

UiO : University of Oslo

**IN3140**

# Evolutionary robotics

Kyrre Glette

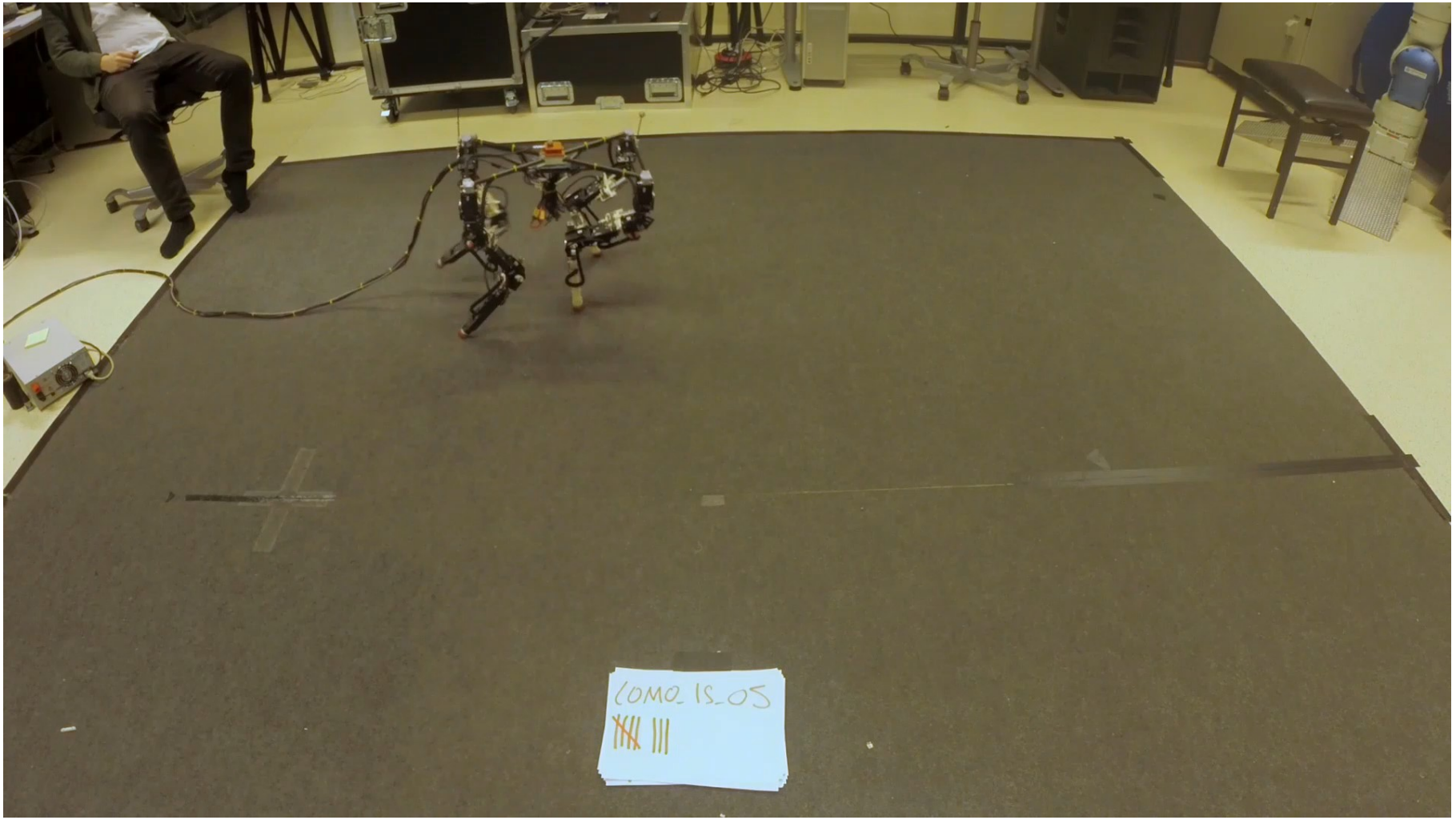
# Today: Evolutionary robotics

- Why evolutionary robotics
- Basics of evolutionary optimization
  - IN3050 will discuss algorithms in detail
- Illustrating examples
  - ROBIN in-house robotic platforms and experiments
- Research challenges
  - Reality gap

# DyRET:
# Dynamic Robot for Embodied Testing



# Evolutionary robotics!?!

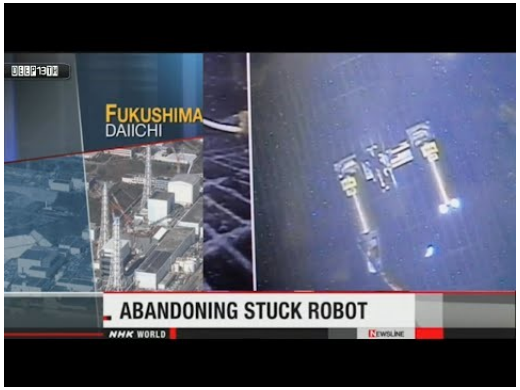


# Need for resilient and adaptive robots!

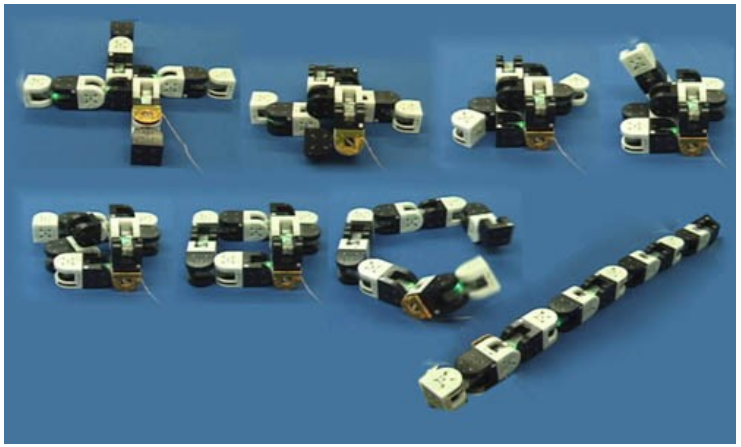


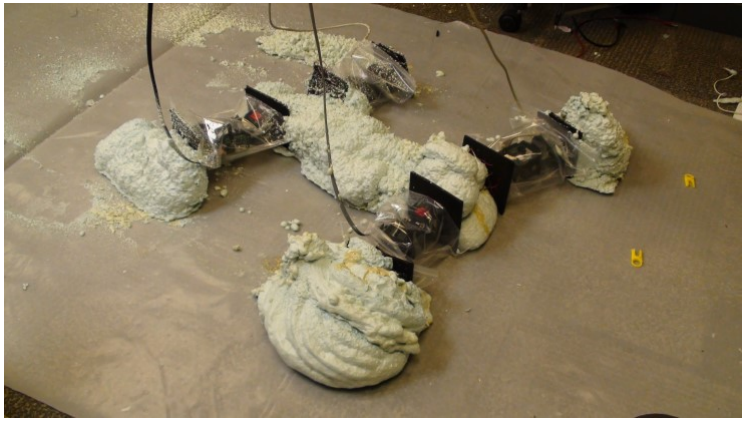


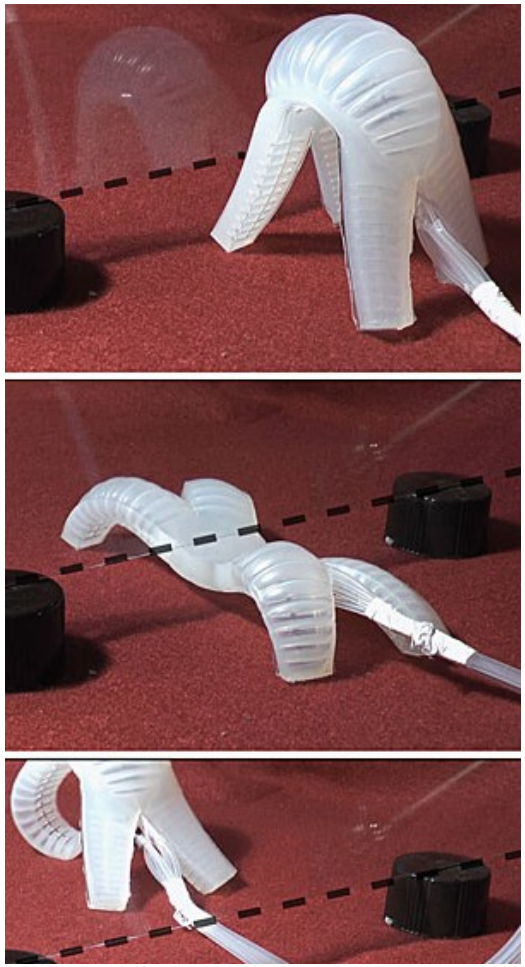




# Future robots & scenarios

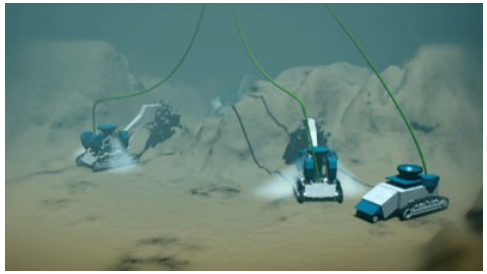










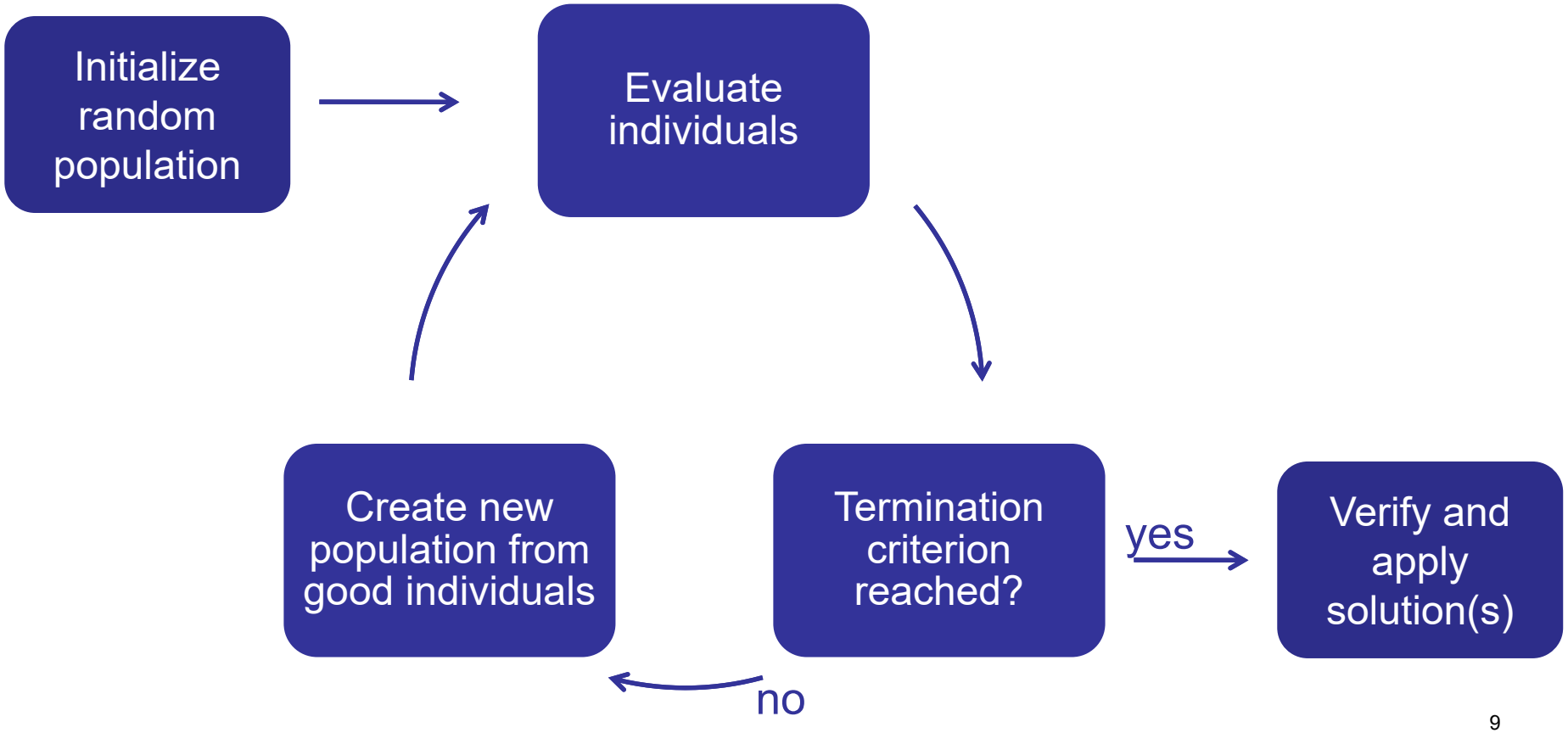


# Why evolutionary robotics?

- 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*

# Evolutionary Algorithm (EA)



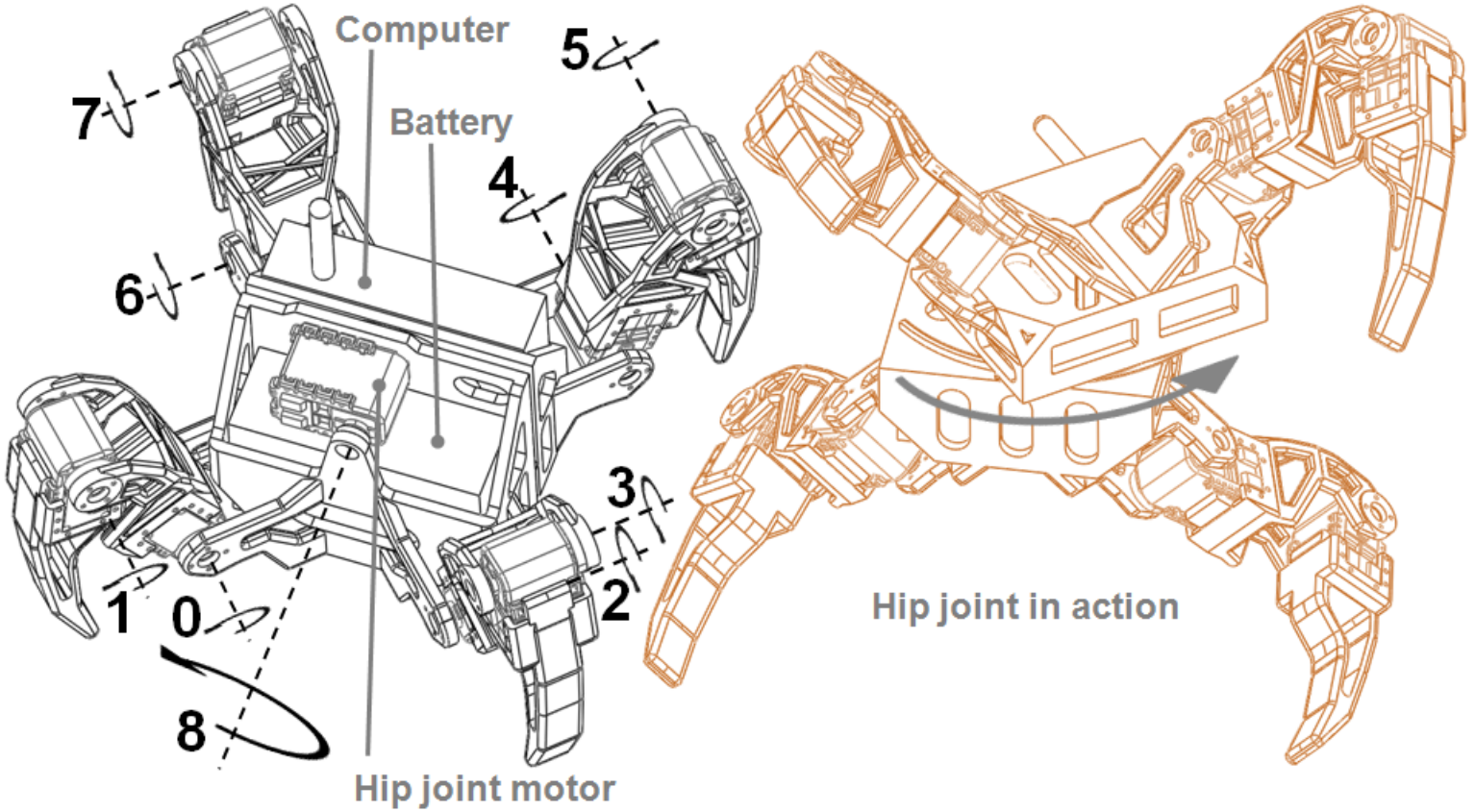
# Evolutionary mechanisms

- 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

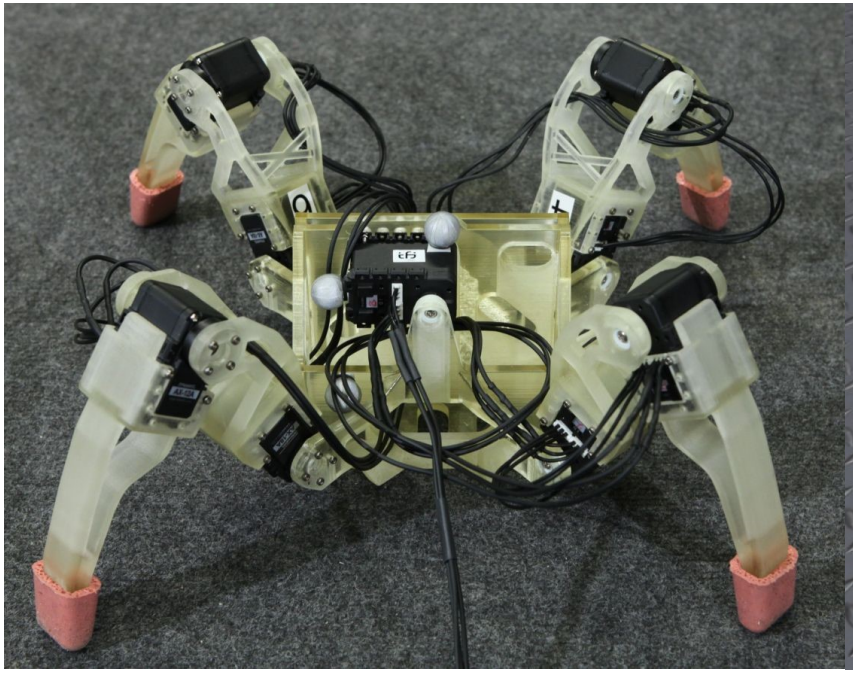
# Simulation

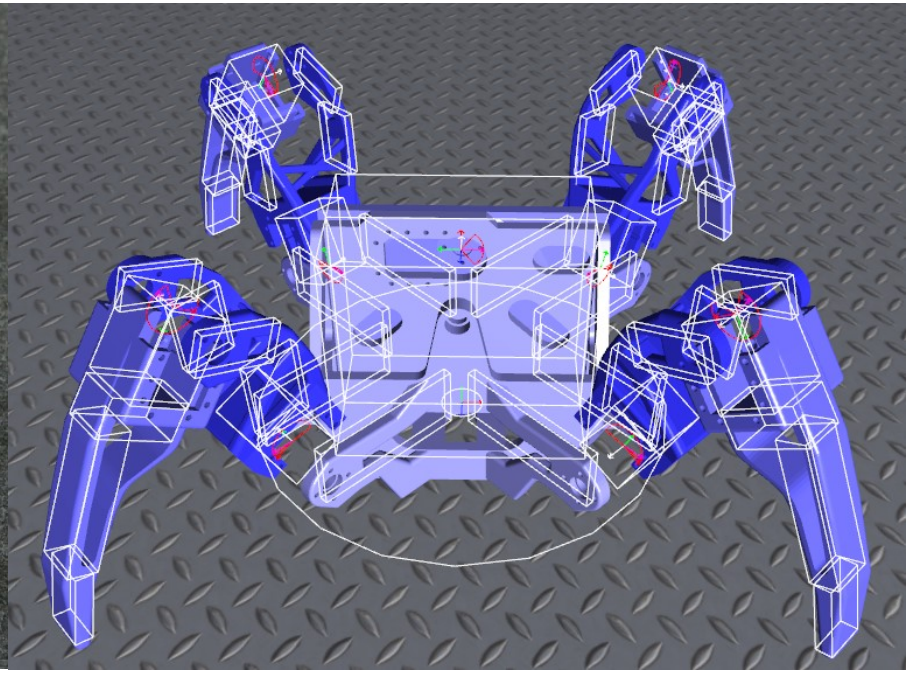
- Evolution on a real robot is impractical
  - Time consuming
  - Requires supervision: can get stuck, fall over
  - Mechanical wear
- Simulation should help
  - Allows automated evaluation
  - Can be much faster
    - especially with parallel computation

# Example: Quadratot



# Quadratot: Hardware and model





3D printed parts

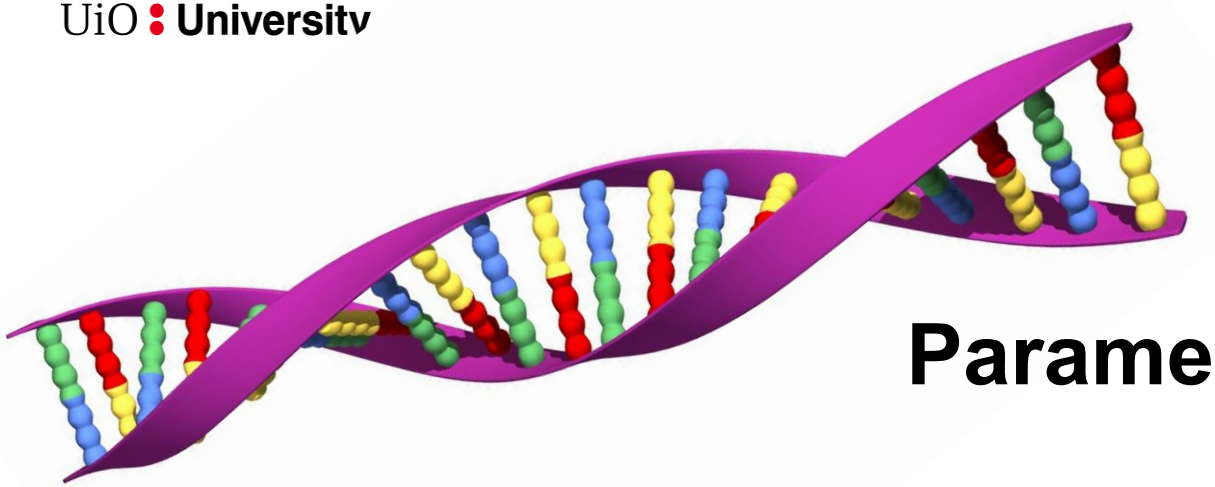
AX12/18 servos

Silicone rubber socks

NVIDIA PhysX

Revolute motor joints

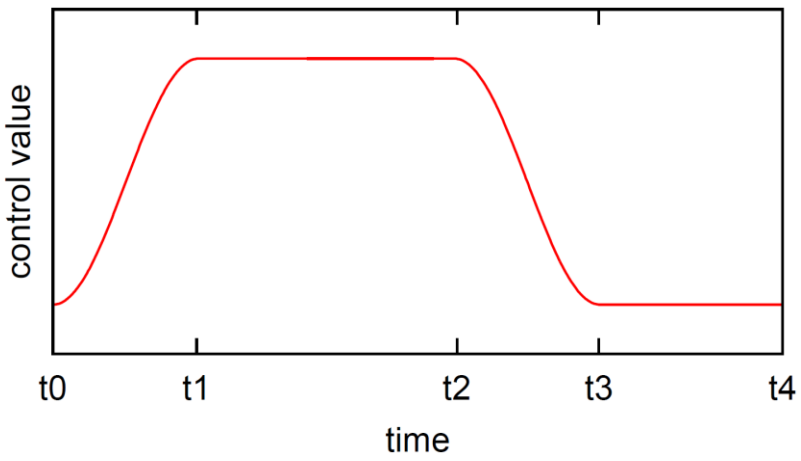
Rigid bodies (boxes)



# Quadratot: Parameterized control (mapping)

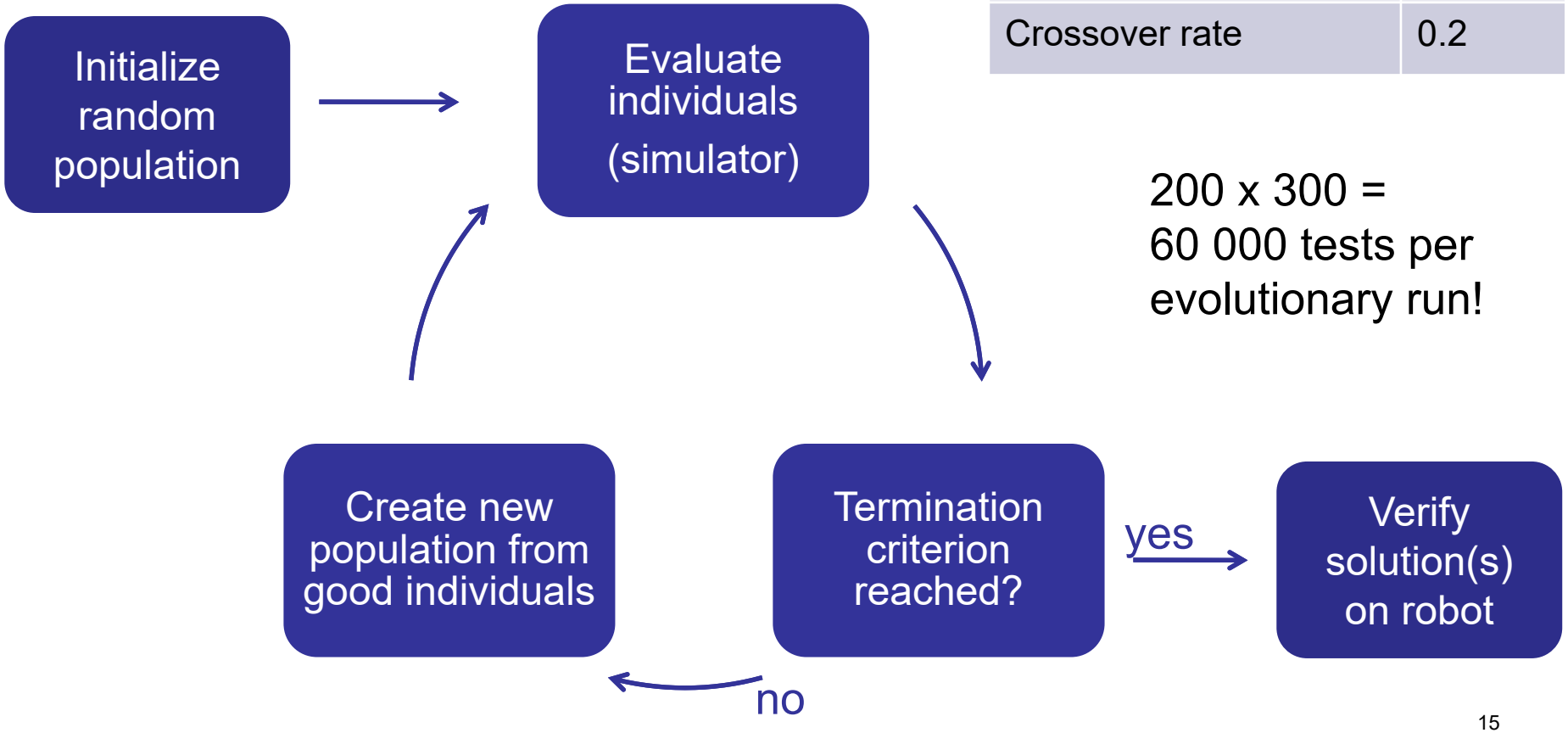
For each joint:

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



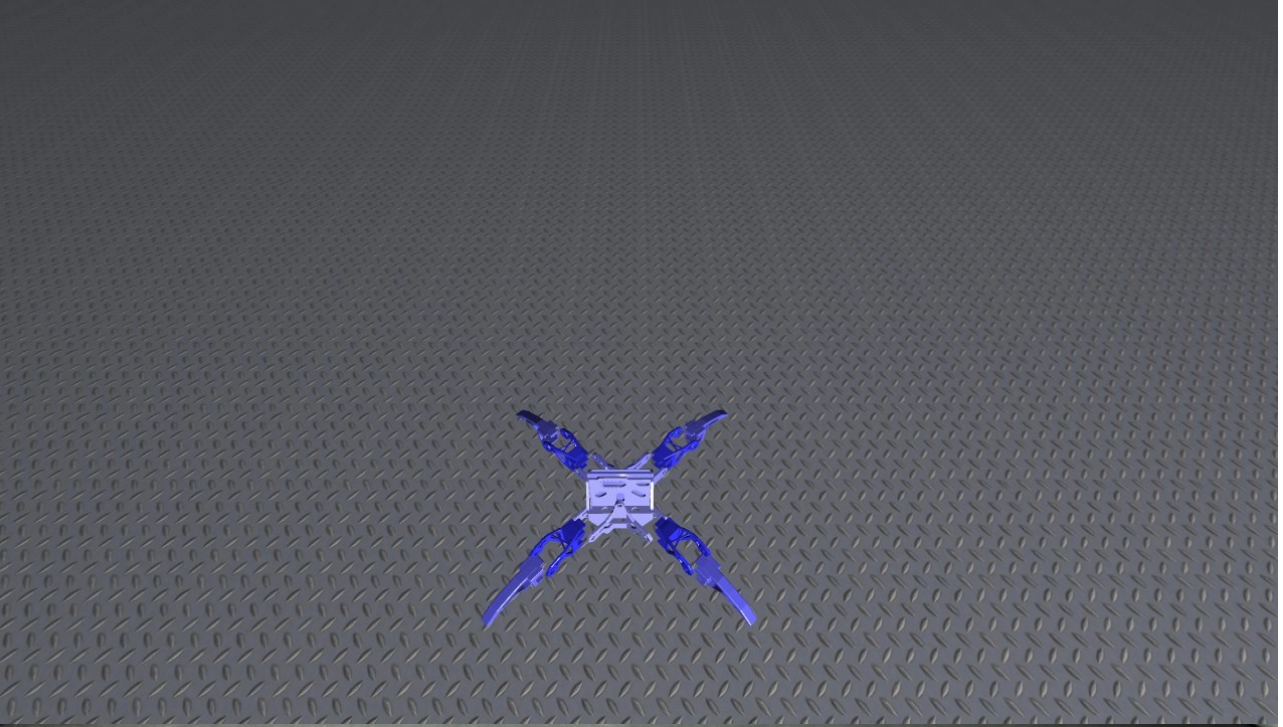
# Quadratot: Genetic algorithm (GA)

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

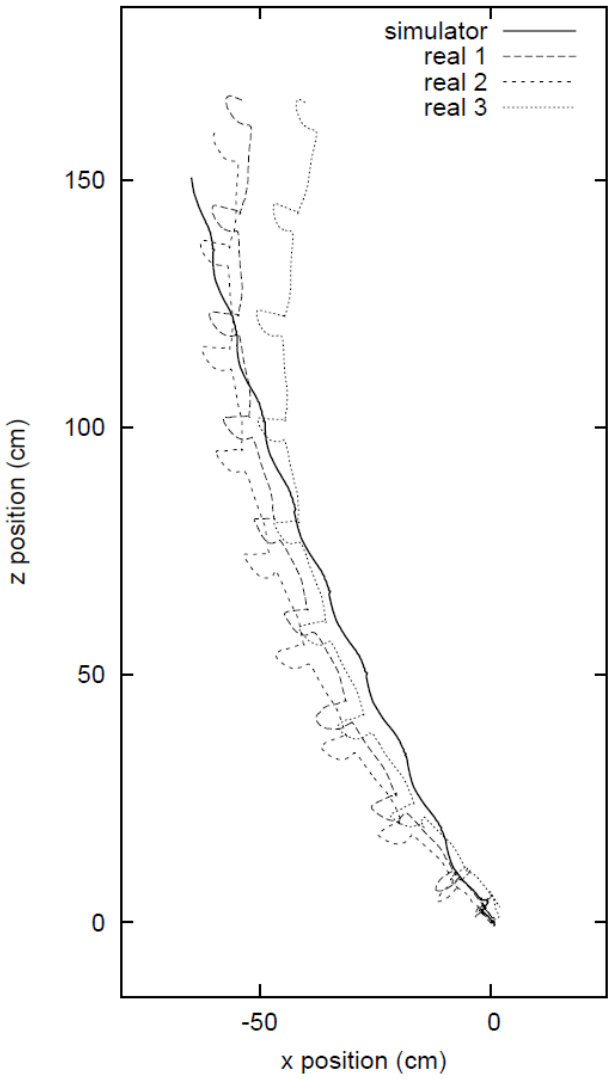



200 x 300 =
60 000 tests per evolutionary run!

# Quadratot: Evolved gait





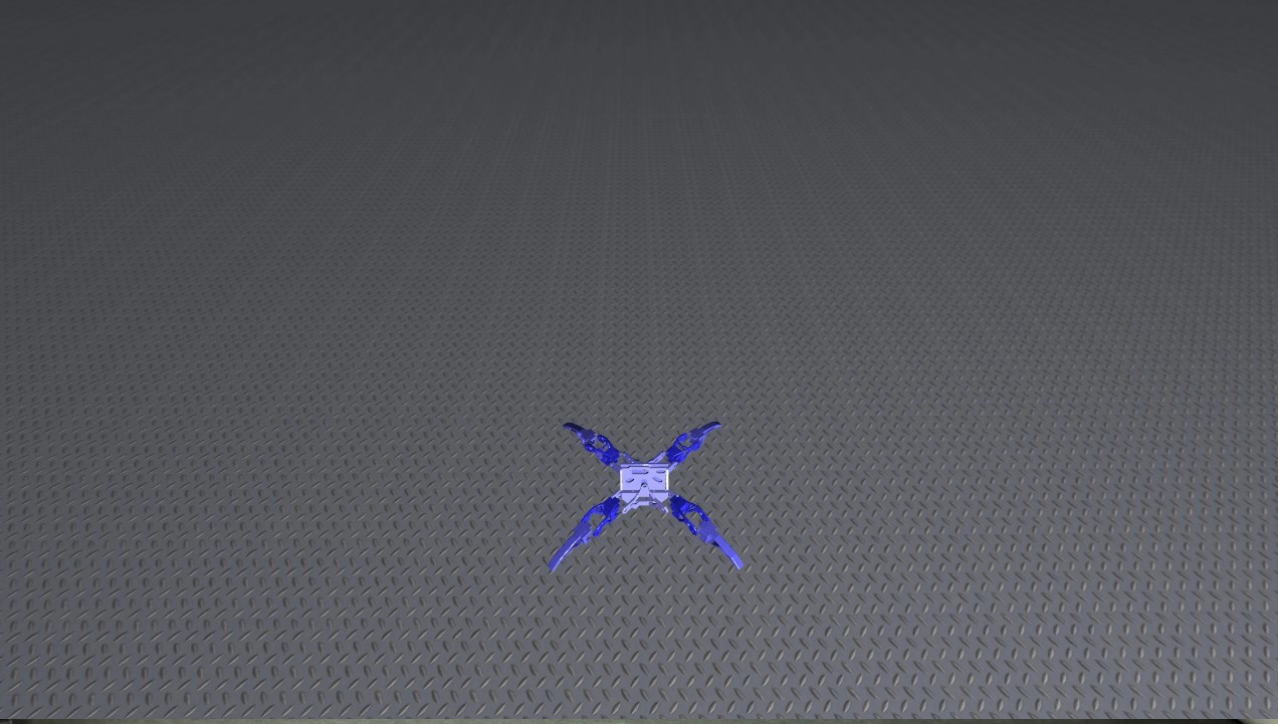


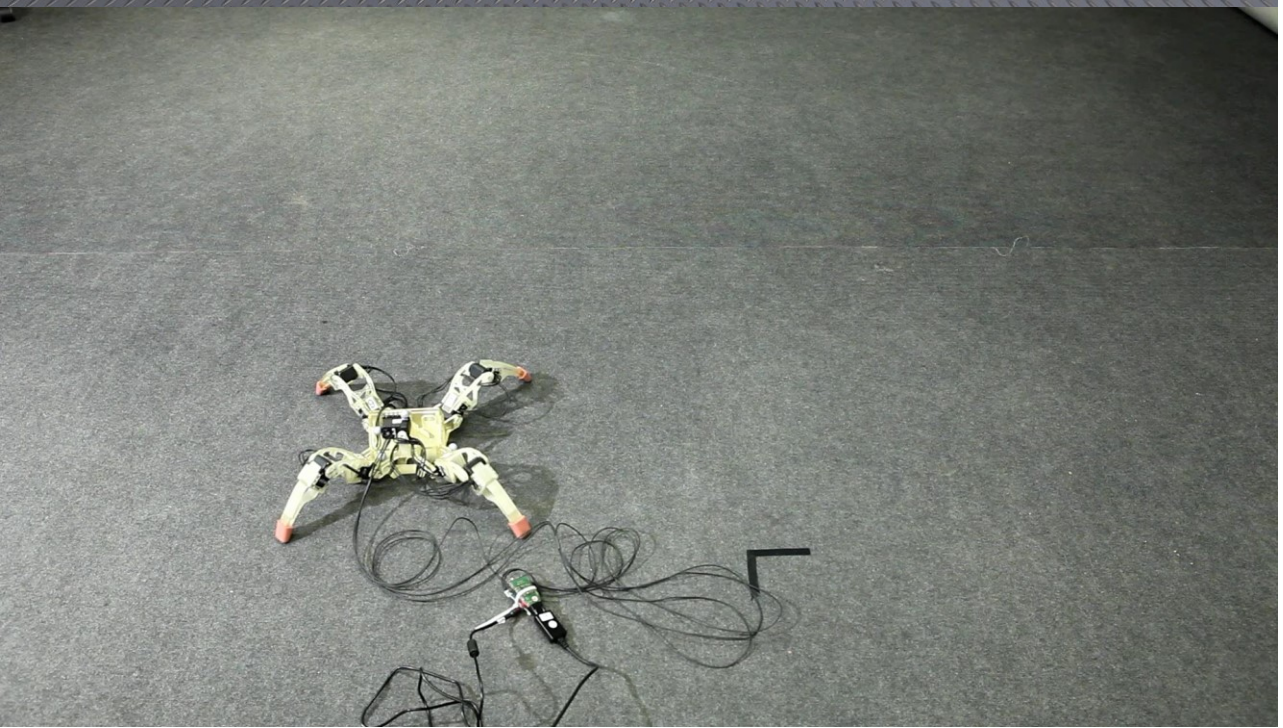
# Challenge: Reality gap

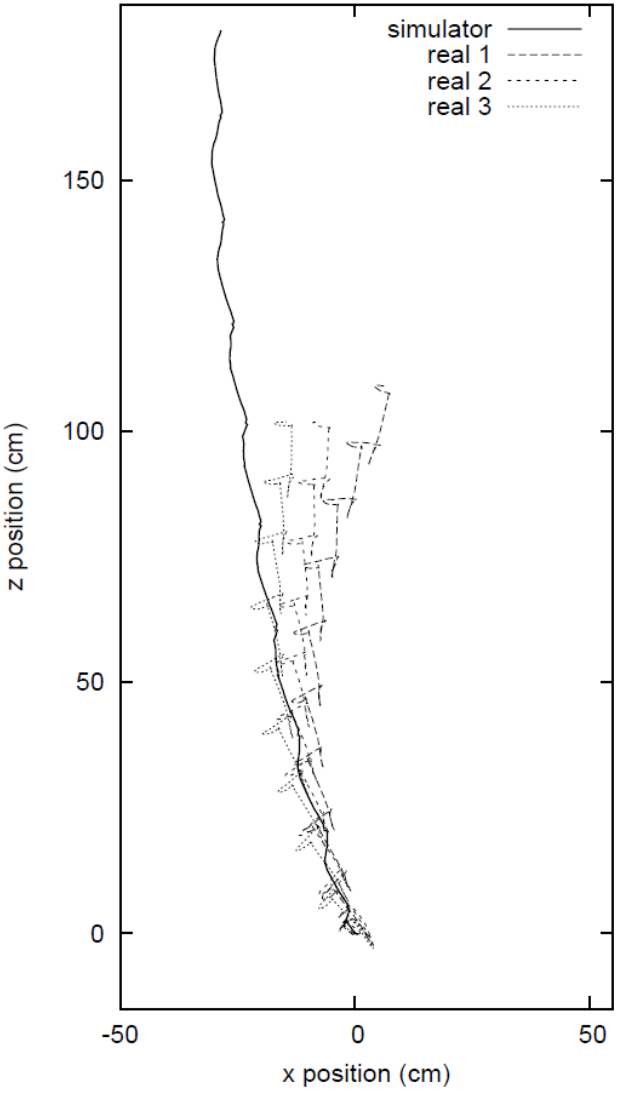
- A simulator cannot capture all aspects of reality
- Evolved solutions may exploit features of the simulator not present in reality

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

# Quadratot: Reality gap







# How to deal with the reality gap?

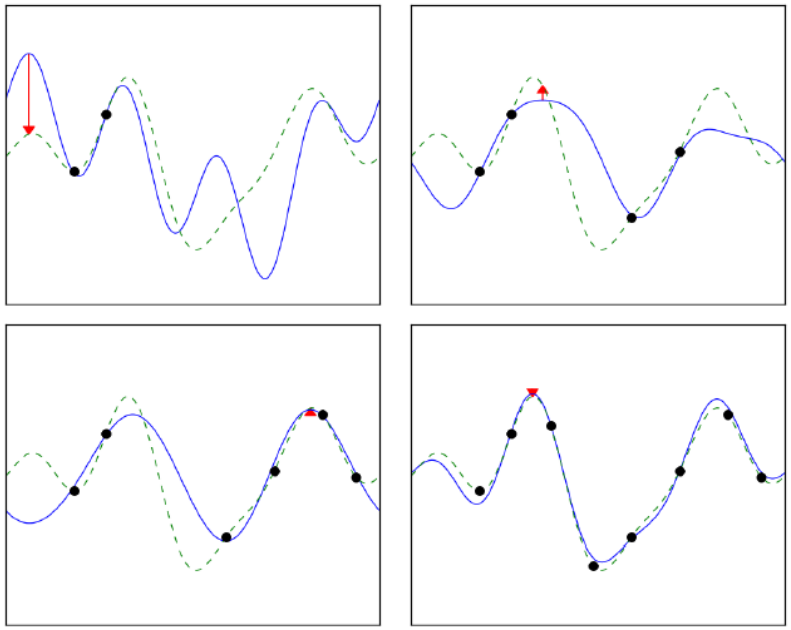
- Ideas?

# 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, add noise
   - 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

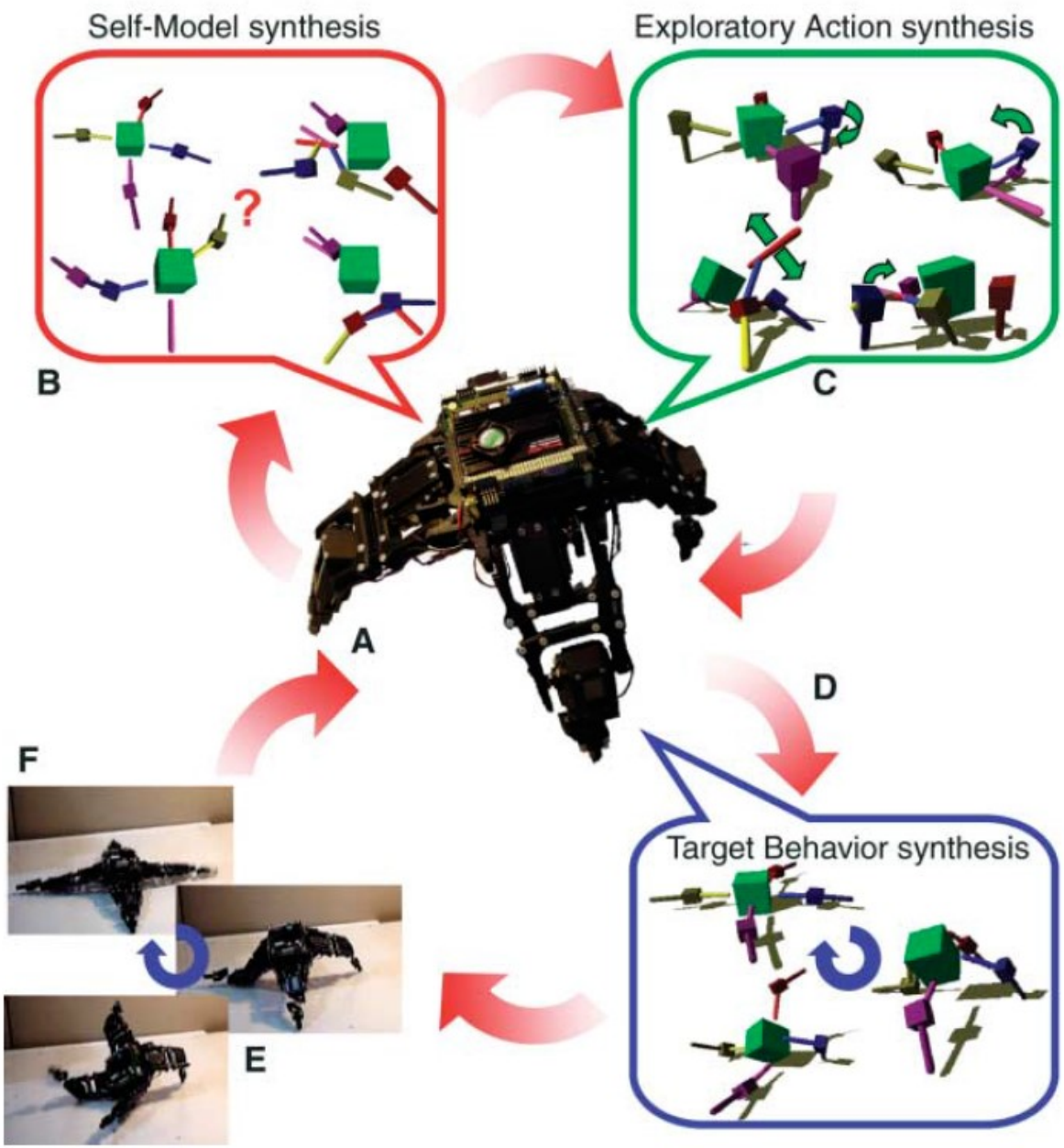
- Sample from real world
  - Test selected solutions on real robot
- Tune (evolve) simulator to fit all samples
- Evolve new solutions using tuned simulator



# Self-modeling robot (Cornell U.)

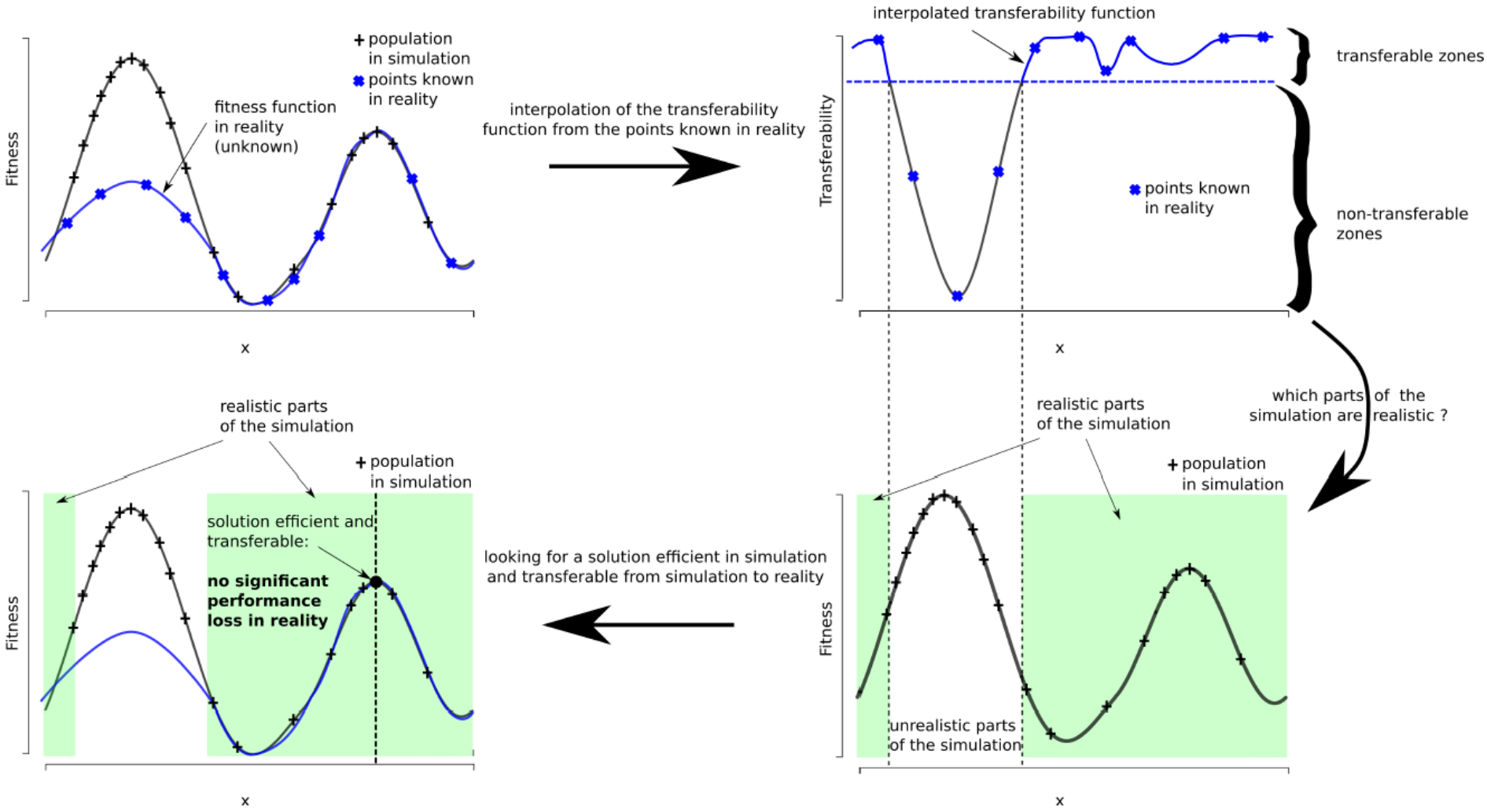
- Creates self-model through exploratory actions
- Uses evolution to search for walking pattern using self-model
- If the robot is broken, a new self-model is constructed

http://youtu.be/3HFAB7frZWM



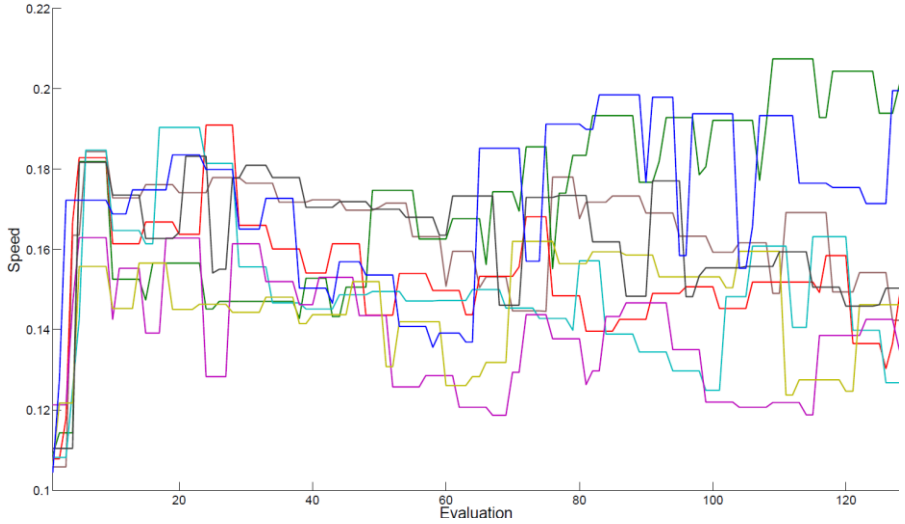
http://youtu.be/qDPbXvADyio
http://youtu.be/MSwdmC0dZ74

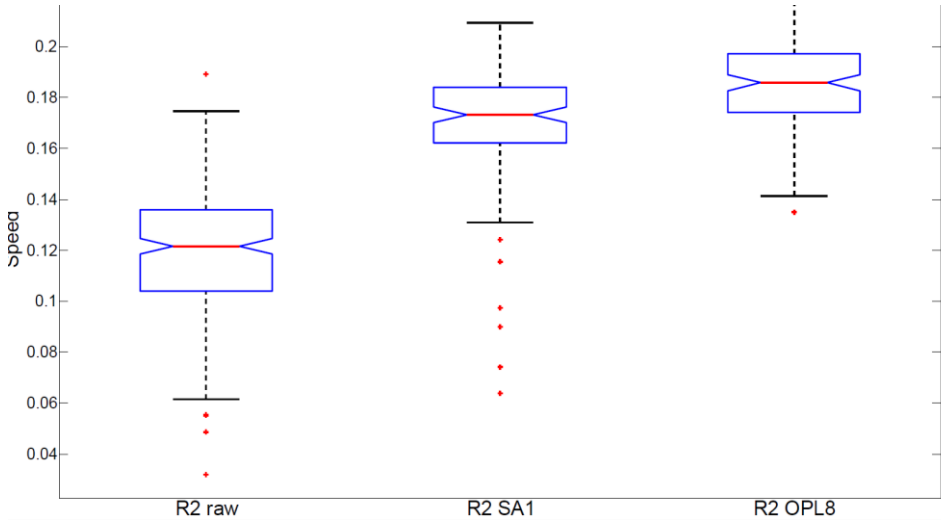
# 2. Transferability (UPMC, Paris)



# 3. Adaptation after transferral (VIDEO)

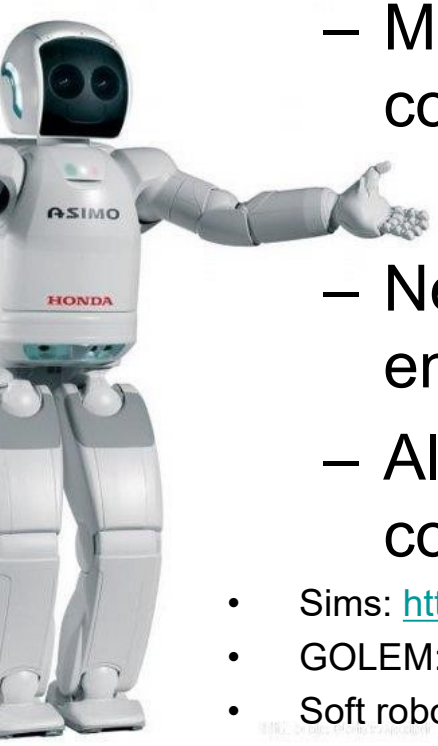
- Reality gap is «accepted»
- Adaptation algorithm is carried out on the real robot
- Needs to take into account lower number of tests and more noise

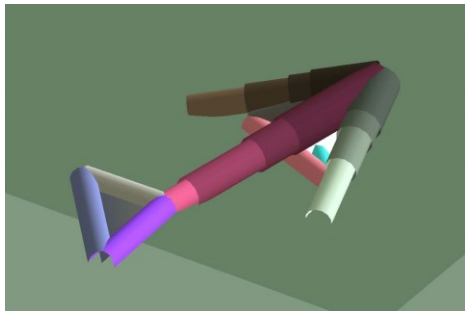


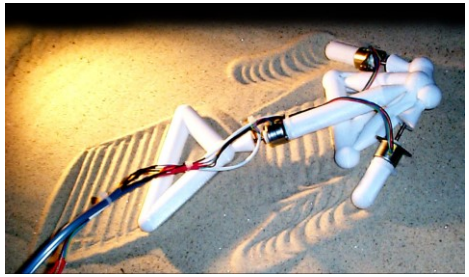



# Evolving shape and control

- Physics simulation allows evolution of shape and control simultaneously
  - More efficient designs for complex problems?



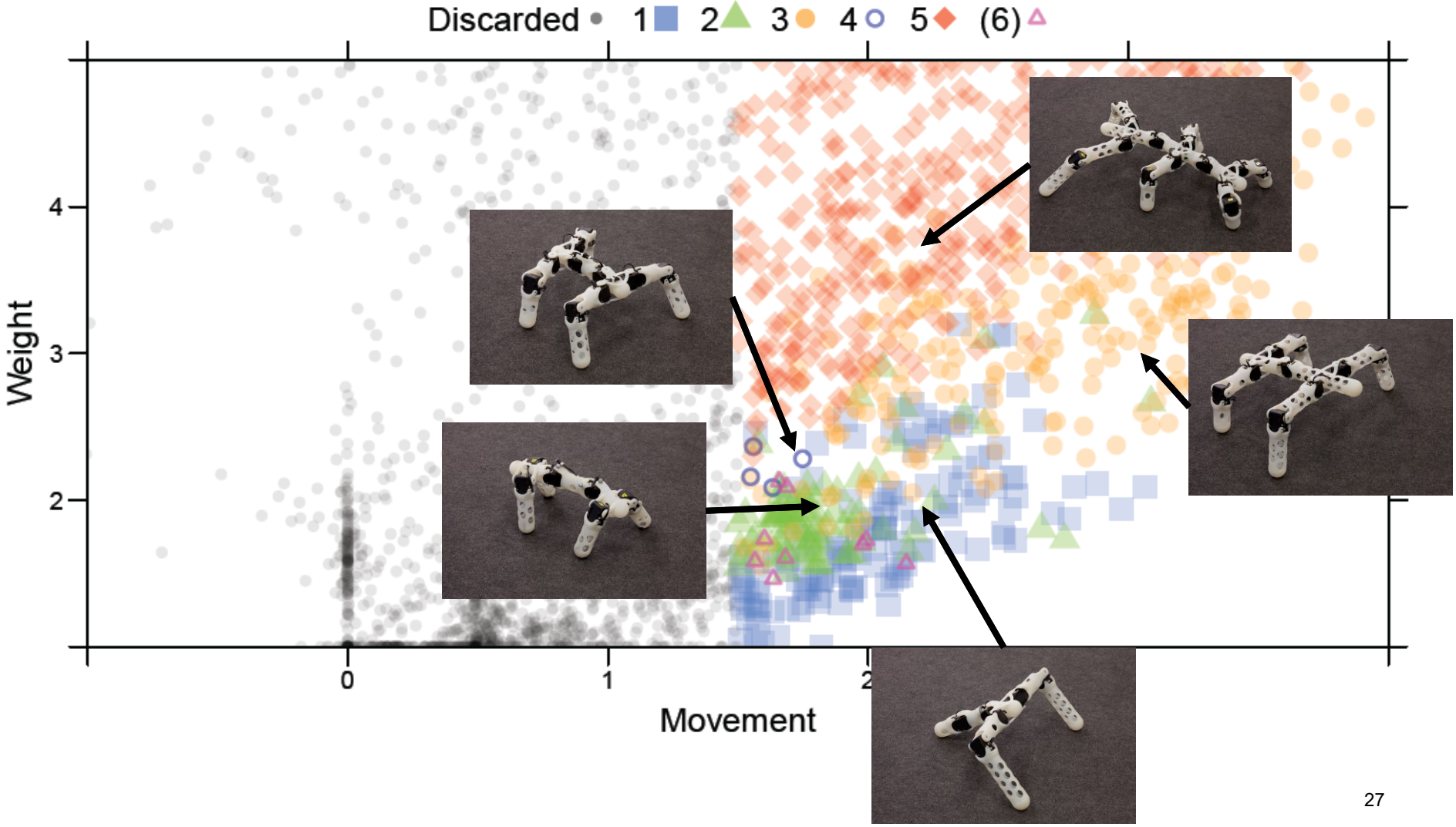





  - New designs for new environments?
  - Allows for offloading computation to the body?

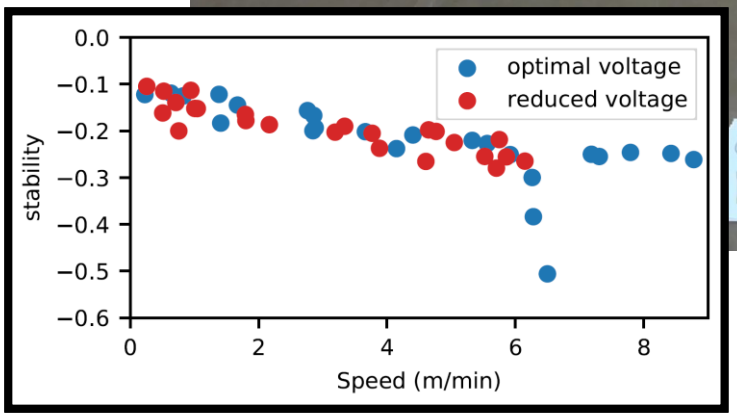
- Sims: http://youtu.be/JBgG_VSP7f8
- GOLEM: http://youtu.be/sLtXXFw_q8c
- Soft robot: http://youtu.be/z9ptOeByLA4

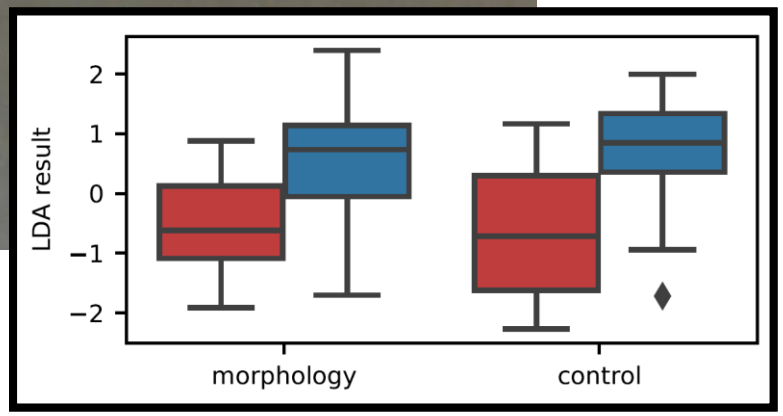
# Evolving different body types



# Evolving shape and control for DyRET

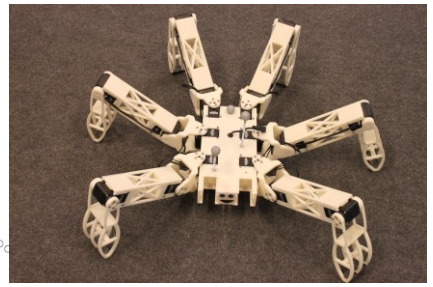


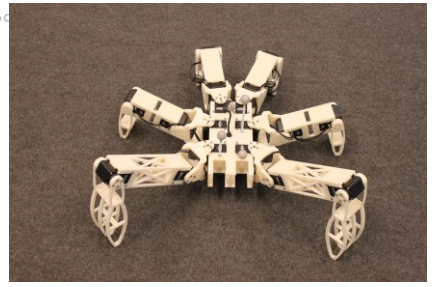


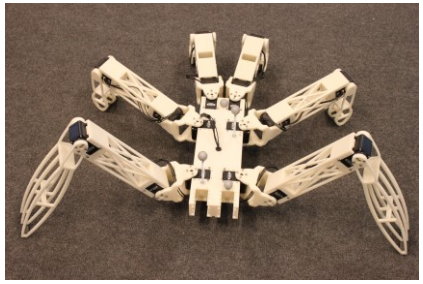



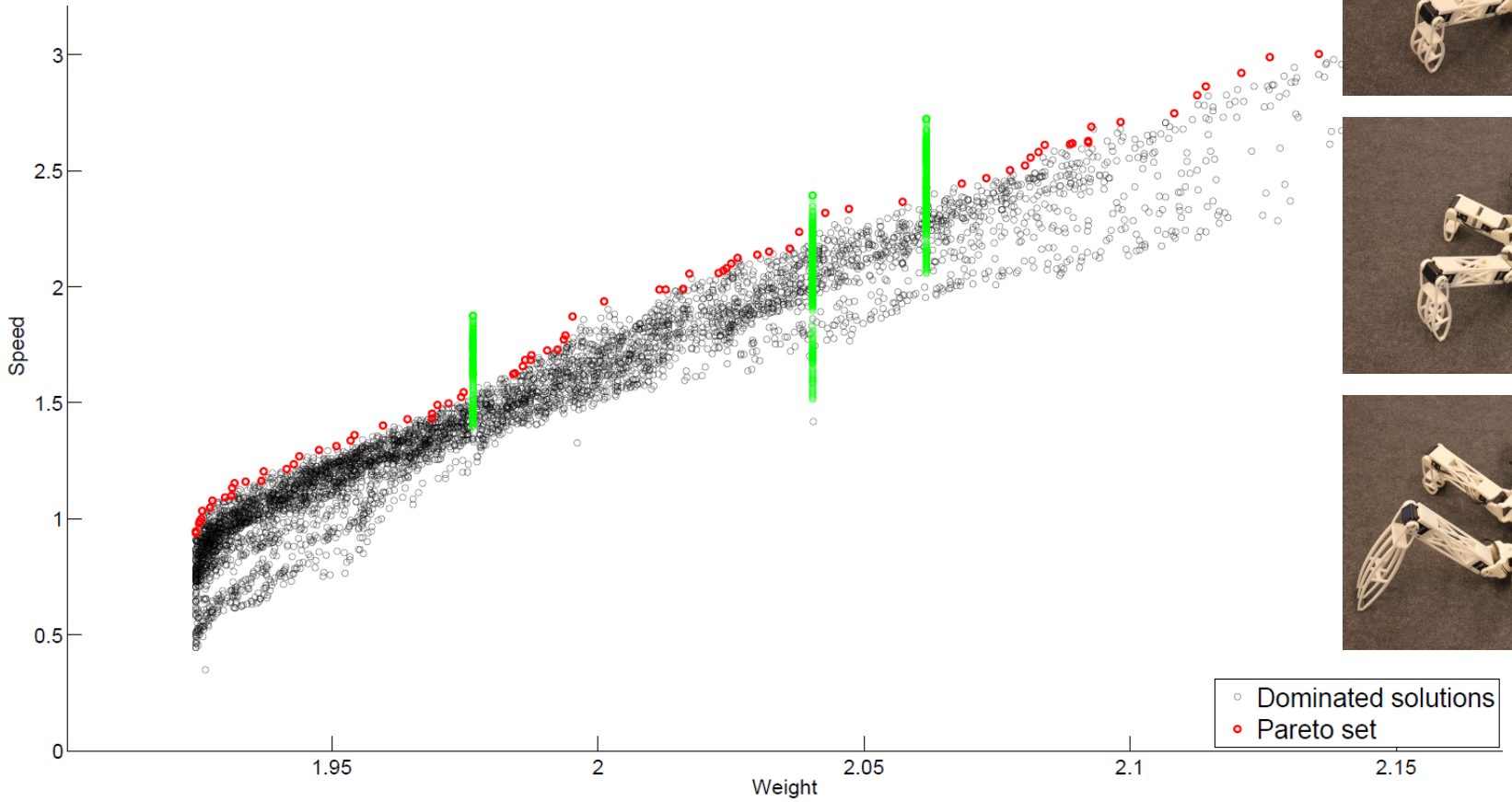
# Example MSc project: Karkinos

- Hybrid automatic / engineered design of robot shape and control









# 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