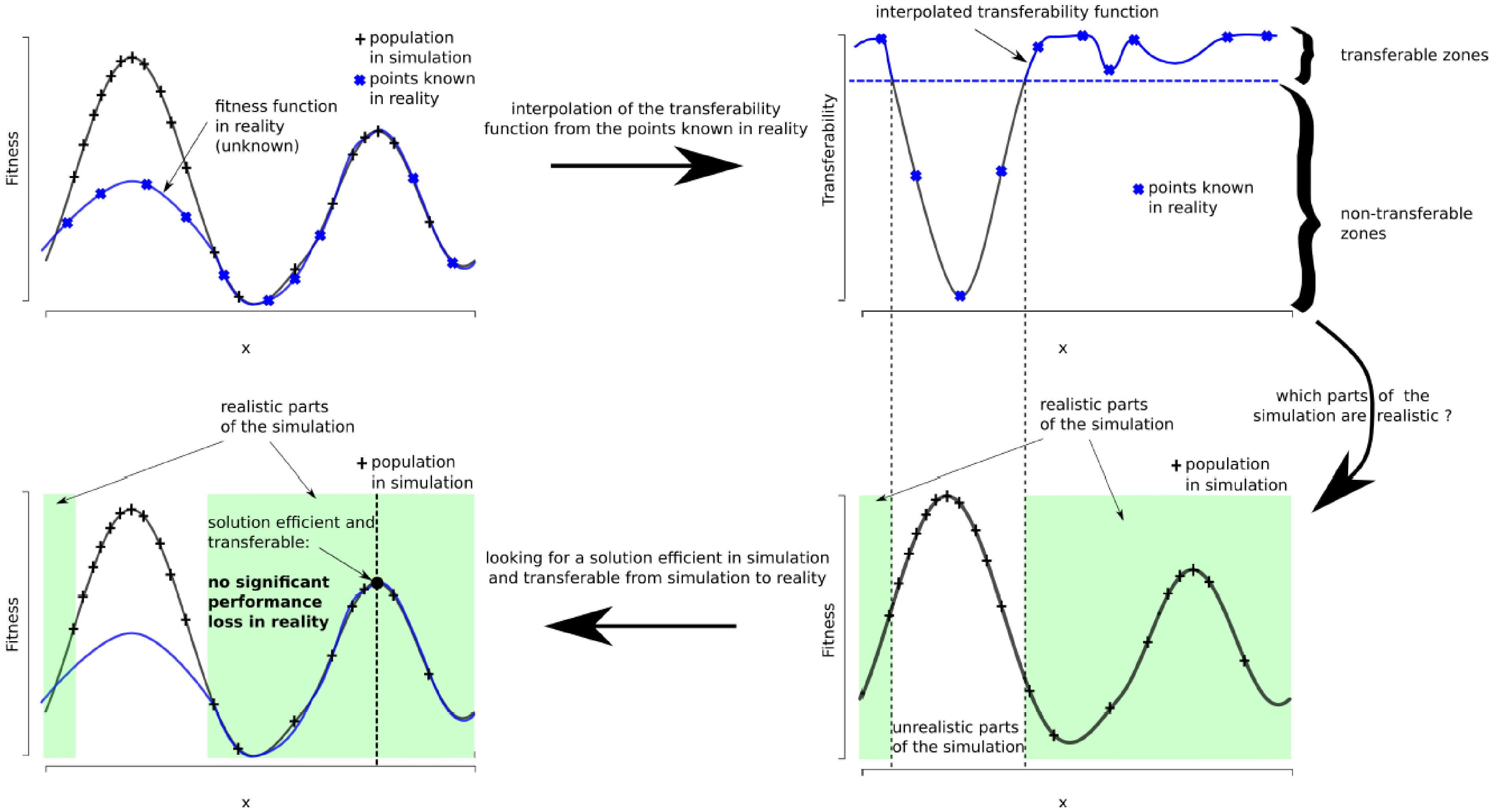
1. Automatic simulator tuning

- Sample from real world
  - Test selected solutions on real robot
- Tune (evolve) simulator to fit all samples
- Evolve new solutions using tuned simulator



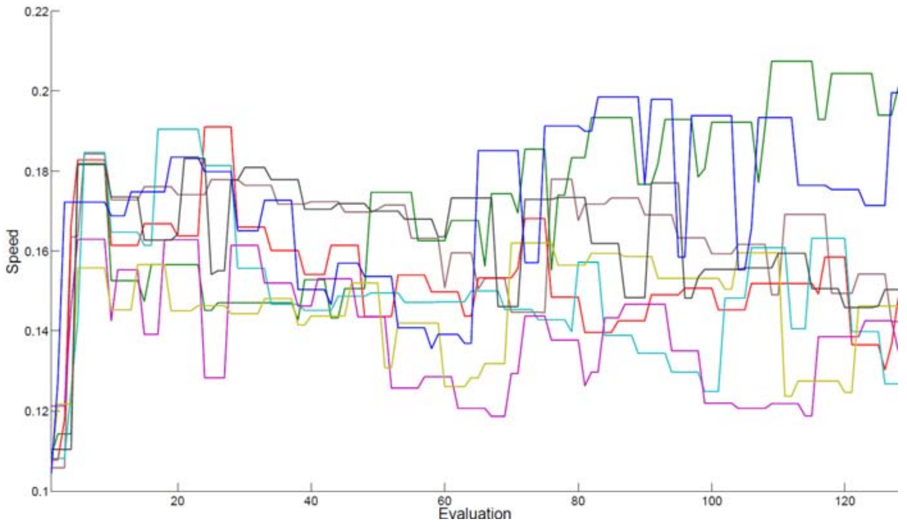
http://youtu.be/qDPbXvADyio

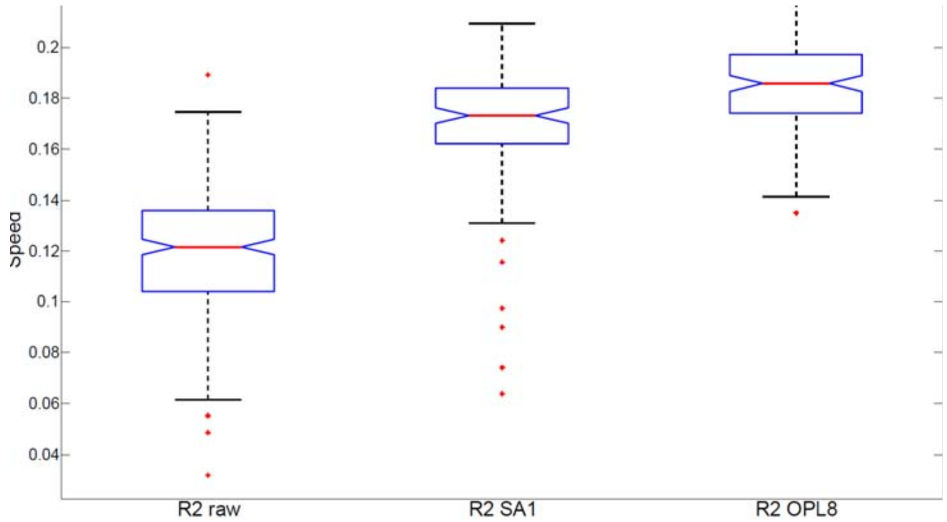
2. Transferability (UPMC, Paris)



3. Adaptation after transferral (video)

- Reality gap is «accepted»
- Adaptation algorithm is carried out on the real robot
- Needs to take into account fewer tests and more noise

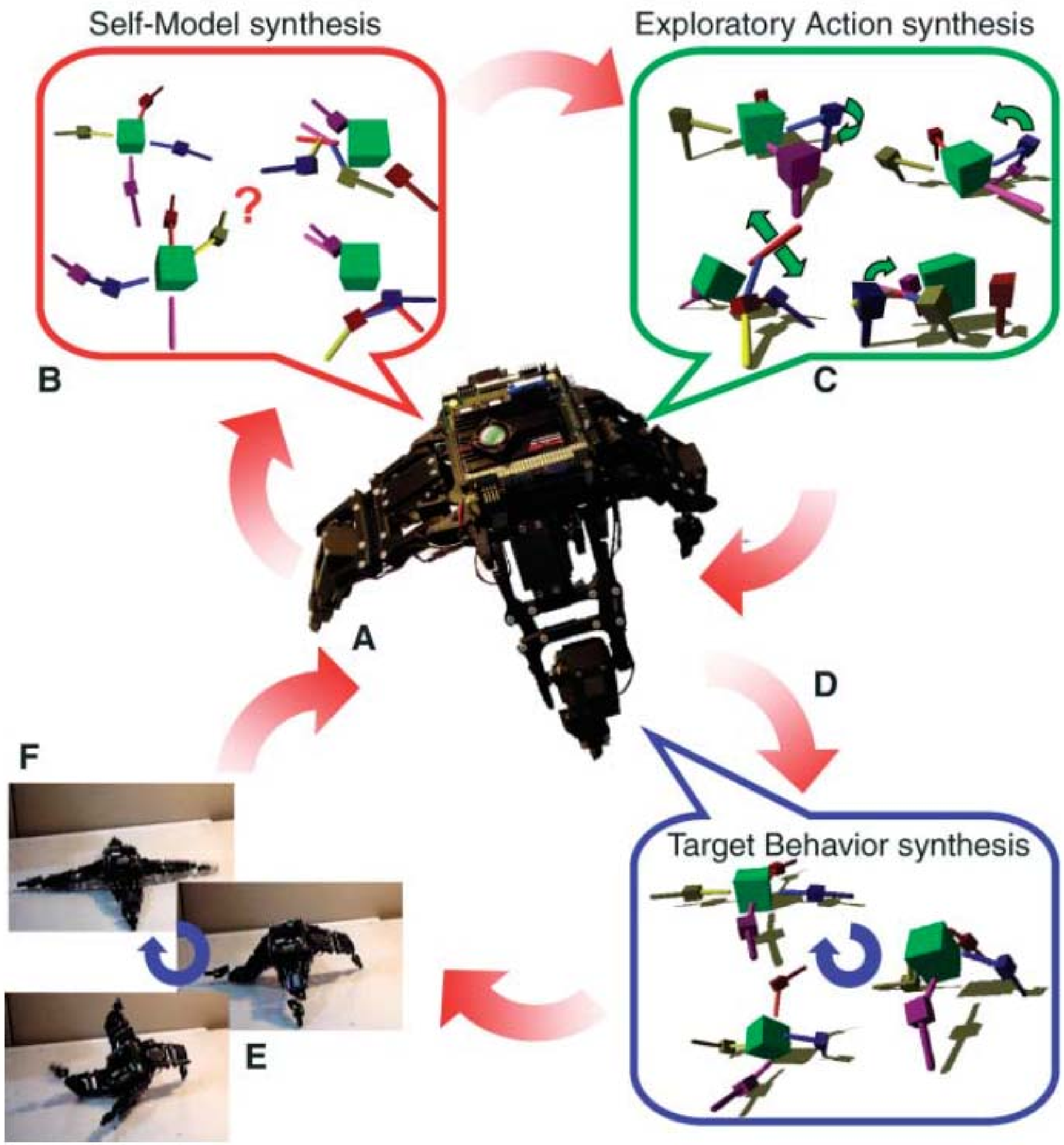




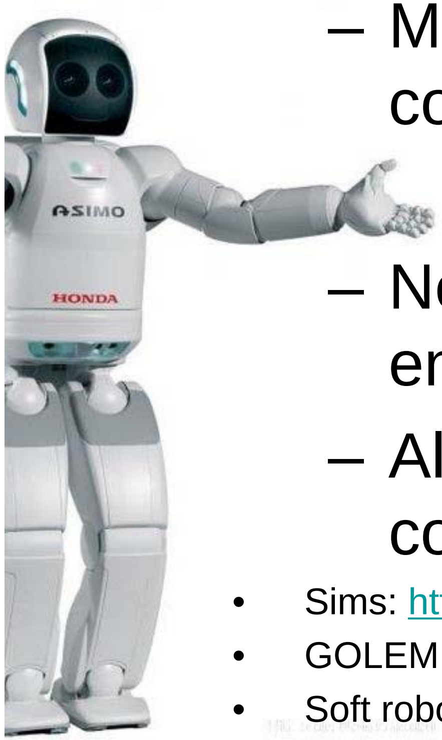
Self-modeling robot (Cornell U.)

- Creates self-model through exploratory actions
- Uses evolution to search for walking pattern using self-model
- If the robot is broken, a new self-model is constructed

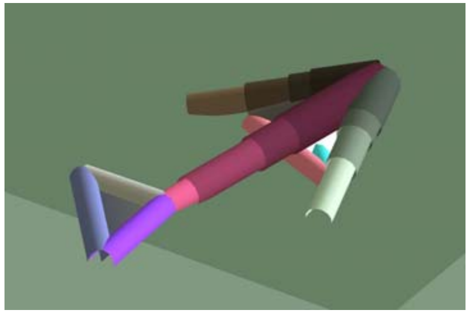
http://youtu.be/3HFAB7frZWM

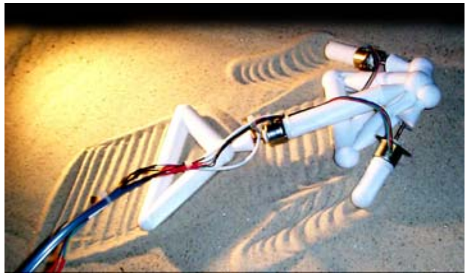


Evolving shape and control



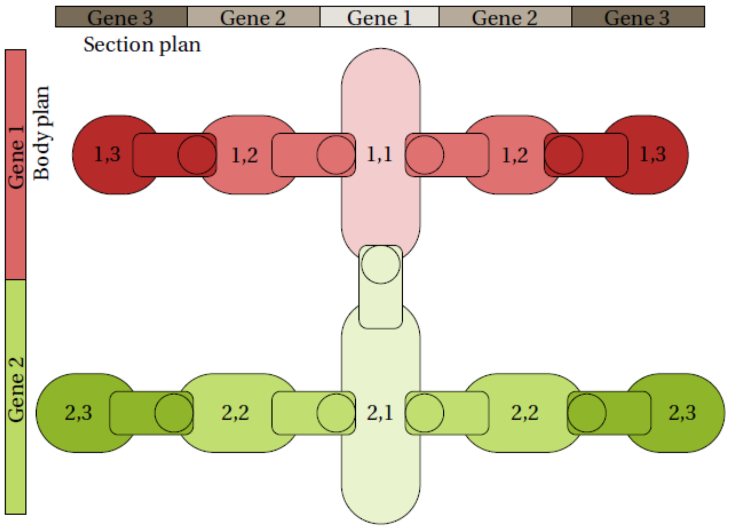
- Physics simulation allows evolution of shape and control simultaneously
  - More efficient designs for complex problems?
  - New designs for new environments?
  - Allows for offloading computation to the body?
- Sims: http://youtu.be/JBgG_VSP7f8
- GOLEM: http://youtu.be/sLtXXFw_q8c
- Soft robot: http://youtu.be/z9ptOeByLA4



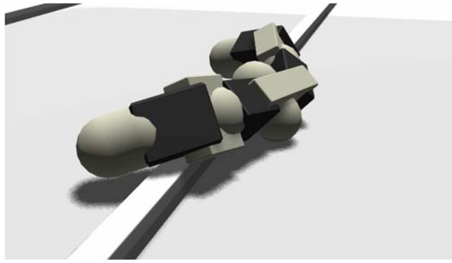


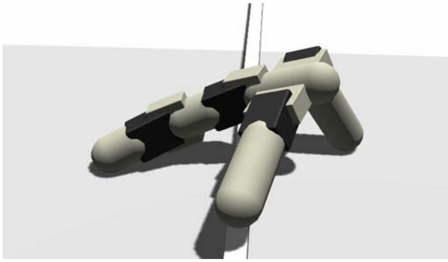
Example: «hox» body evolution

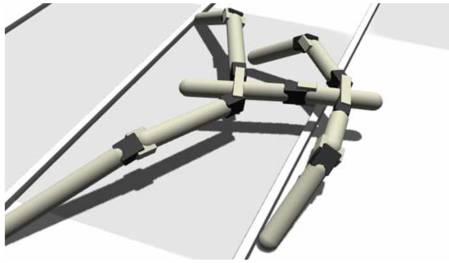
- Bio-inspired, generative approach
  - Allows a variety of bodies from a compact code
- Designed for production with 3D printer and commercial servos

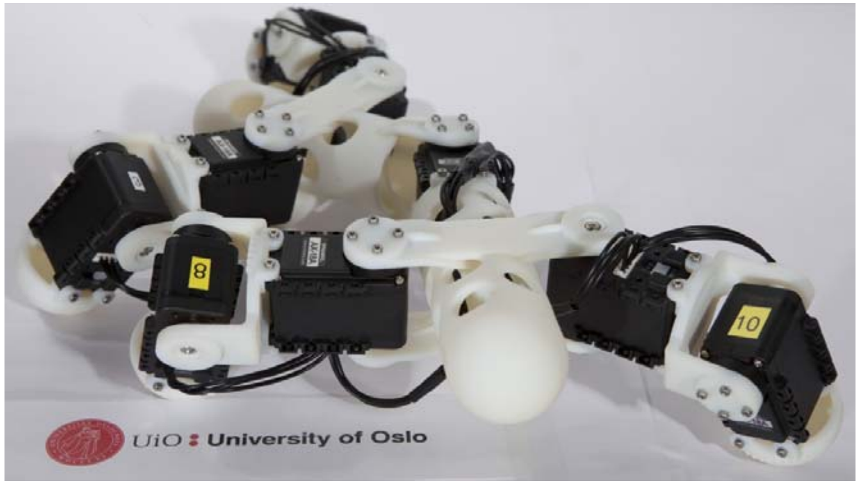


«hox»: Some results (video)



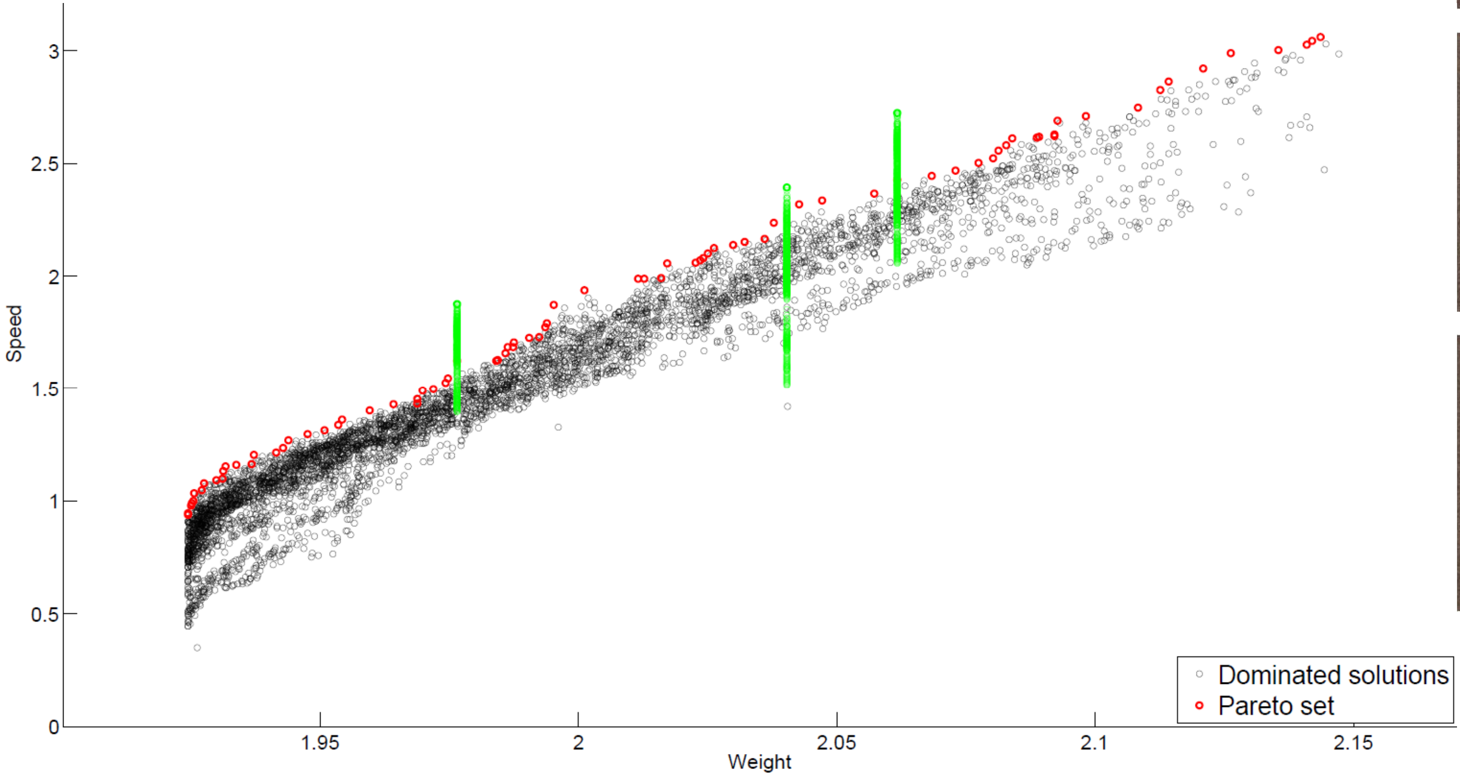


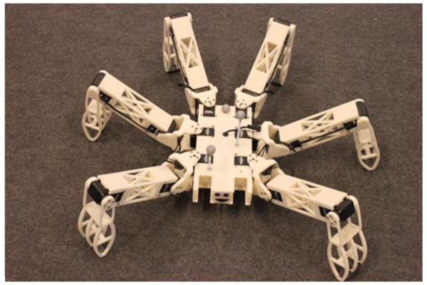


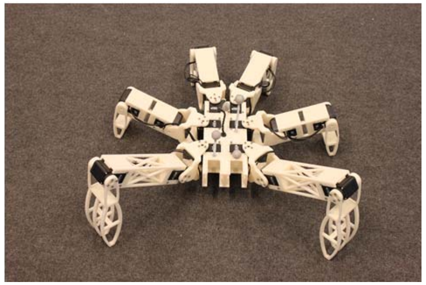


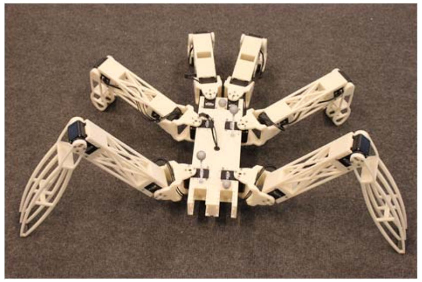
Example: Karkinos (MSc. project)

- Hybrid automatic / engineered design of robot shape and control









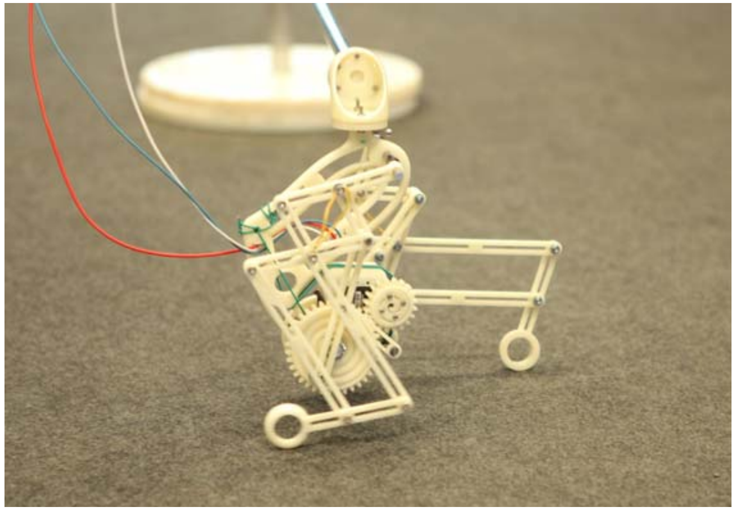
Master's projects in evolutionary robotics at the ROBIN group

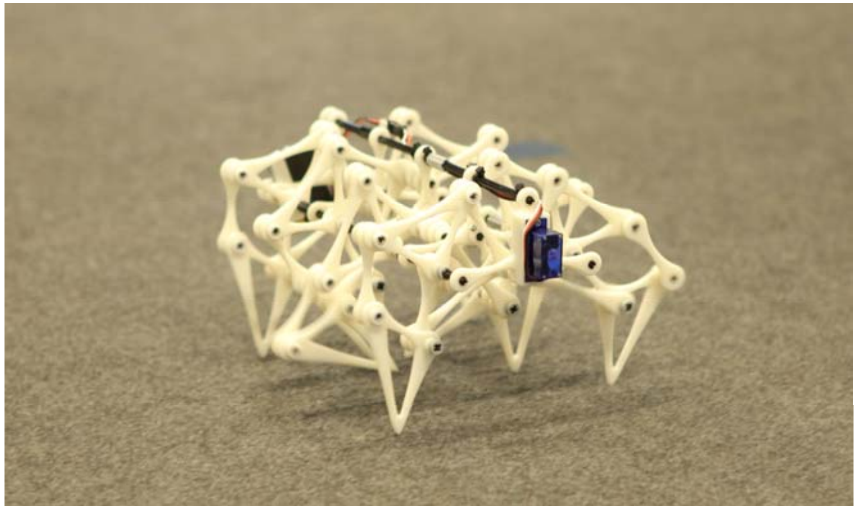
- Integration of locomotion learning platform
  (evolutionary algorithm + simulator + hardware interface + motion capture)
- Evolution of locomotion patterns for robots
  (walking, crawling, obstacles, robustness, neural networks, comparing with other learning methods)
- Reality gap research
  (testing various algorithms for a smooth transfer from simulator to reality)
- Design and build new robot
  (CAD, 3D print, electronics, simulator)

http://www.mn.uio.no/ifi/studier/masteroppgaver/robin/

Relevant courses

- INF3490 Biologically inspired computing
- INF4500 Rapid prototyping of robotic systems





Summary

- Evolutionary robotics can be useful for adaptation, optimization, design exploration
- Simulation is useful for evolutionary search
- The reality gap remains a research challenge
  - Simulator tuning, transferability, online adaptation
- Co-evolution of body and control gives new possibilities

- Please continue with MSc. studies at ROBIN