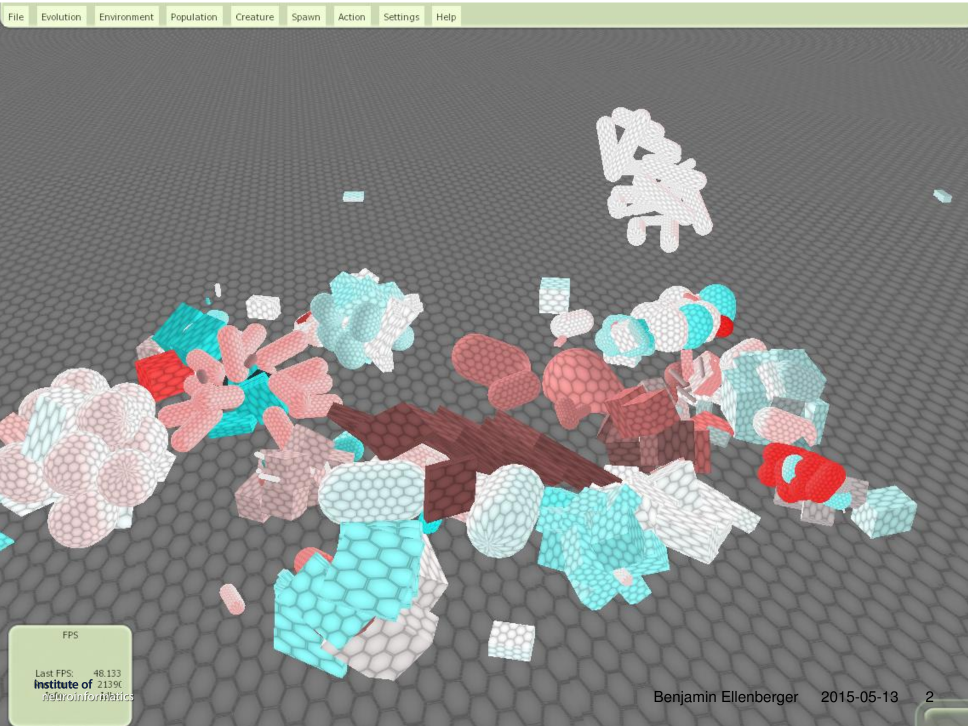
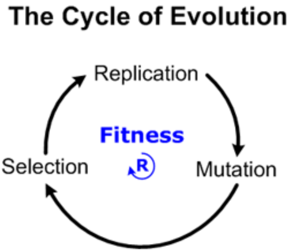
ETH zürich

How Evolution _might_ make creatures walk
A multistage optimization problem

Benjamin Ellenberger

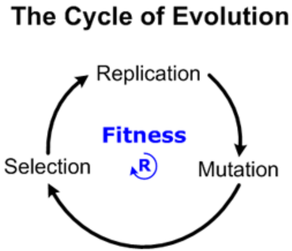

File
Evolution
Environment
Population
Creature
Spawn
Action
Settings
Help
FPS
Last FPS: 48.133

How does evolution optimize?



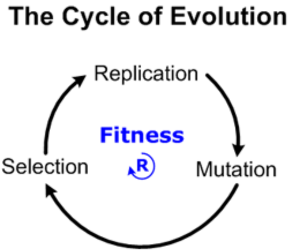
How does evolution optimize?

- Variation-Selection-Replication cycle



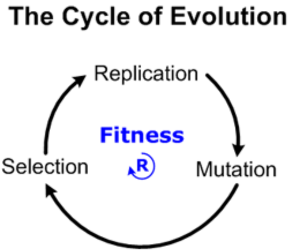
How does evolution optimize?

- Variation-Selection-Replication cycle
- Indirect, developmental, messy encoding (redundant, prone to sudden variation)



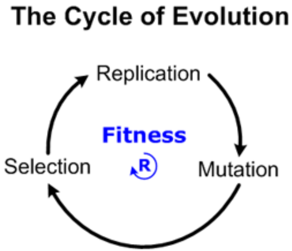
How does evolution optimize?

- Variation-Selection-Replication cycle
- Indirect, developmental, messy encoding (redundant, prone to sudden variation)
- Embryogenesis/ Genotype (DNA) to Phenotype (Animal body) transcription



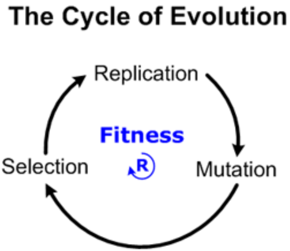
How does evolution optimize?

- Variation-Selection-Replication cycle
- Indirect, developmental, messy encoding (redundant, prone to sudden variation)
- Embryogenesis/ Genotype (DNA) to Phenotype (Animal body) transcription
- Development of animal after birth (Metamorphosis/ Growth etc.)



How does evolution optimize?

- Variation-Selection-Replication cycle
- Indirect, developmental, messy encoding (redundant, prone to sudden variation)
- Embryogenesis/ Genotype (DNA) to Phenotype (Animal body) transcription
- Development of animal after birth (Metamorphosis/ Growth etc.)
- Learn to adapt
- Nature vs. nurture debate



Contents

Evolving Virtual Creatures

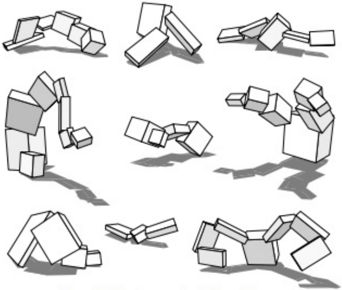


Figure 7: Creatures evolved for walking.

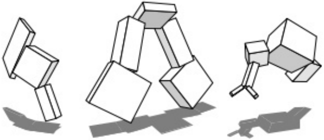


Figure 8: Creatures evolved for jumping.

1 Image from Sims K. - Evolving Virtual Creatures (1994)

Evolving Virtual Creatures

- Creatures are built from 3D Primitives and Joints

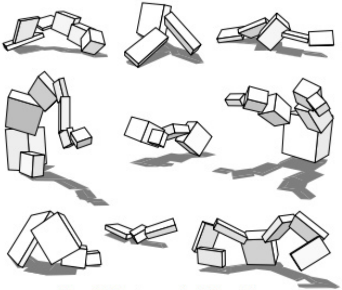


Figure 7: Creatures evolved for walking.

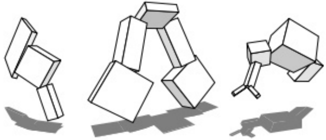


Figure 8: Creatures evolved for jumping.

[1] Image from Sims K. - Evolving Virtual Creatures (1994)

Evolving Virtual Creatures

- Creatures are built from 3D Primitives and Joints
- Sensors, Controller and Effectors make it move

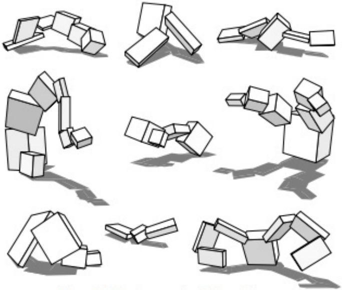


Figure 7: Creatures evolved for walking.

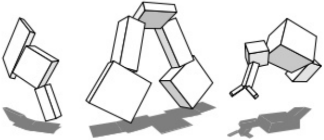


Figure 8: Creatures evolved for jumping.

[1] Image from Sims K. - Evolving Virtual Creatures (1994)

Evolving Virtual Creatures

- Creatures are built from 3D Primitives and Joints
- Sensors, Controller and Effectors make it move
- Body and controller co-evolved

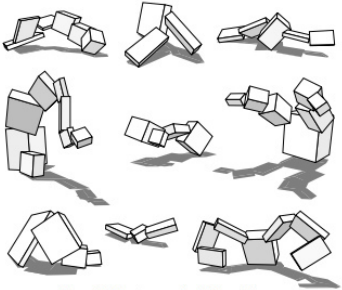


Figure 7: Creatures evolved for walking.

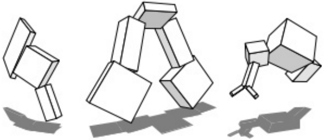
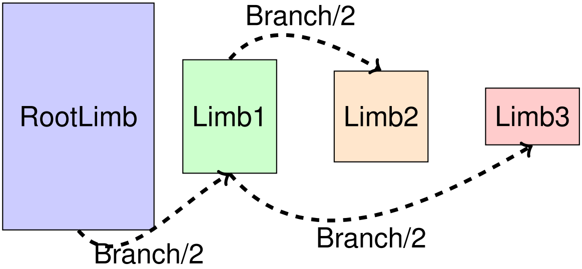


Figure 8: Creatures evolved for jumping.

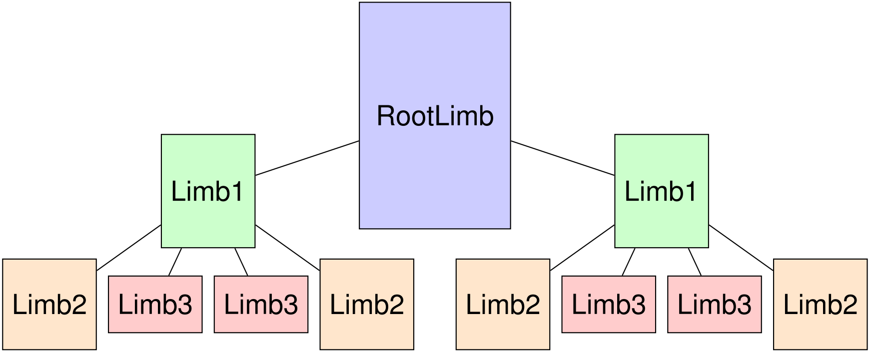
[1] Image from Sims K. - Evolving Virtual Creatures (1994)

Genetic language: Genotype

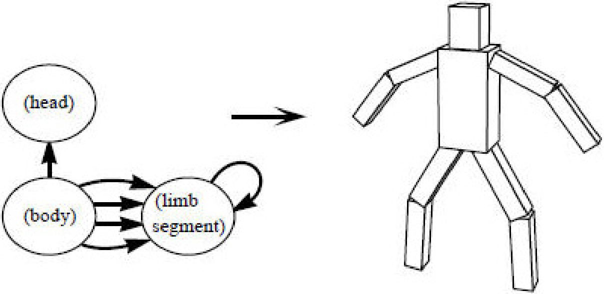


- **Limb** Part of creature body

Genetic language: Phenotype

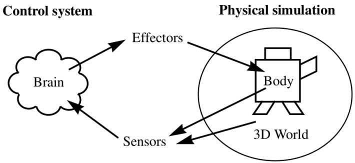


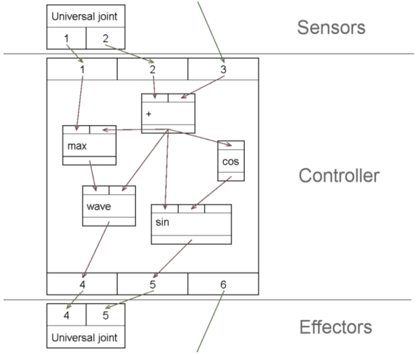
Genetic language: Phenotype cont.



[1] Image from Sims K. - Evolving Virtual Creatures (1994)

Execution of creatures





[1] Image from Sims K. - Evolving Virtual Creatures (1994)

Evolution

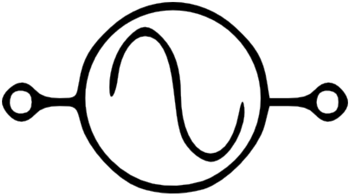
- Selection: Only a certain percentage of creatures are selected for new generation
- Cross-over: Only certain percentage of creatures are allowed to breed
- Mutation
  - Other creatures are subject to mutation
  - Mutation of gene
  - Mutation of gene attributes
  - Mutation of gene branches
- Successful creatures stay in the population and the population is refilled with newly bred and mutated ones

Contents

Contents

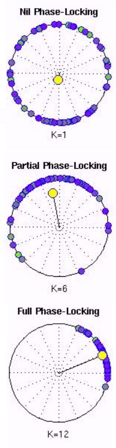
Oscillator controller

- Sine-wave controller taking frequency, amplitude, X-shift,Y-shift as input which are determined evolutionarily.



Kuramoto model

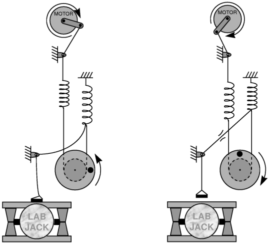
- A model proposed by Yoshiki Kuramoto[1] to describe synchronization.
- $\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N} \sum_{j=1}^{N} \sin(\theta_j - \theta_i), \qquad i = 1 \ldots N$



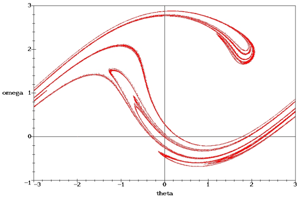
[1] Kuramoto Y. (1984). Chemical Oscillations, Waves, and Turbulence. New York, NY: Springer-Verlag

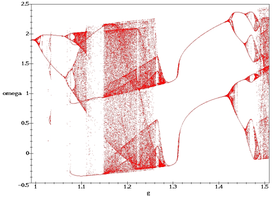
Simplifying control using simple limiters[2]

- Generally the chaos controller is more complex than the system it controls



(a) (b)

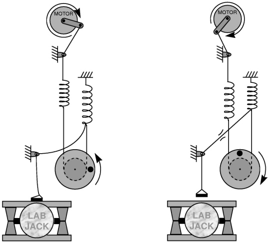




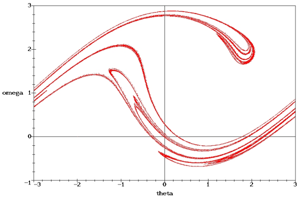
[2] Corron N. et al. - Controlling Chaos with Simple Limiters (2000)

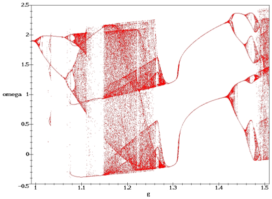
Simplifying control using simple limiters[2]

- Generally the chaos controller is more complex than the system it controls
- Not true for the simple limiter



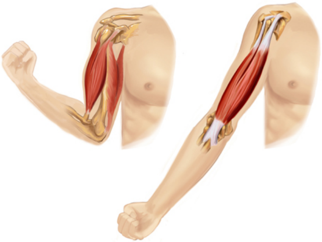
(a) (b)





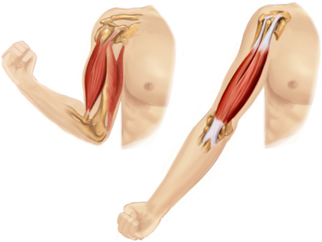
[2] Corron N. et al. - Controlling Chaos with Simple Limiters (2000)

Simple limiters in nature



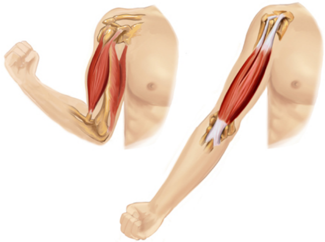
Simple limiters in nature

- Muscle length & joint limits



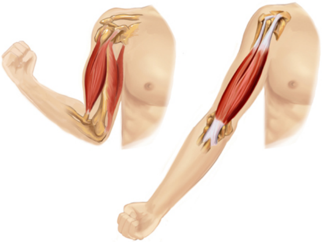
Simple limiters in nature

- Muscle length & joint limits
- The weight of the limbs



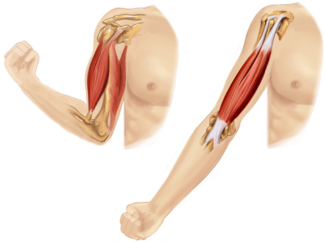
Simple limiters in nature

- Muscle length & joint limits
- The weight of the limbs
- The relative position of limbs connected by joints
  (Direction of force applied to joints)

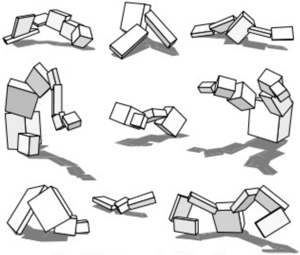


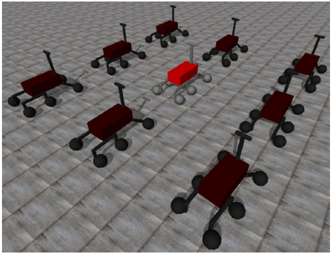
Simple limiters in nature

- Muscle length & joint limits
- The weight of the limbs
- The relative position of limbs connected by joints (Direction of force applied to joints)
- The fact that physical objects can not interpenetrate each other



In silico[3]: Will the simple limiters be used?

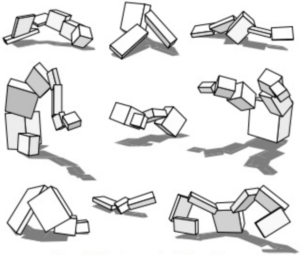


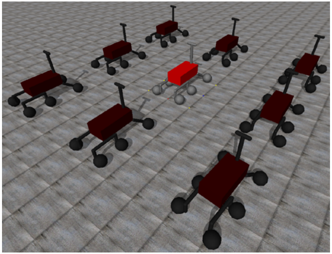


[3] In silico = in simulation

In silico[3]: Will the simple limiters be used?

- Hypothesis 1: Simple limiters are an omnipresent feature used to reduce the control complexity and to control chaotic movement

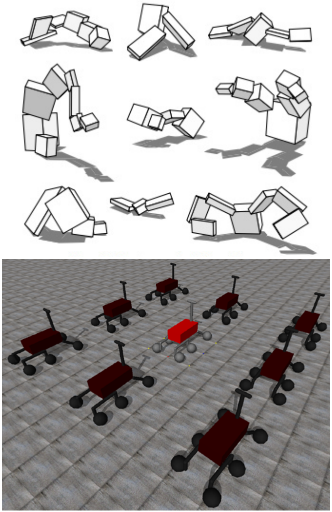




[3] In silico = in simulation

In silico[3]: Will the simple limiters be used?

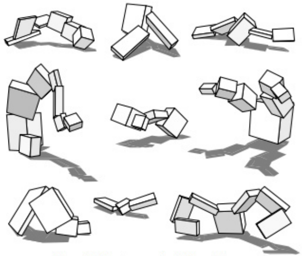
- Hypothesis 1: Simple limiters are an omnipresent feature used to reduce the control complexity and to control chaotic movement
- Hypothesis 2: If the world is flat then less simple limiters are used, if the world is dynamic and bumpy then more simple limiters are used.

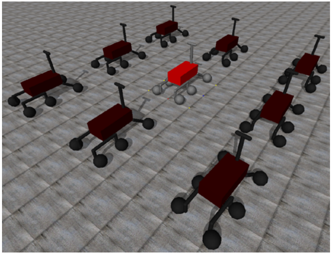


[3] In silico = in simulation

In silico[3]: Will the simple limiters be used?

- Hypothesis 1: Simple limiters are an omnipresent feature used to reduce the control complexity and to control chaotic movement
- Hypothesis 2: If the world is flat then less simple limiters are used, if the world is dynamic and bumpy then more simple limiters are used.
- Question remains: Can you live limitless?

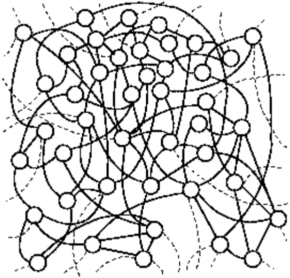




[3] In silico = in simulation

Reinforcement Learning and Neural network controller

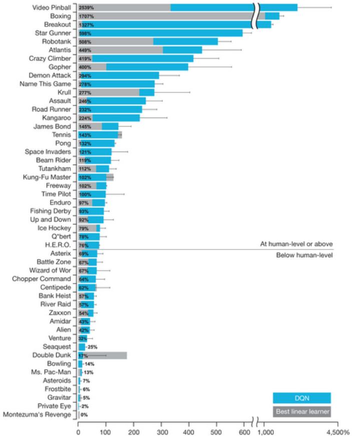
- Reinforcement learning[4] to make a particular creature learn how to walk
- Actor & Critic Architecture of two Neural Networks



[3] Sutton, R. & Barto, A. Reinforcement Learning: An Introduction (MIT Press, 1998)

Google Deepmind [5]: Reinforcement Learning on Atari Games

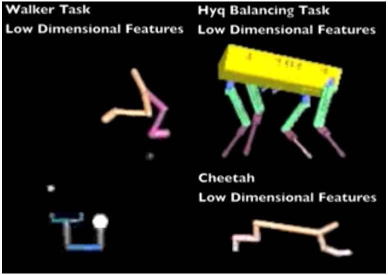
- can learn successful play policies directly from high-dimensional sensory inputs using end-to-end reinforcement learning
- Only for discrete set of actions



[4] V. Mnih et al., Human-level control through deep reinforcement learning, Nature 518, 529–533 (26 February 2015)

Google Deepmind [6]: Continuous control with deep reinforcement learning

- can learn successful policies directly from high-dimensional sensory inputs using end-to-end reinforcement learning
- can solve problems such as cartpole swing-up, dexterous manipulation, legged locomotion
- actor-critic algorithm that can operate over continuous action spaces



[4] T.P. Lillicrap et al., Continuous control with deep reinforcement learning, http://arxiv.org/abs/1509.02971 Stat. ML. (9 Sep 2015)

Discussion!

- Any questions?
- What experiments do you have in mind?
- What else would you change, extend, enhance, improve etc.?
- If you have any ideas later, email me: be.ellenberger@gmail.com
- You can look at my progress: https://github.com/benelot/minemonics

References

- Sims K. - Evolving Virtual Creatures (1994)
- Sims K. - Evolving 3D Morphology and Behavior by Competition (1994)
- Krcah P. - Evolving Virtual Creatures Revisited (2007)
- Corron N. et al. - Controlling Chaos with Simple Limiters (2000)
- Schmidt N. - Bootstrapping perception using information theory: case studies in a quadruped running robot running on different grounds (2013)
- Stoop R. - Theory and Simulation of Neural Networks (2014)