Swiss Federal Institute
of Technology Lausanne

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Evolutionary Robotics
A Short Tutorial

Dario.Floreano@epfl.ch
asl.epfl.ch

Evolutionary Robotics

Evolutionary Robotics is automatic generation of control systems and morphologies of autonomous robots. It is based on a process of *Artificial Evolution* without human intervention.

Two motivations:

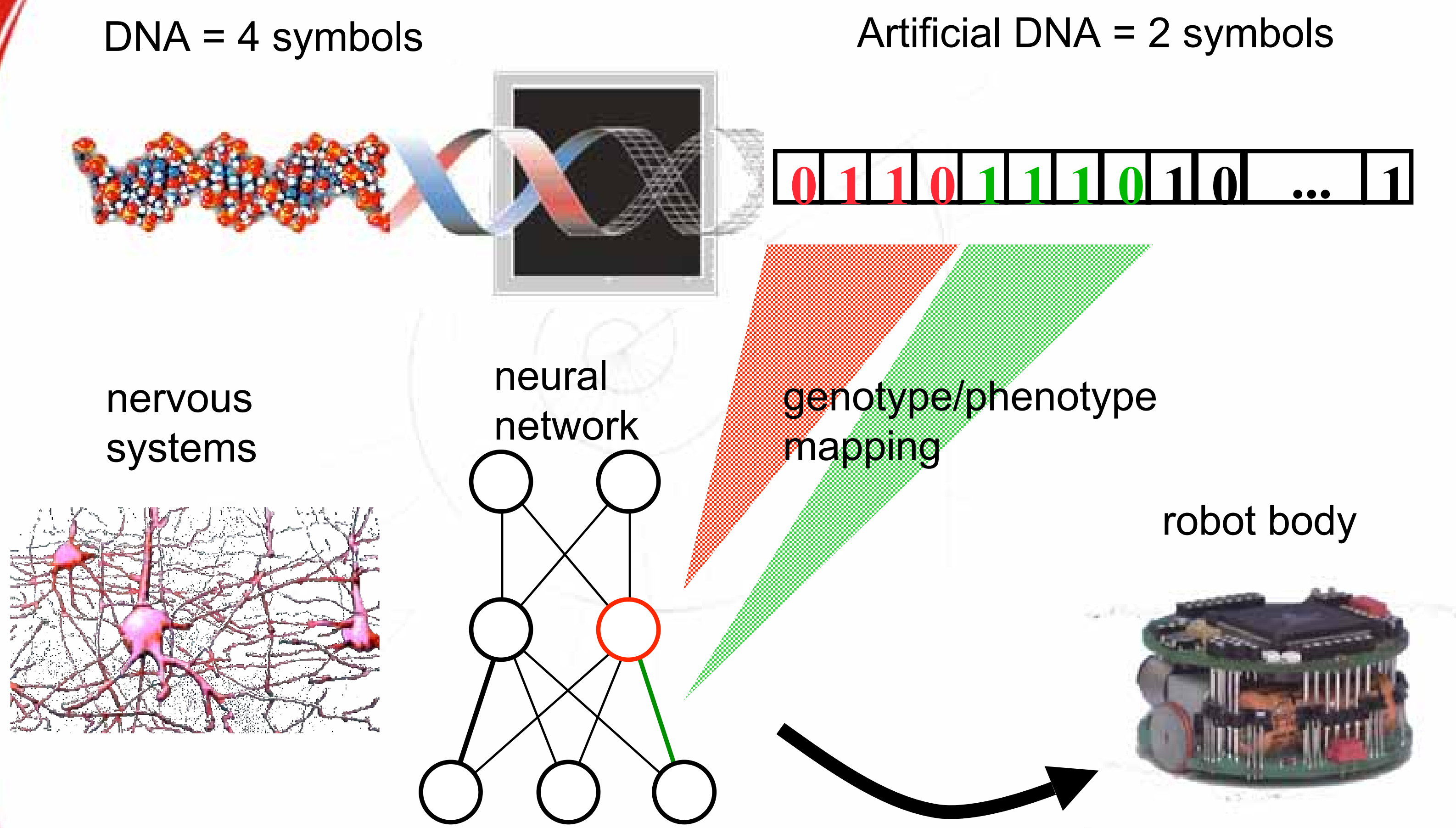
- It is difficult to design autonomous systems using a purely top-down engineering process because the interaction between the robot and its environment is very complex and hard to predict.

  In ER the engineer defines the control components and the selection criterion and lets artificial evolution discover the most suitable combinations while the robot interacts with the environment.

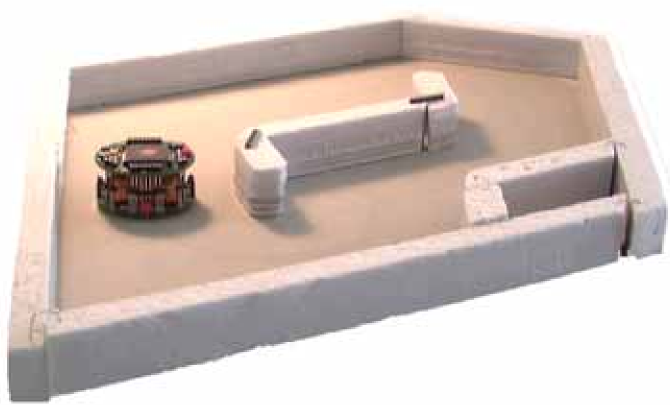
- A *synthetic* (as opposed to an *analytic*) approach to the study of the mechanisms of adaptive behavior in machines and animals.

  ER was first suggested by a neurophysiologist (Braitenberg, 1984) as a way to show that evolution can generate simple artificial neural circuits that display apparently complex behaviors.

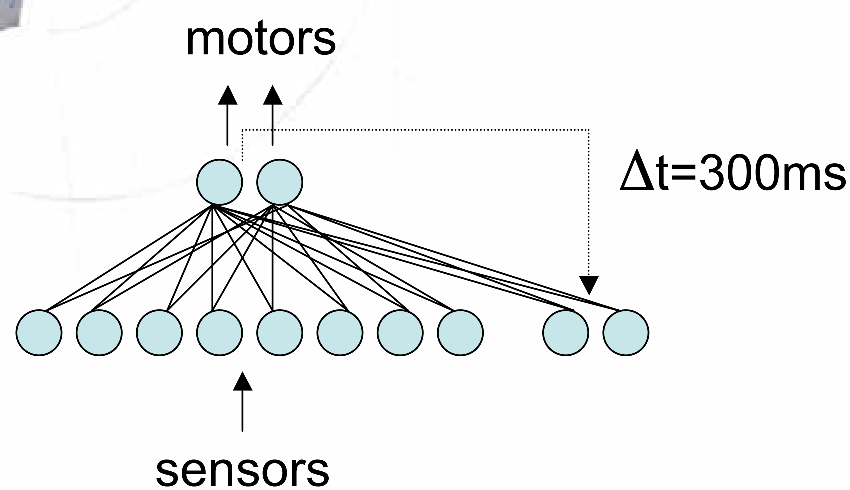
Robot Biology



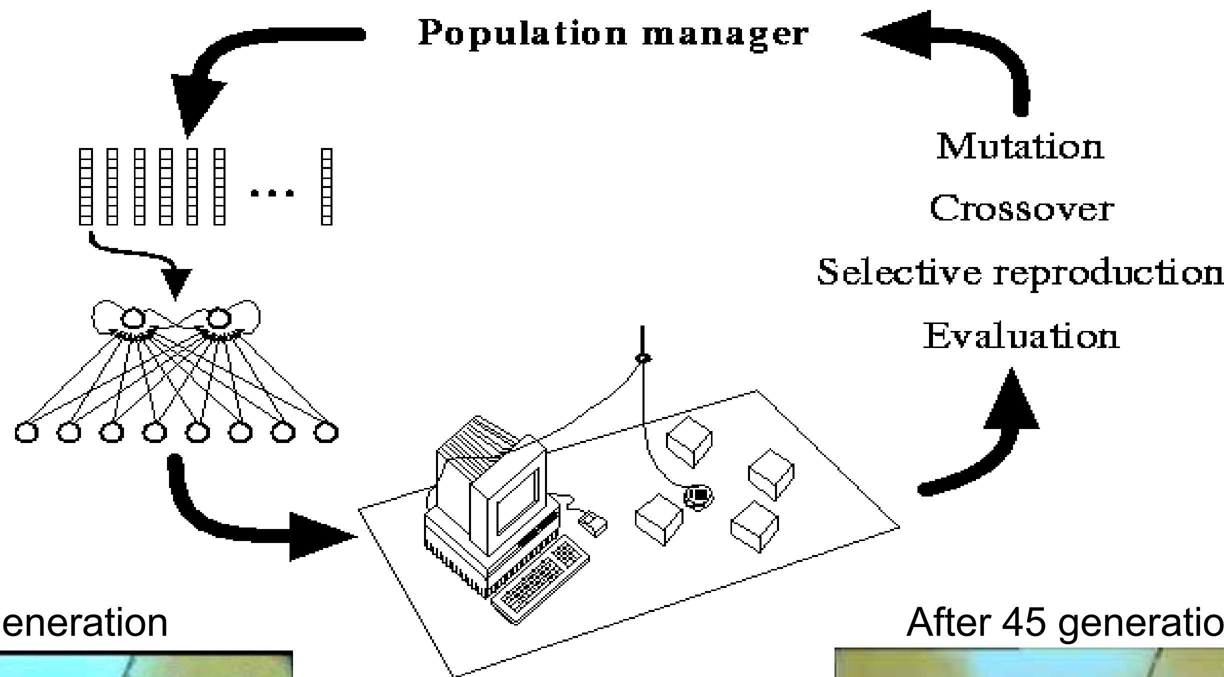
Simple Navigation



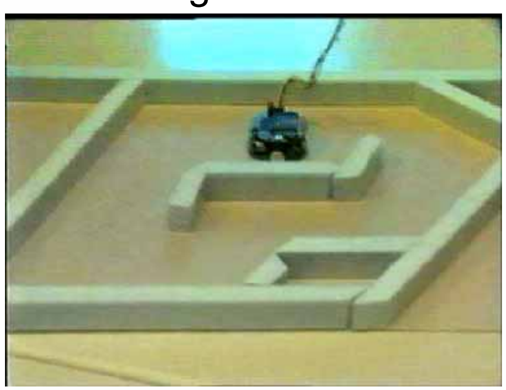
Fitness = V x Δv x (1-s)

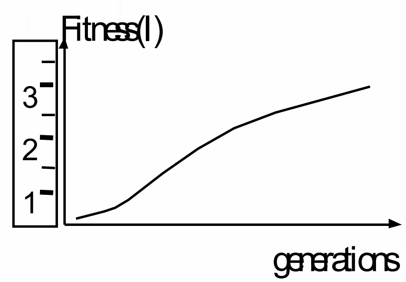


Method



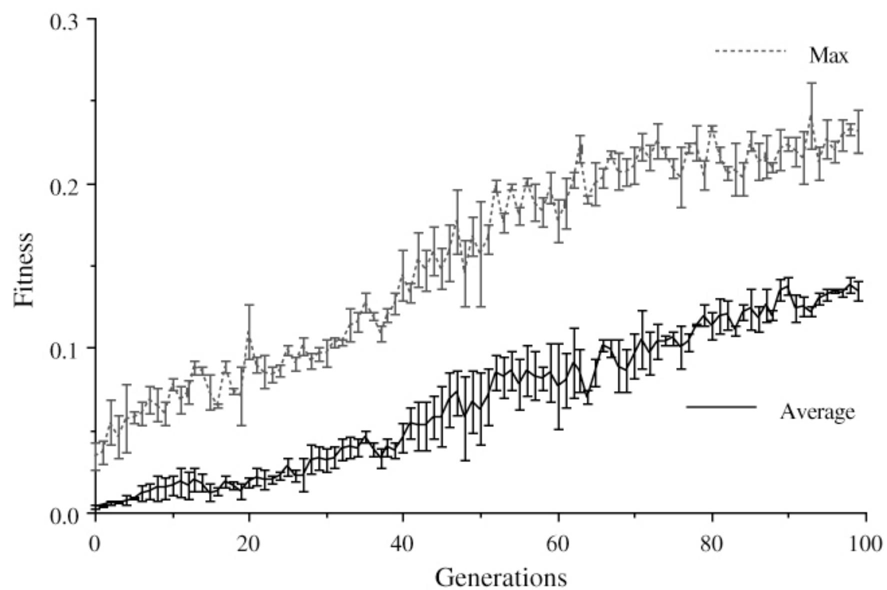
Initial generation

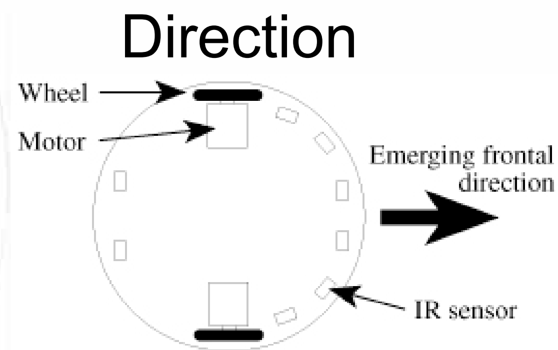




After 45 generations

Results





Speed = 60%

The average and best population fitness are typical measures of performance.

Evolved robots always have a preferential direction of motion and speed.

Fitness Functions

Design of fitness functions that can generate desired behaviors is one of the most difficult parts of Evolutionary Robotics. Very often the evolved robot will maximize the fitness criterion using very simple behaviors.

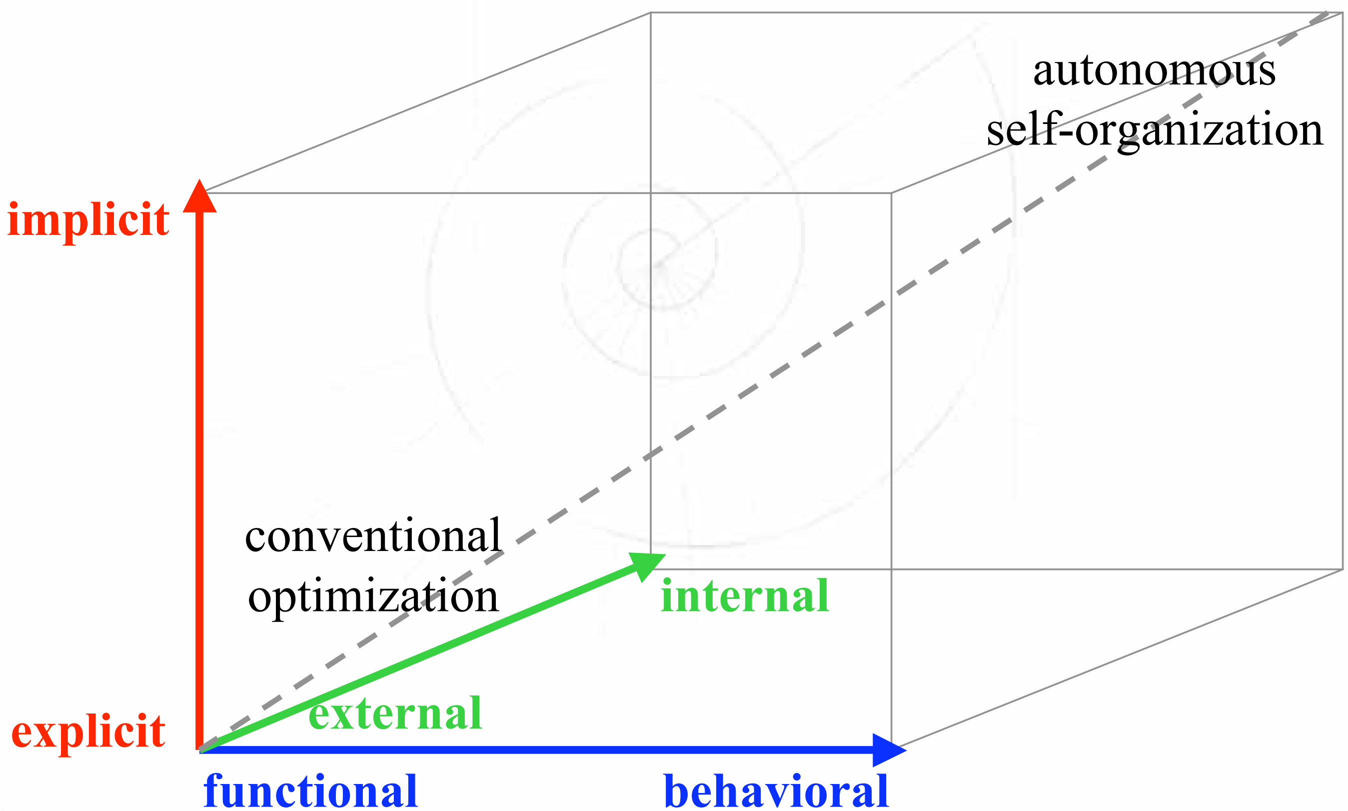
Such evolved solutions can be interesting and surprising, but not what the engineer had in mind.

The concept of fitness function is strange for a biologist. In biology the fitness of a species is the amount of individuals in the population (growing populations are fitter than shrinking populations). In genetic algorithms, the fitness is a measure of performance of an individual and a selection criterion.

The choice of a fitness function makes all the difference between an optimization process and autonomous artificial life.

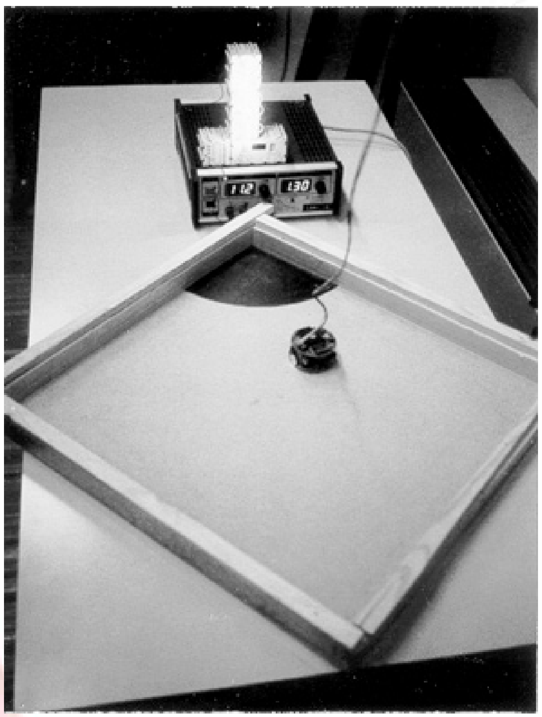
Fitness Space

Fitness Space is a method to conceive and compare fitness functions

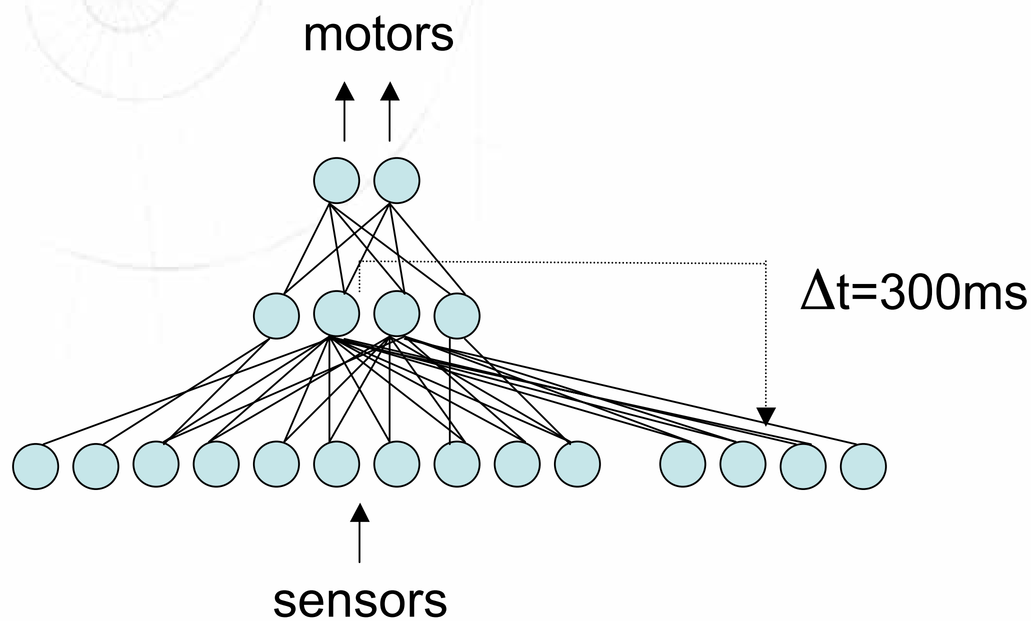


Environment Role

Let us now put the robot in a more complex environment and make the fitness function even simpler. The robot is equipped with a battery that lasts only 20 s and there is a battery charger in the arena.

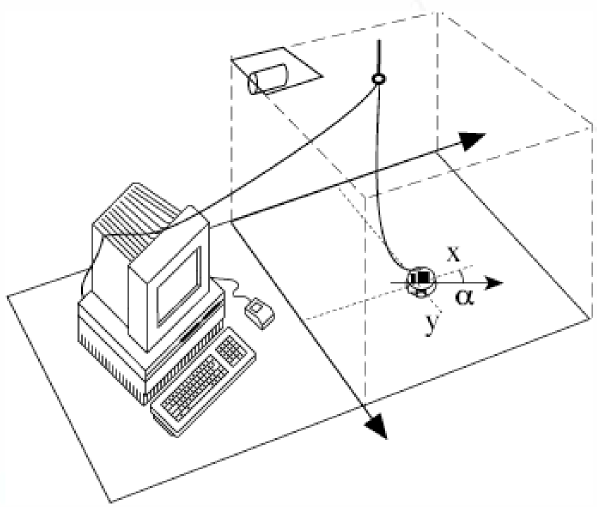


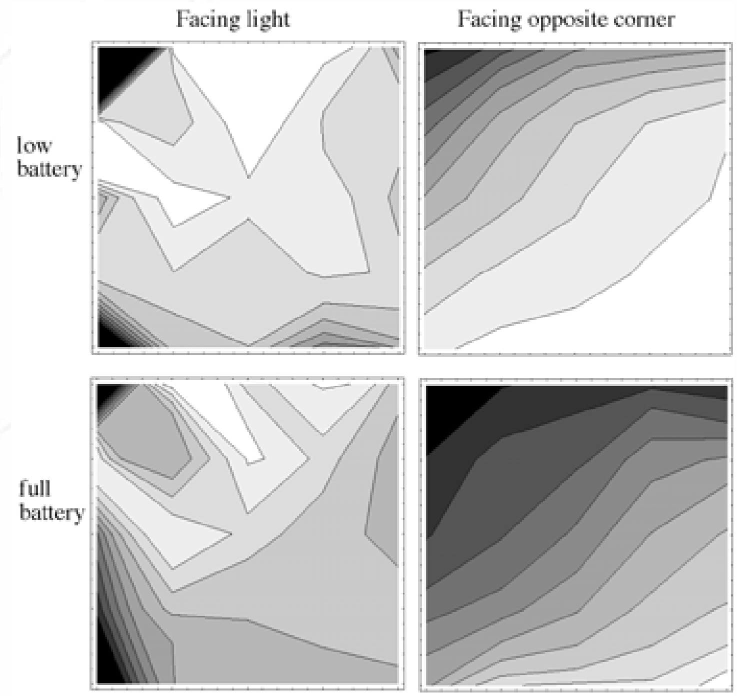
$$\text{Fitness} = V \times (1-s)$$



Machine Neuro-ethology

After 240 generations, we find a robot capable of moving around and going to recharge 2 seconds before the batteries are completely discharged.



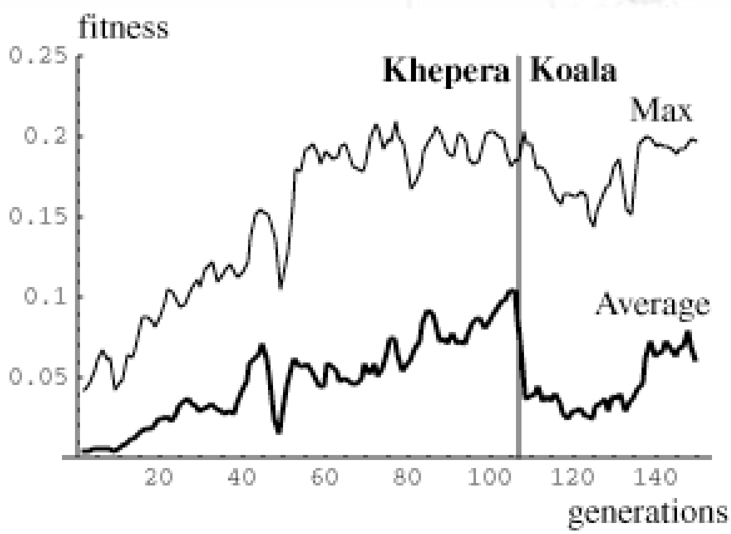


Neural Activity Maps

Fast re-adaptation

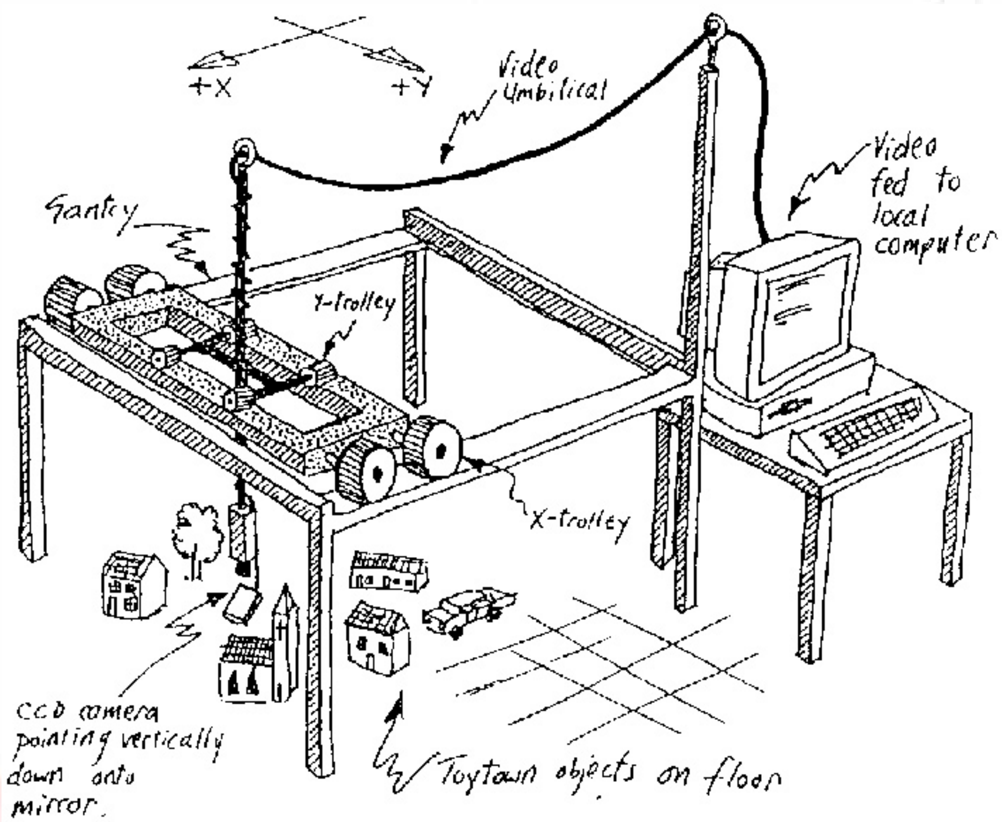
After an initial drop in average and best fitness, the performance goes back to the previous maximum levels much faster than it took to get there with the Khepera.

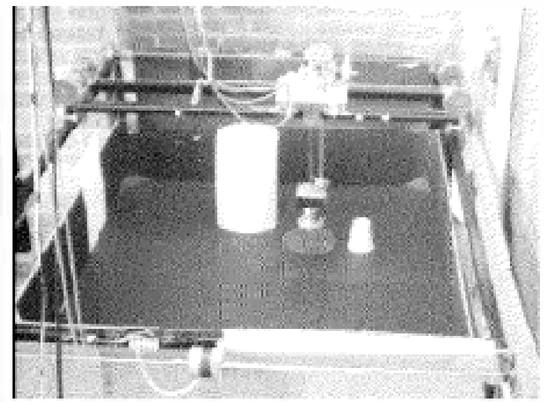
The evolved behavior of the Koala is different from that on the Khepera because the body shape and its relation with the environment is different.

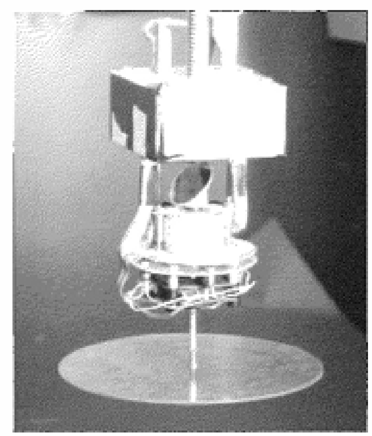


'Seeing the Light'

The Sussex group investigated evolution of vision-based behaviors. They solved the energy fitness problem using a suspended camera with bumpers (*gantry robot*).

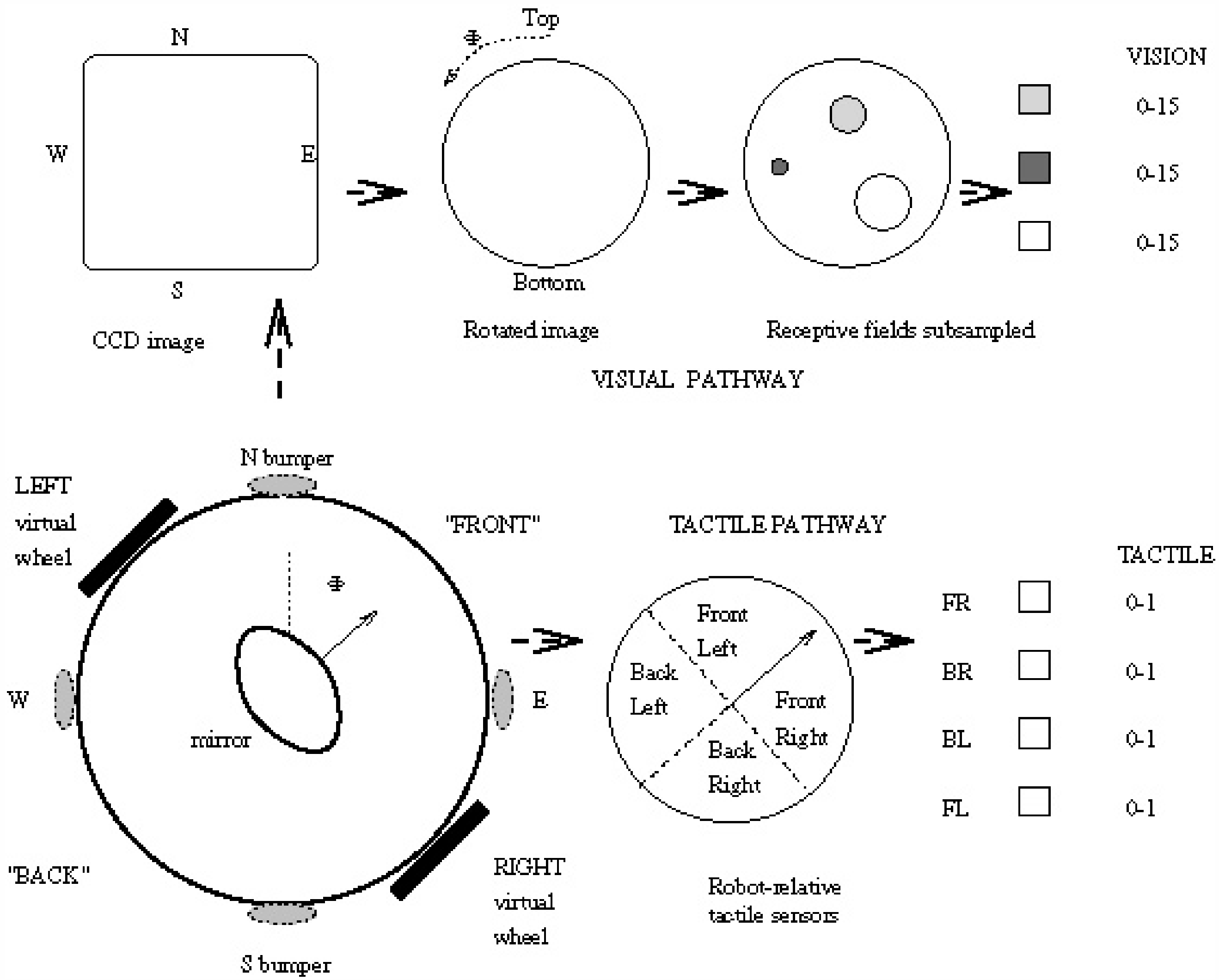






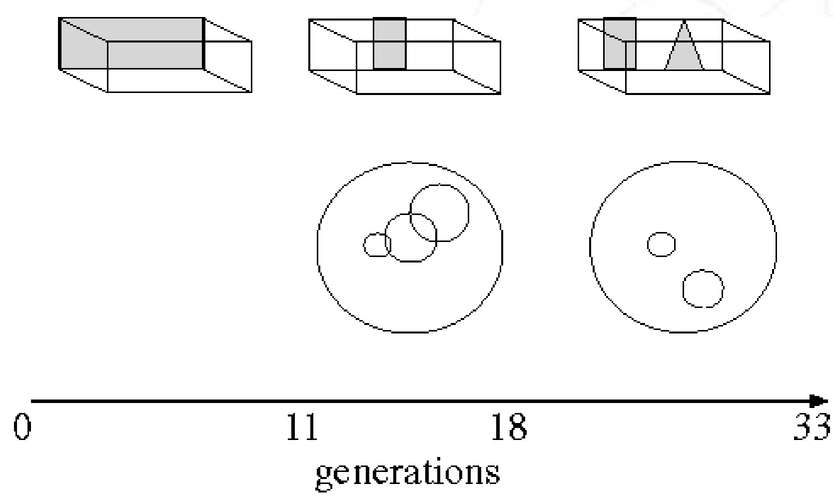
[Harvey et al. 1994]

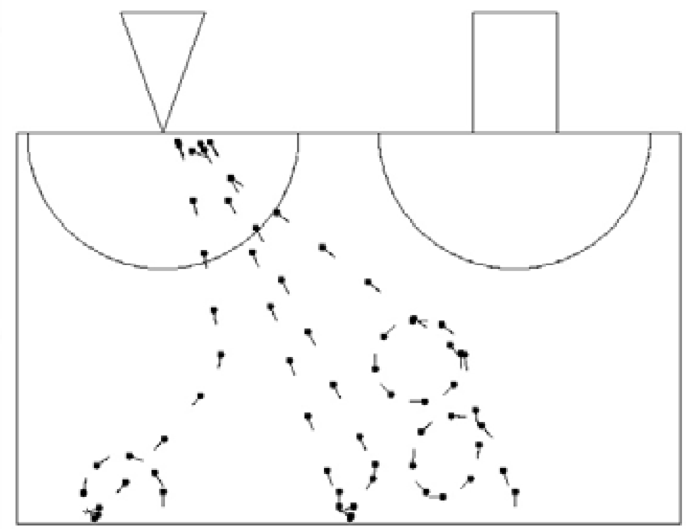
Architecture



Incremental Evolution

In the first stage, one full wall was covered with white paper and the robot was asked to move toward the wall. In the second stage, the white target surface was restricted to a 22 cm wide band. Finally, in the third stage the white paper was substituted by two white shapes, a rectangle and a triangle, and the robot was asked to move toward the triangle.

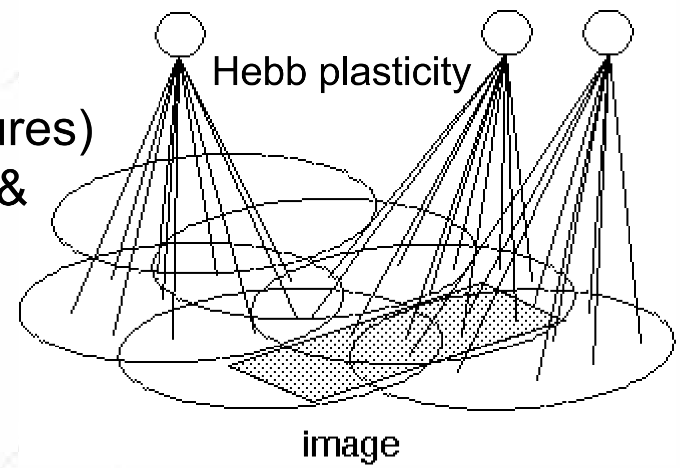




Evolved controllers used only two photoreceptors whose activation time, triggered by the left-wing rotation, was sufficient to discriminate between the two patterns.

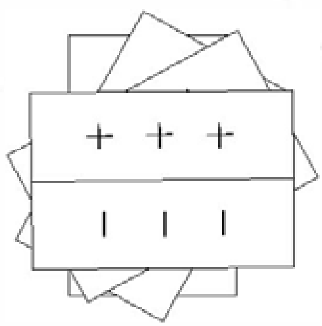
Feature Selection

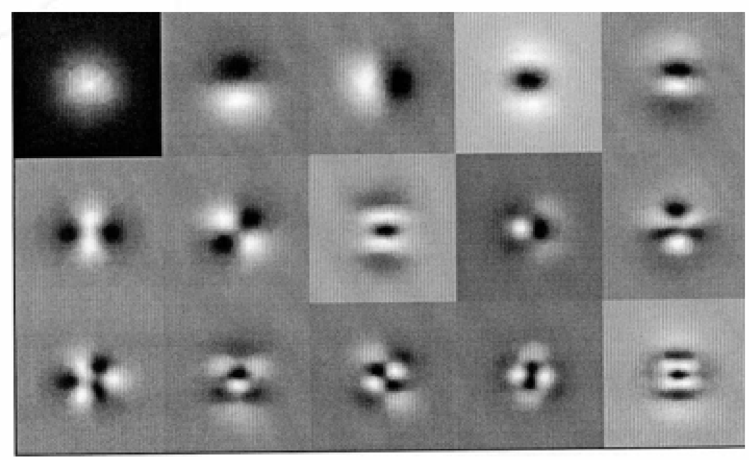
Process whereby visual neurons become sensitive to certain sensory patterns (features) during the developmental process (Hubel & Wiesel, 1959)



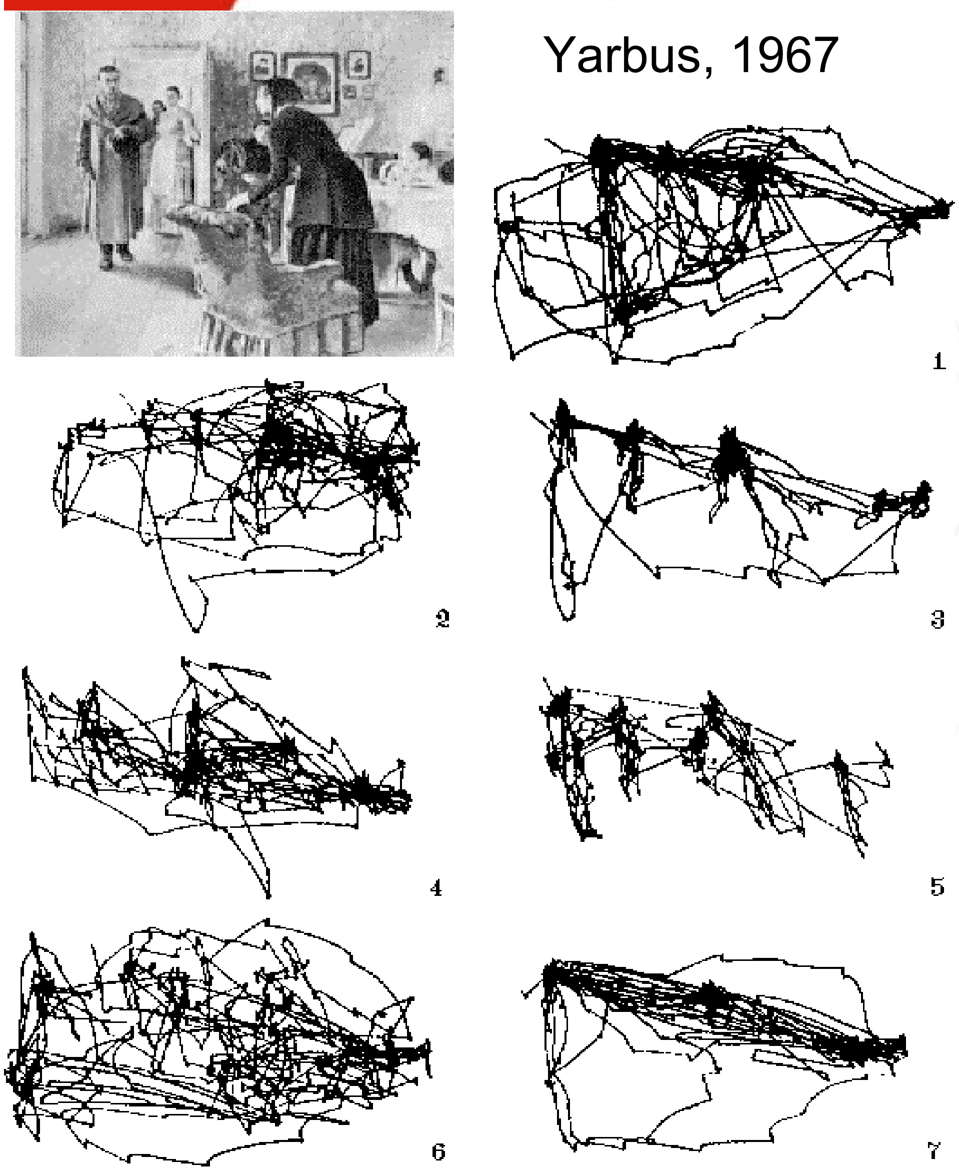
Center-Surround

Oriented Edges



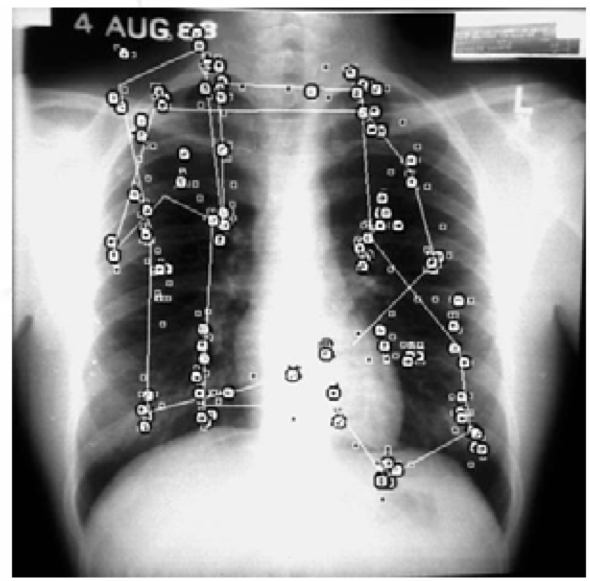


Active Vision



Yarbus, 1967

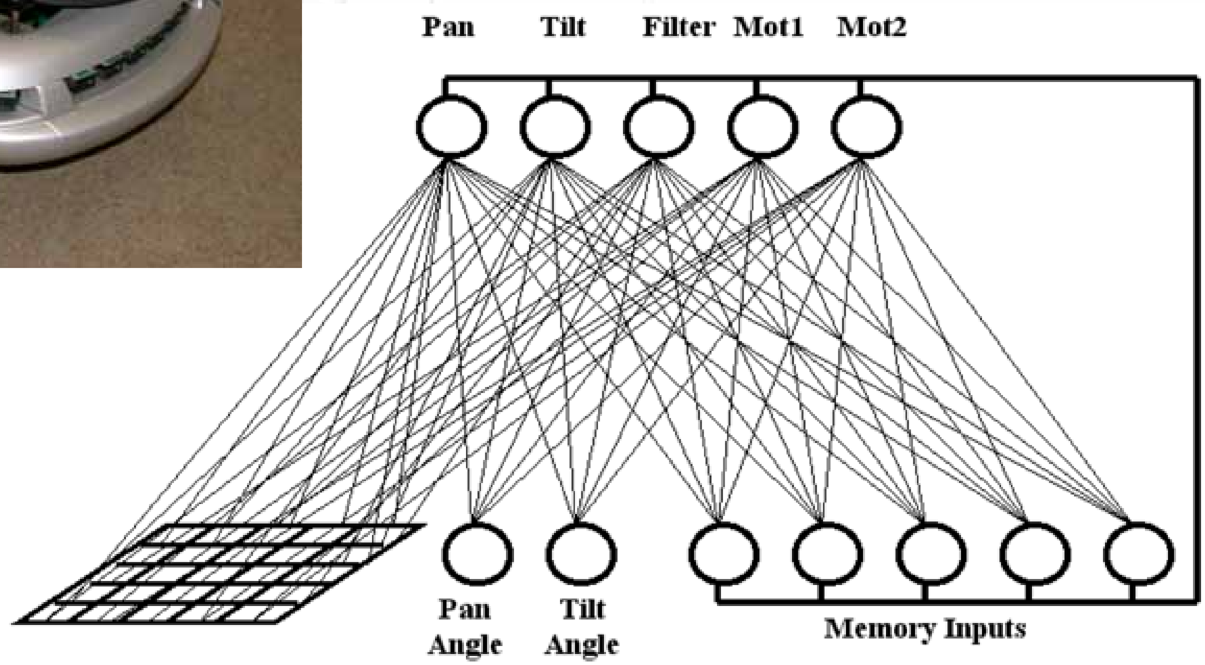
Process of selecting by motor actions sensory patterns (features) that make discrimination easier (Bajcsy, 1988)



Co-evolution F.S. + A.V.

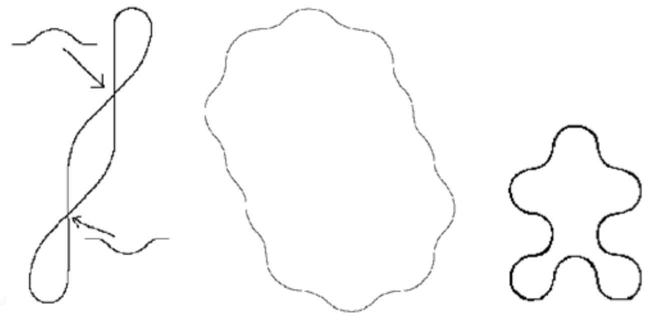
Goal: Robot must move around simple arena using only vision information from a pan/tilt camera.

Output of vision system is movement of camera (pan/tilt) and of robot wheels (mot1/mot2). Filter as before.



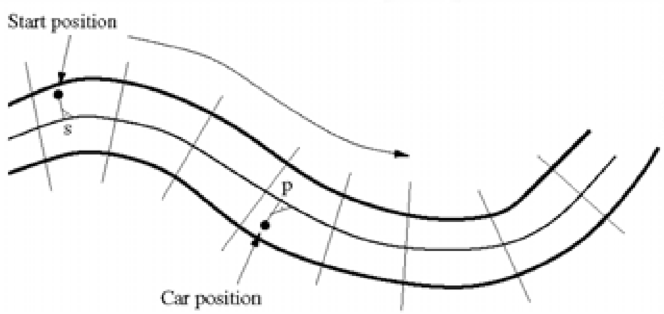

Environment

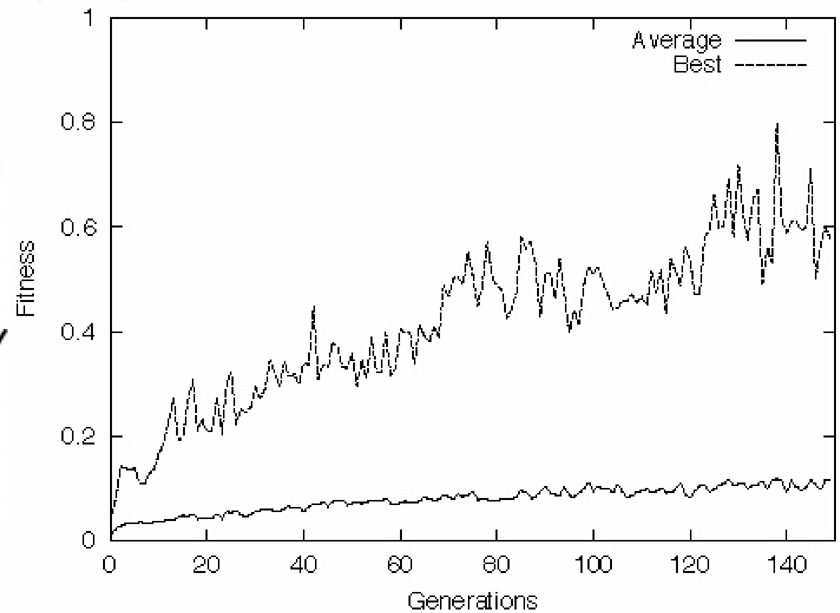
Evolving a Car Driver

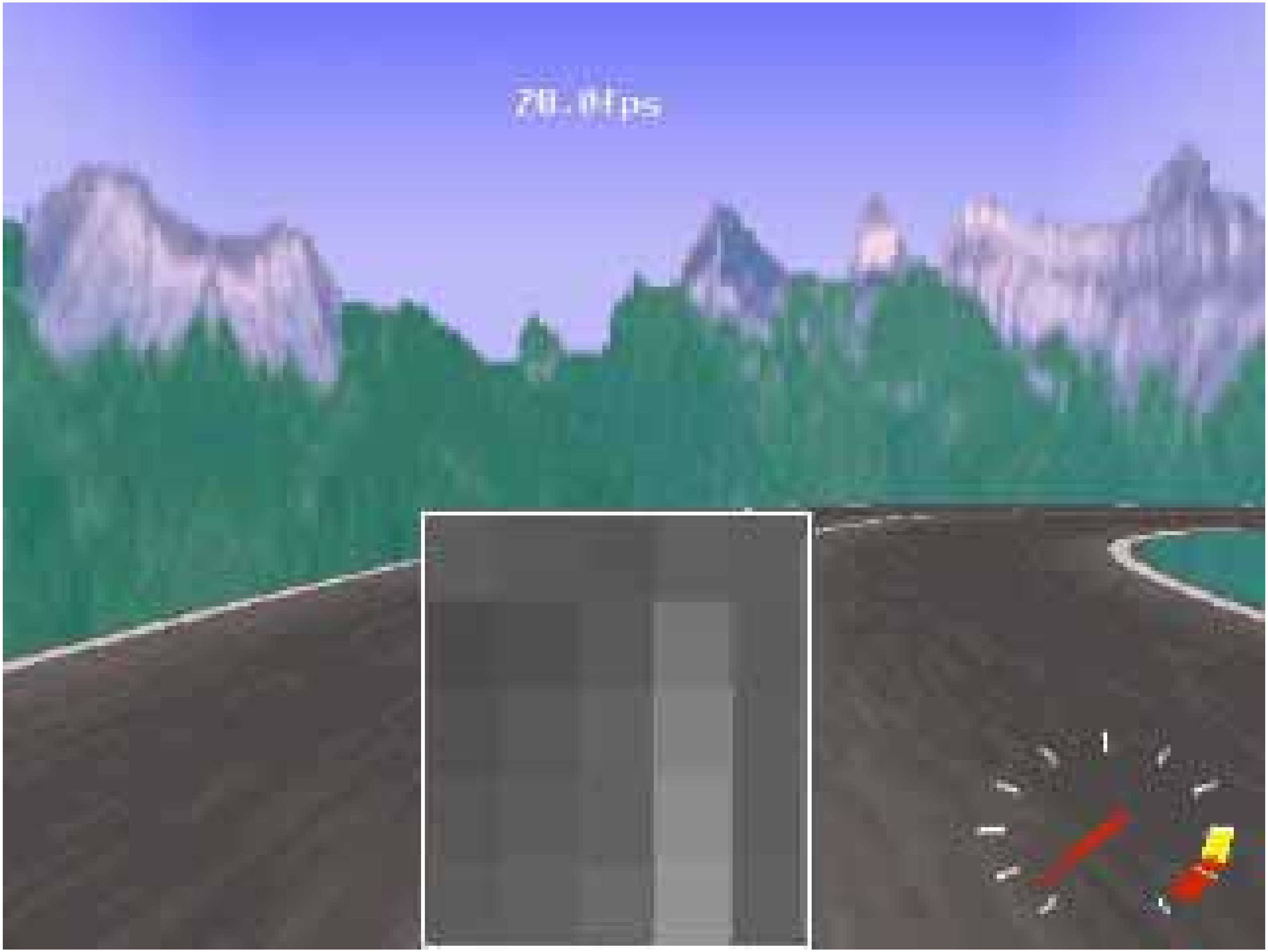


Fitness = percentage of covered distance D in R races on M circuits (limited time for each race).

$$F = \frac{1}{R * M} \sum_{r=1}^{R} \sum_{m=1}^{M} D_{r,m}$$

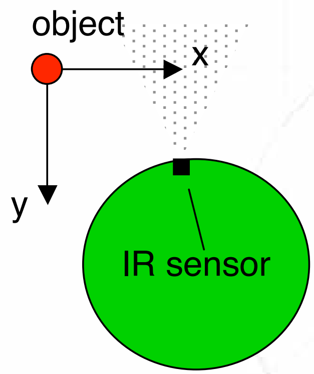




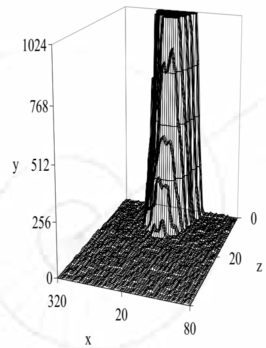

70.0fps

Simulations

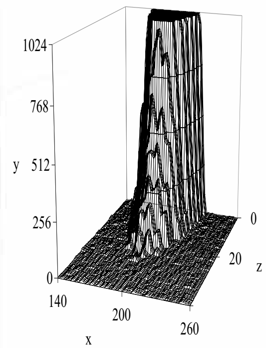
Different physical sensors and actuators may perform differently because of slight differences in their electronics or mechanics.



khepera



4th IF sensor



8th IF sensor

Physical sensors deliver uncertain values and commands to actuators have uncertain effects.

The body of the robot and the environment should be carefully (not accurately) reproduced in the simulation.

Simulation: Noise

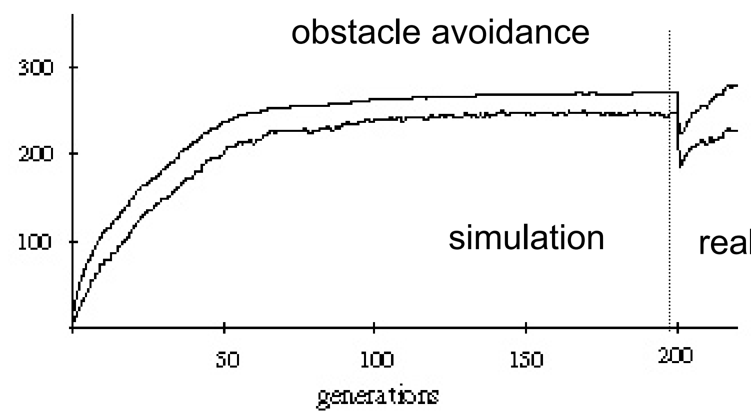
The simplest and most often used way to ensure that simulation results transfer to real robots consists of adding noise from a uniform distribution centered about zero to the precise values produced by analytical models.

Noise can/should be added to:

- computed speeds (kinematic equations)
- cartesian coordinates (trigonometric transformations)
- sensor values (usually linearly monotonic functions)

Typical noise values in the literature are in the range of 5% of the signal

However, this method does not yet guarantee a perfect transfer [Miglino et al., 1995] because the noise in the environment is not uniform.



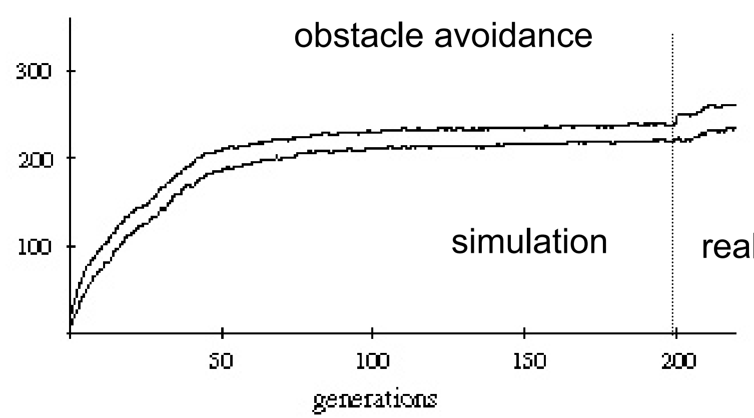
Simulation: Sampling

Sampling consists in measuring the values returned by the robot sensors for given objects and by actuators for given speeds.
The values are stored in a look-up table and accessed by the simulator.
Furthermore, some noise (5%) is added to the values.

distance	angle	sample val
1 mm	0 deg.	0.98
2 mm	0 deg.	0.95
...	...	...
1 mm	2 deg.	0.96

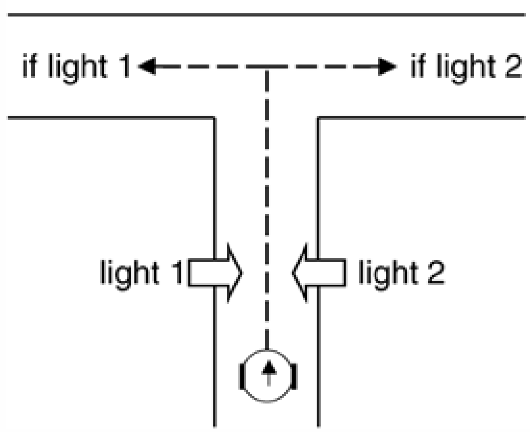
speed left	speed right	sample x,y
5 mm/sec	5 mm/sec	5.0, 5.0
5 mm/sec	-5 mm/sec	0.1, 0.0
...	...	...
10 mm/sec	10 mm/sec	9.9, 9.8

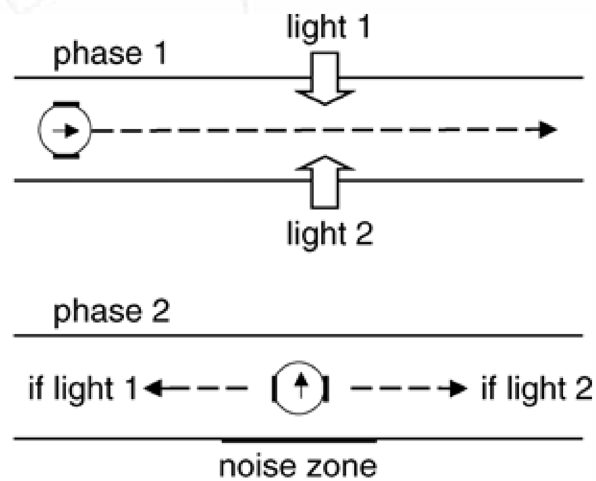
This method guarantees an excellent transfer from simulated to real robot [Miglino et al., 1995], but it is feasible only for simple sensors and for simple environments (squared and circular objects).



Simulation: Minimal

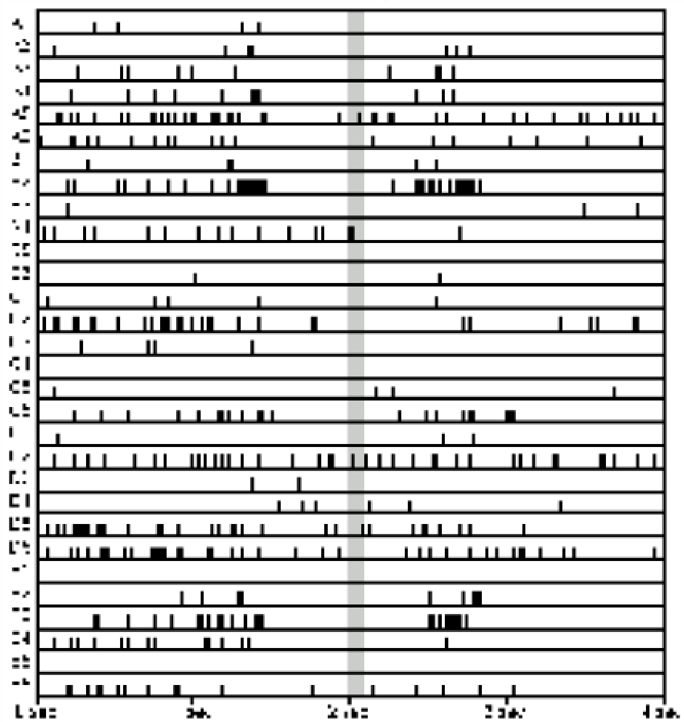
Minimal simulations [Jakobi, 1997] model only those characteristics of the interaction between robot and environment that are relevant for the expected behavior (*base set features*). The remaining features are considered *implementation-specific* and therefore are simpified and varied randomly from one trial to the next so that evolution does not rely on them.





Minimal simulations speed up significantly computing time and transfer well to the real world, but require the programmer to *know in advance* what will be the relevant features that must be accurately modeled.

Brains & Models

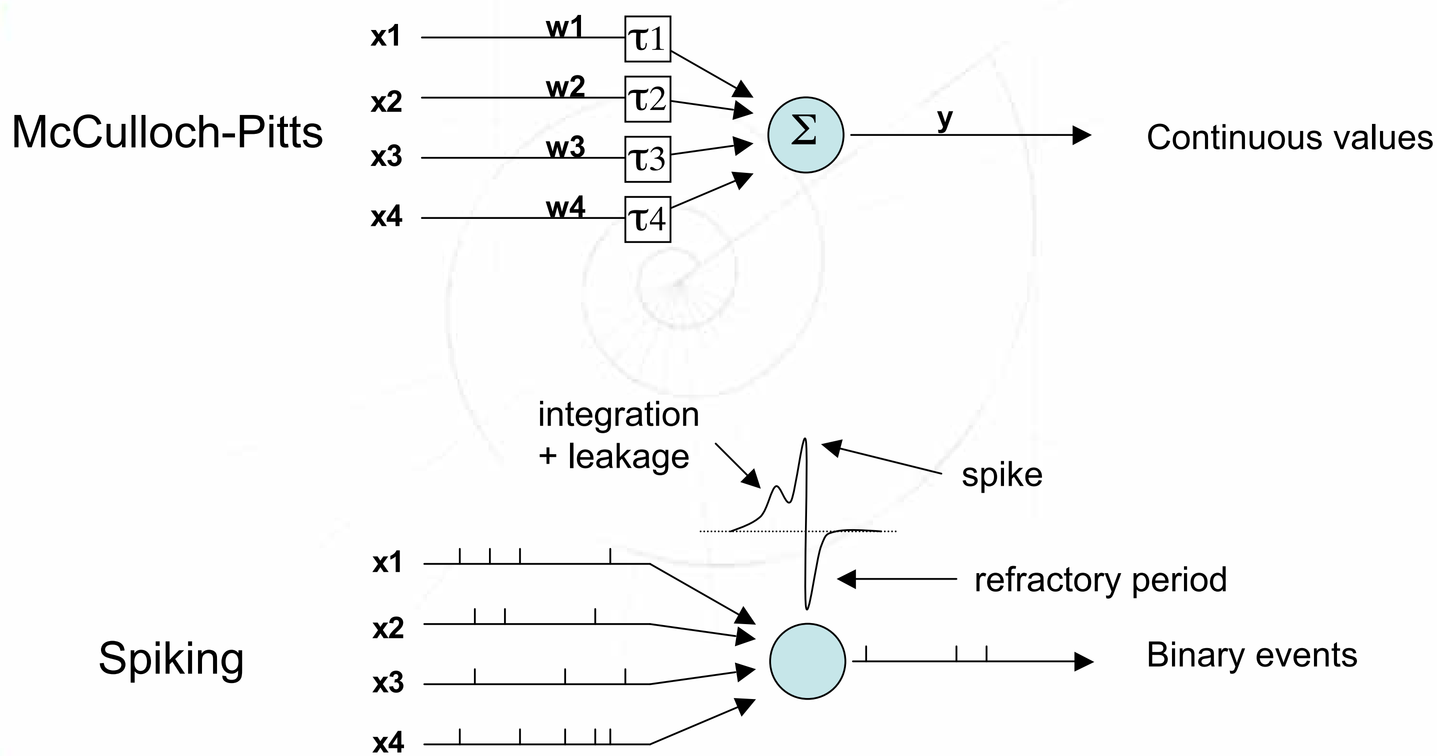


100 ms

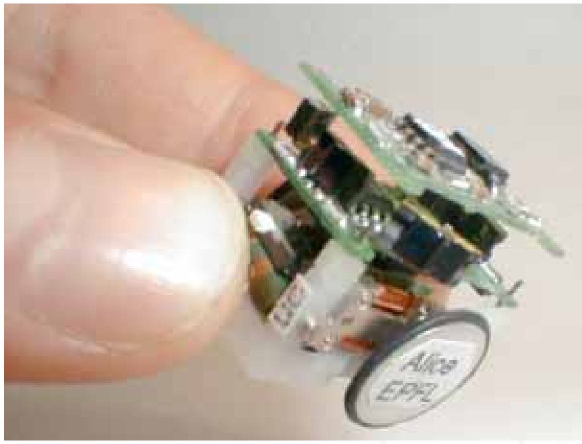
Firing rate → McCulloch-Pitts → Connectionism

Firing time → Pulsed neurons → Computational Biology

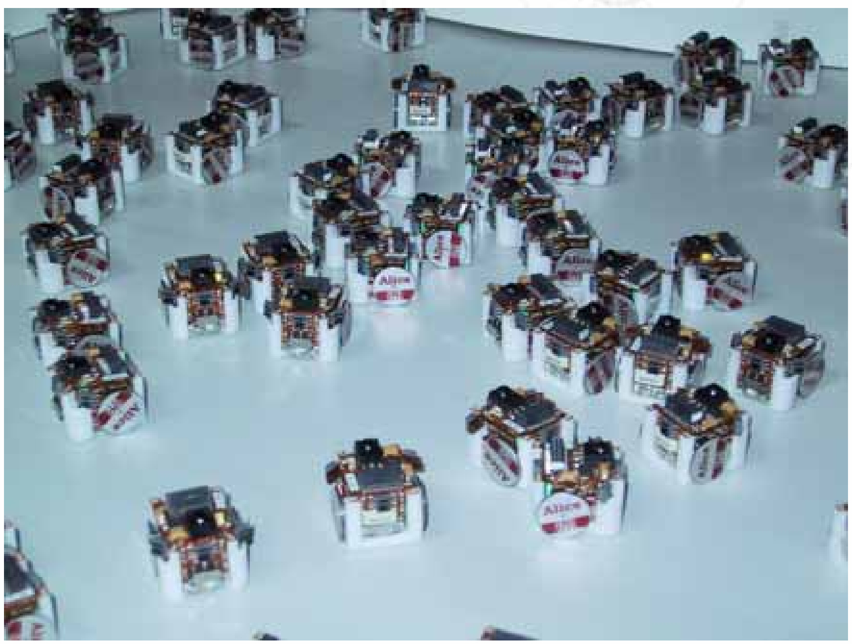
Artificial Neurons



Alice Micro-Robot



- microcontroller PIC16F84
- 2mA @ 5V
- 10 hours autonomy
- 2 swatch motors
- 4 proximity sensors
- modular (vision, radio, etc.)



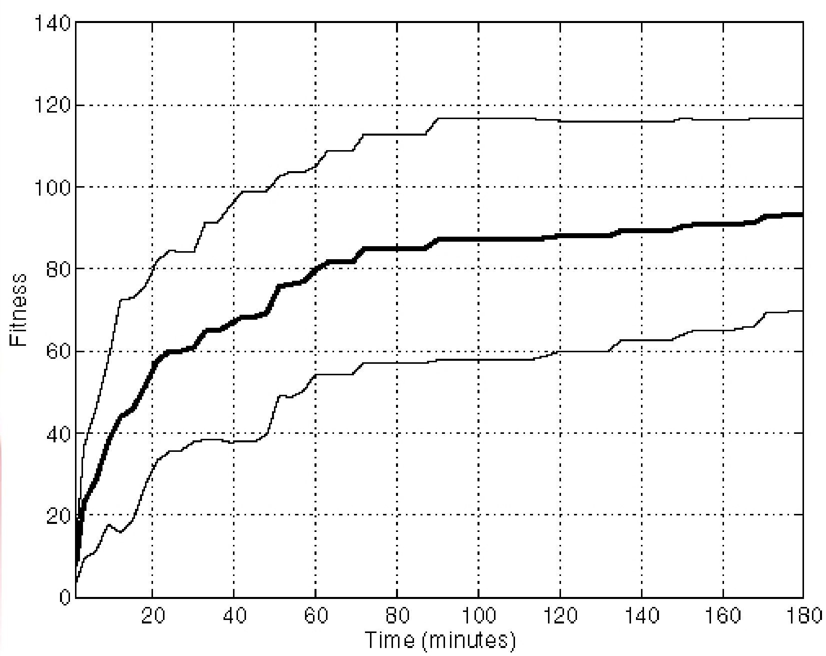
Bit-size Evolution

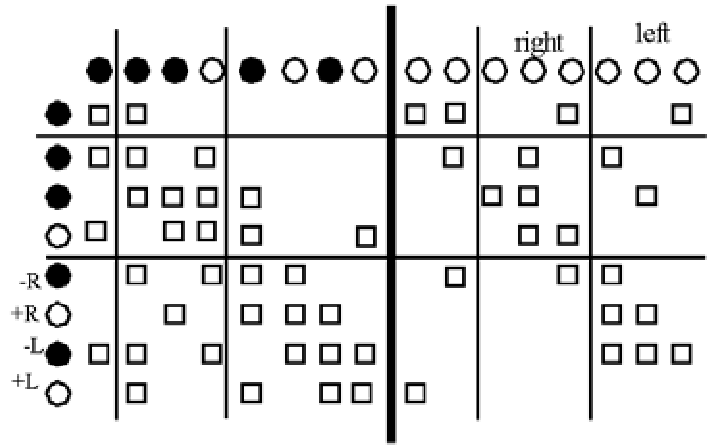
[Floreano et al, 2002]

Forward navigation with obstacle avoidance

$$\text{Fitness} = V \times \Delta v \times (1-s)$$

Steady-state evolution





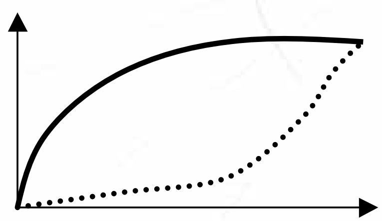
- bias: ↻
- IR Right:↺
- IR Left: ↻

Demo

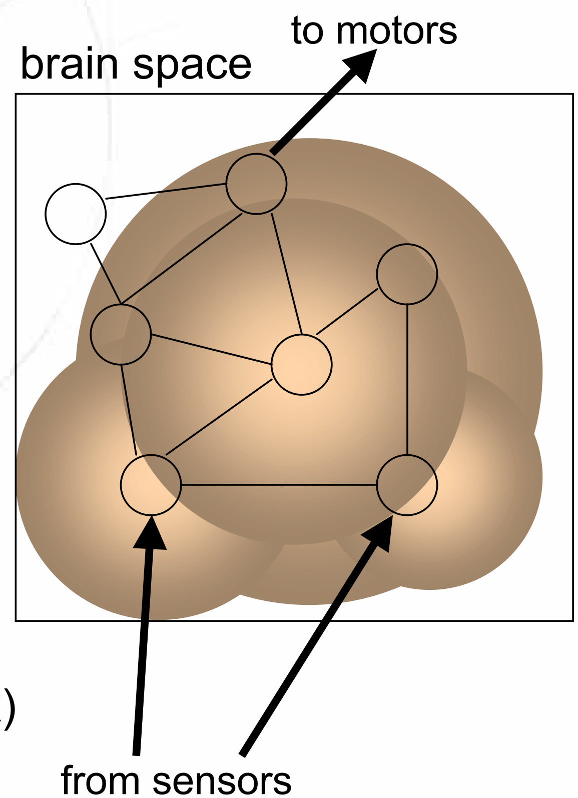
Evolution of Neural Gas

Computational view of the brain is based on wire metaphor and local communication. However, biological neurons can communicate across larger areas emitting gas.

An effect of gas is to change the response of other neurons that are sensitive to it.

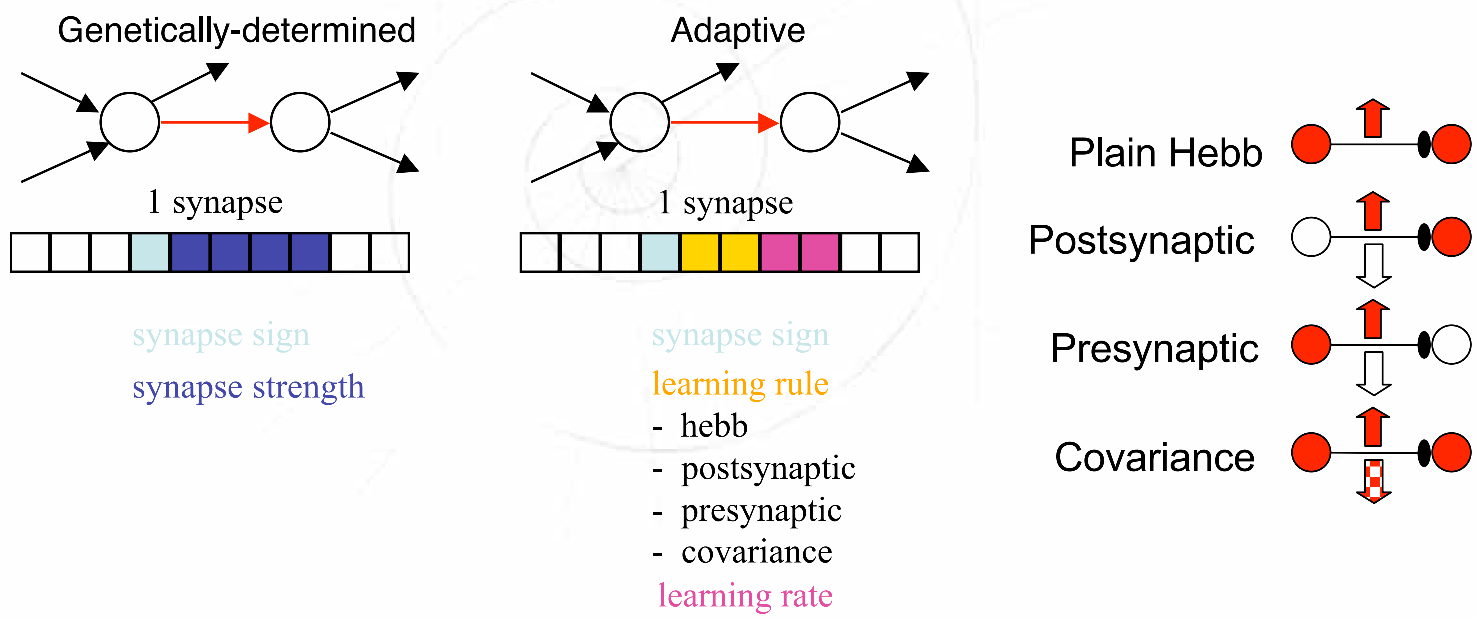


Smith & Husbands [1998, 2000] have explored evolution of gas emitting controllers for vision-based navigation (gantry robot task)



Evolution OF Learning

Floreano and Mondada [1994] suggested to genetically encode and evolve different types of learning rules found in biological brains. The rules are applied to the synaptic weights starting always from random initial conditions.

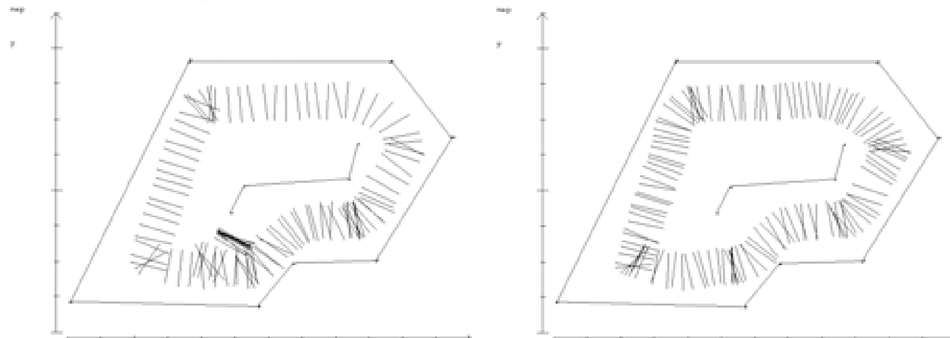


Important aspects:

- A neural network can use different learning rules in different parts
- There is no need of teacher or reinforcement learning, no gradient descent and local minima
- The Baldwin effect cannot take place, individuals are selected for their ability to learn

On-line self-adaptation

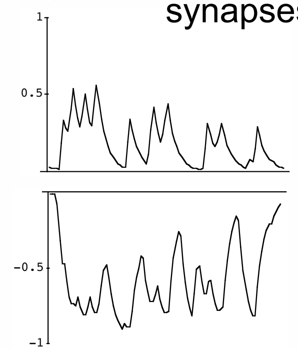
For sake of comparison, a Khepera robot has been evolved in the looping maze used earlier. Evolved robots display the ability to develop the obstacle avoidance navigation in few seconds after being created and improve it over time.



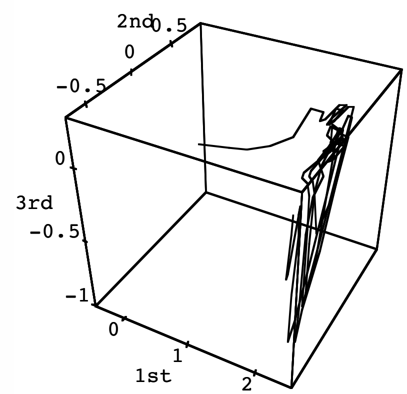
In addition, they perform well in different environments by developing suitable strategies. Contrary to conventional models, several synapses continue to change, but the overal pattern of change is dynamically stable.

Test in new environment

Continuously changing synapses



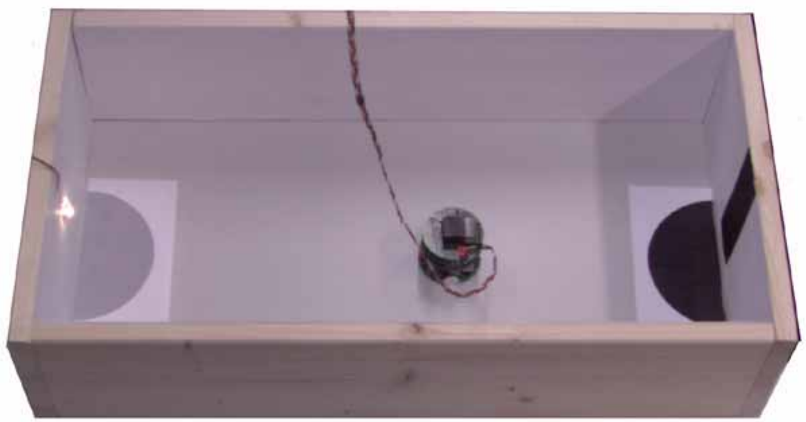
Dynamic stability

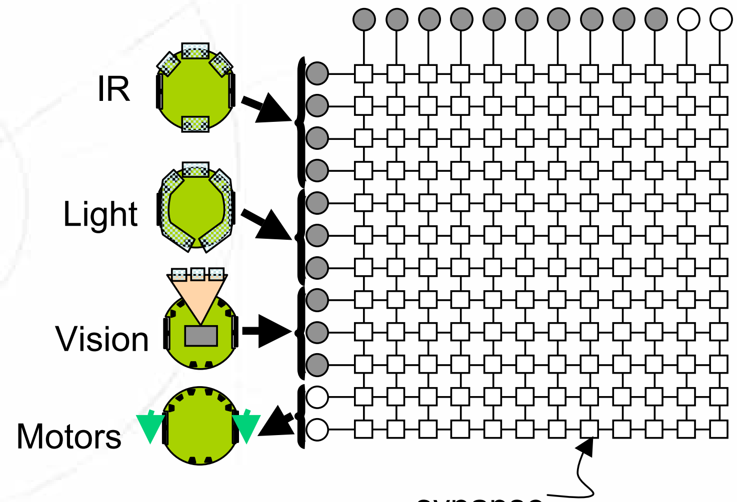


A Sequential Task

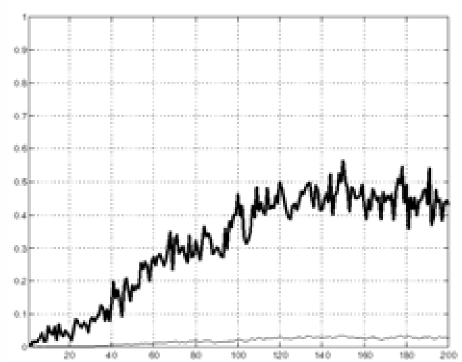
A Khepera robot is evolved to switch on a light and go under the light, but this sequence of actions is not directly rewarded by the fitness function. Two conditions are measured, evolving weights or learning rules.

Fitness= time_gray_light / total_time

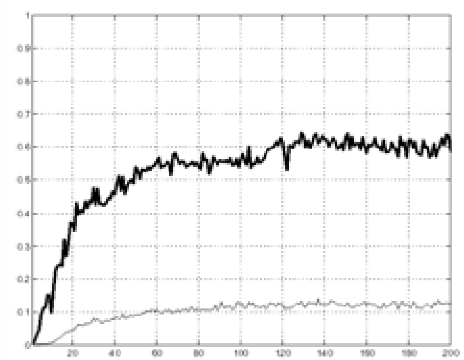




evolution of weights

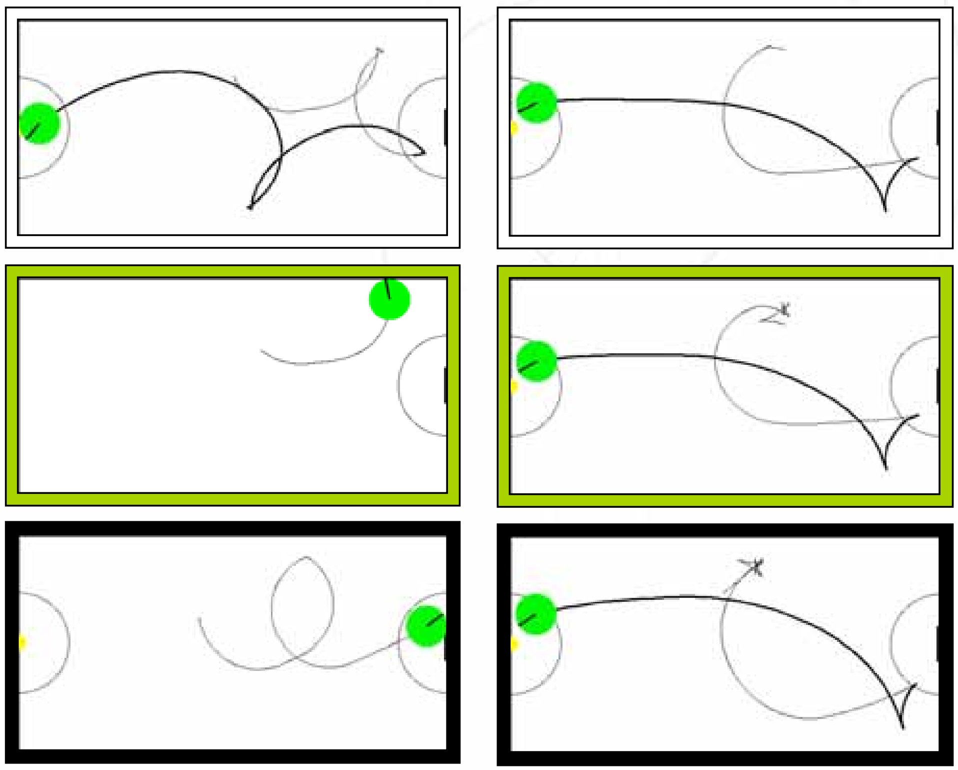


evolution of rules



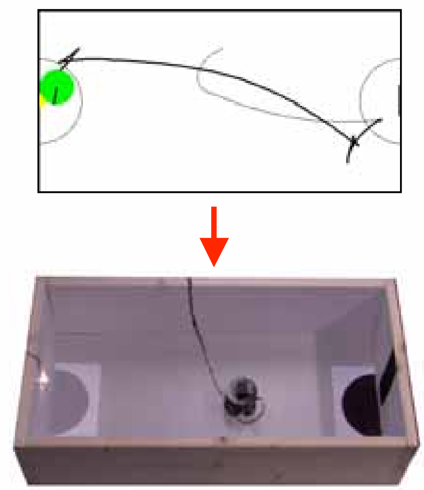
Sensory appearance

Let now take best evolved individuals and put them in conditions that are different from those experienced during evolution.
Evolved adaptive individuals can cope with new colours of the walls whereas genetically-determined individuals fail.



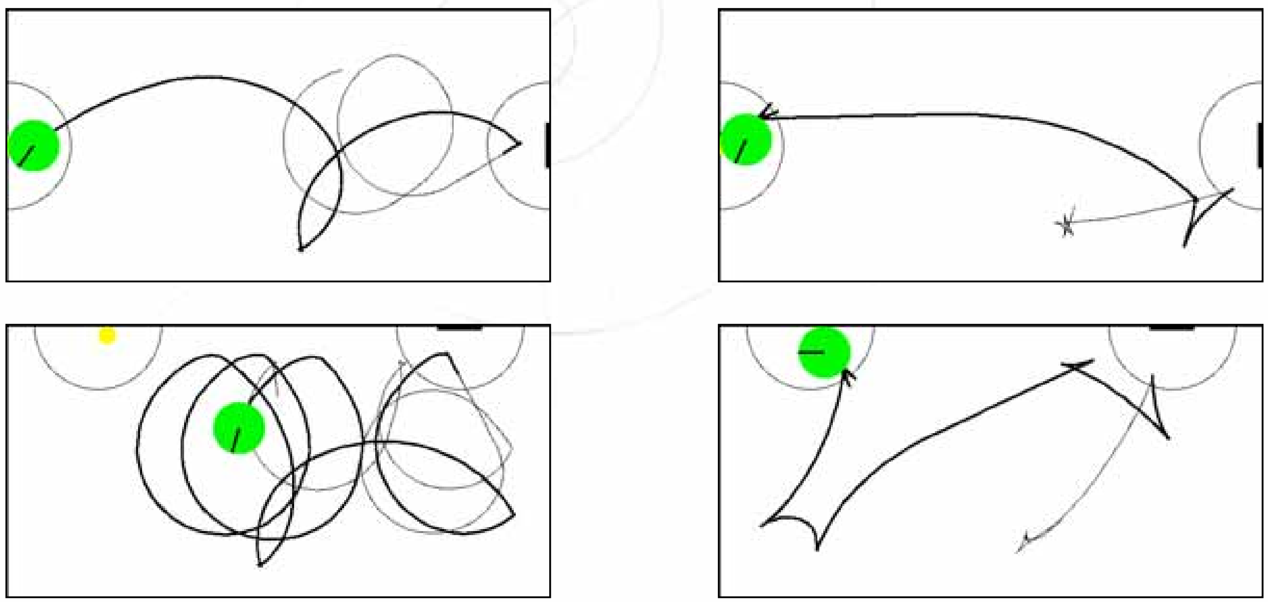
Genetically-determined Adaptive

Similarly, evolved adaptive individuals transfer smoothly from simulated to real world.



Environmental layout

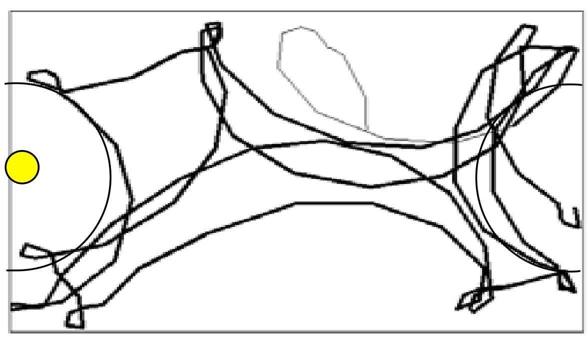
Another important feature of this environment is the position of the the light switch and of the light bulb.
Evolved adaptive individuals can cope with new positions of the two landmarks whereas genetically-determined inviduals cannot.



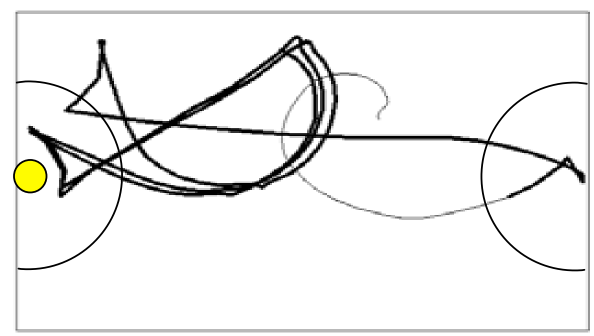
Genetically-determined

Adaptive

Morphology



Genetically-determined



Adaptive

Co-evolution

Competitive Co-Evolution is a situation where two different species co-evolve against each other. Typical examples are:

- Prey-Predator
- Host-Parasite

Fitness of each species depends on fitness of opponent species.

Potential advantages of Competitive Co-evolution:

– It may increase adaptivity by producing an evolutionary *arms race* [Dawkins & Krebs, 1979]

– More complex solutions may *incrementally* emerge as each population tries to win over the opponent

– It may be a solution to the *boostrap* problem

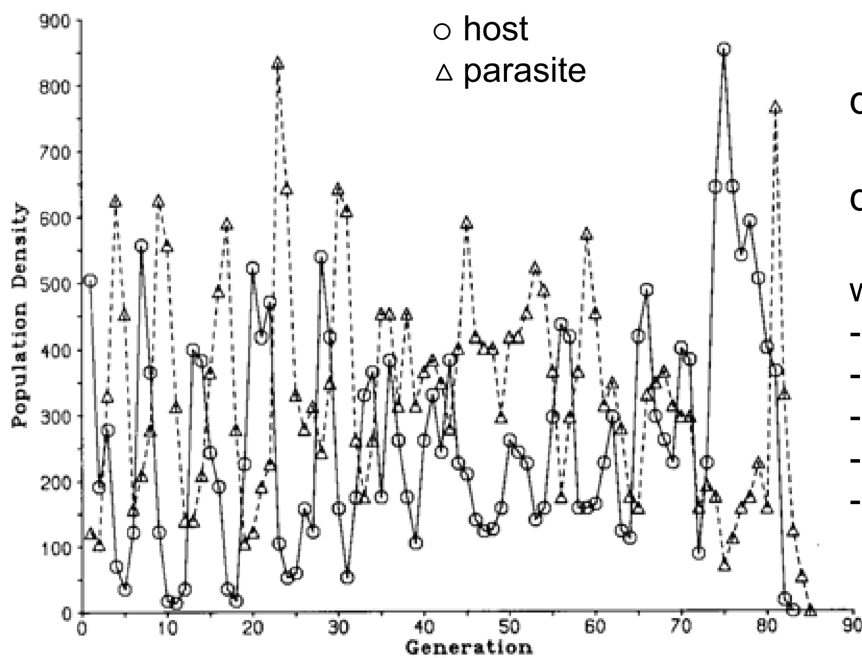
– Human-designed fitness function plays a less important role (= autonomous systems)

– Continuously *changing fitness landscape* may help to prevent stagnation in local minima [Hillis, 1990]

Formal Models

Formal models of competitive co-evolution are based on the Lotka-Volterra set of differential equations describing variation in population size.

Notice that in biology what matters is variation in population size, not behavioral performance, which is difficult to define and measure!



$$dN1/dt = N1\ (r1 - b1N2)$$

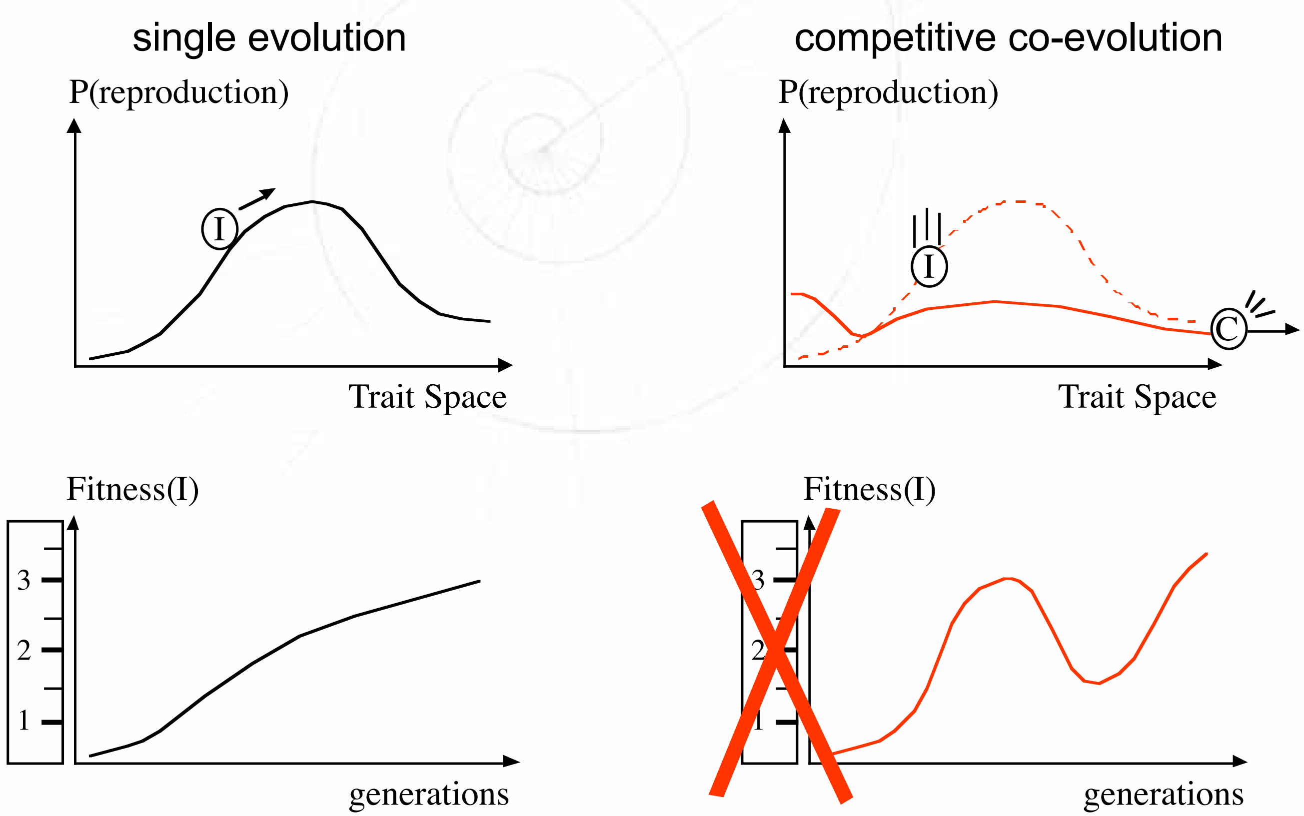
$$dN2/dt = N2\ (-r2 + b2N1)$$

where:

- N1, N2 are the two populations
- r1 is increment rate of prey without predators
- r2 is death rate of predators without prey
- b1 is death rate of prey caused by predators
- b2 is ability of predators to catch prey

Complications: Landscape

Whereas in single-species evolution the fitness landscape is static and fitness is a monotonic function of progress, in competitive co-evolution the fitness landscape can be modified by the competitor and fitness function is no longer an indicator of progress.

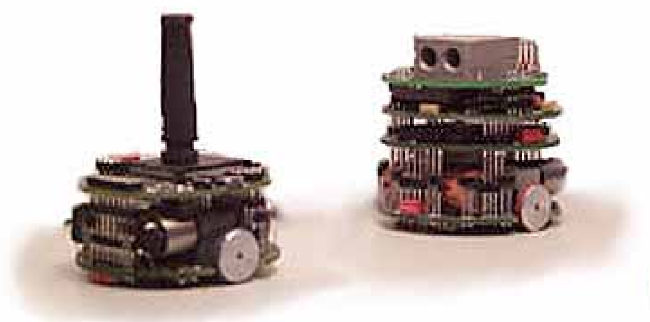


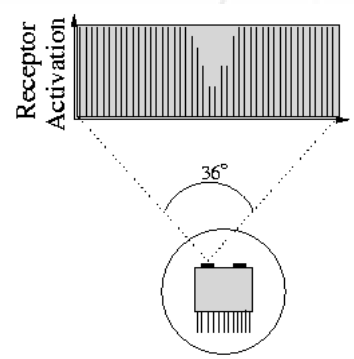
Predator-Prey Robots

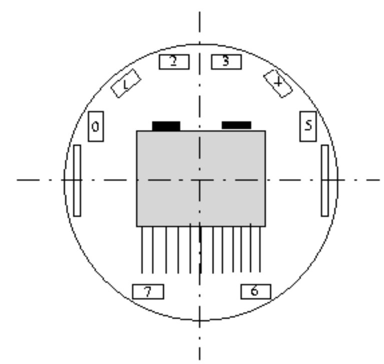
Let us consider the case of two co-evolutionary robots, a predator and a prey, that evolve in competition with each other. Questions:

a) can we evolve functional controllers with simple fitness functions?
b) what are the emerging dynamics?
c) do we observe incremental progress?
d) are co-evolved solutions better than evolved solutions?

Goal = Predator must catch the prey, prey must avoid predator
Prey = proximity sensors only, twice as fast as predator
Predator = proximity + vision, but half max speed of prey



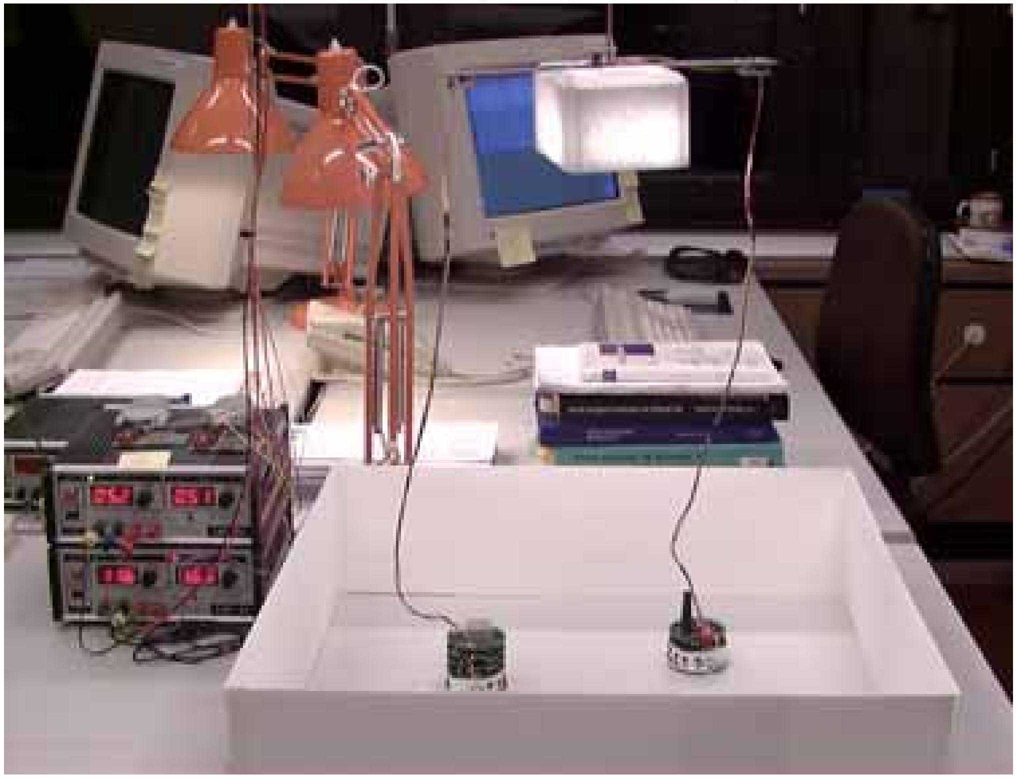




Experimental Setup

The two robots are positioned in a white arena. Predator and prey are tested in tournaments lasting 2 minutes. Robots are equipped with contact sensors.

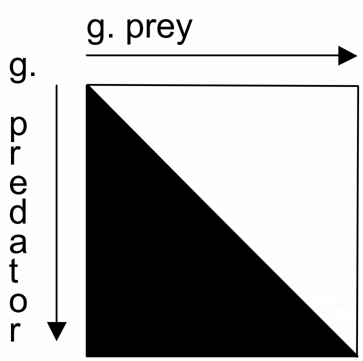
Fitness prey = TimeToContact   Fitness predator = 1-TimeToContact



Measuring Progress

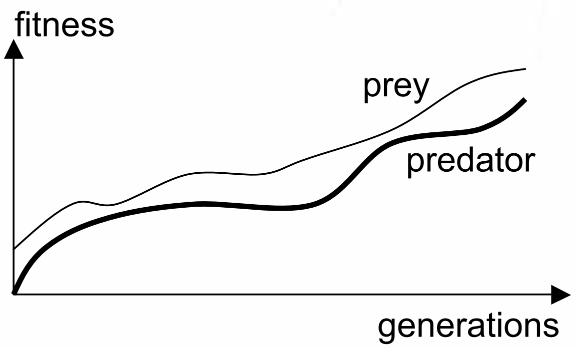
Progress can be measured by testing evolved individuals against all best opponents of previous generations. There are two ways of doing so.

CIAO graphs [Cliff & Miller, 1997]



These graphs represent the outcome of tournaments of the Current Individual vs. Ancestral Opponent across generations. Ideal continuous progress would be indicated by lower diagonal portion in black and upper diagonal portion in white.

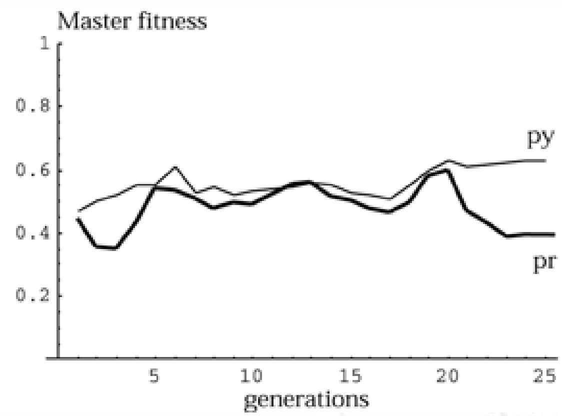
MASTER tournaments [Floreano & Nolfi, 1997a]



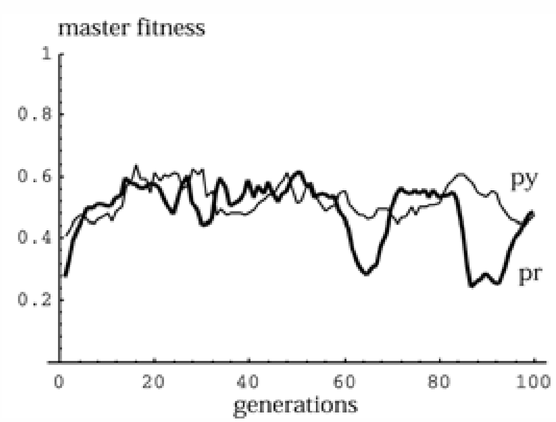
These graphs plot the average outcome of tournaments of the current individual against all previous best opponents. Ideal continuous progree would be indicated by continuous growth.

Limited Progress

with real robots

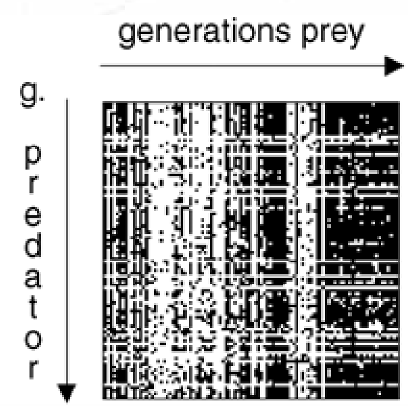


with simulated robots



Progress analysis of co-evolved robots using Master Tournament technique shows that there is some progress only during the initial 20 generations. After that, the graphs are flat or even decreasing. In other words, individuals born after 50 generations may be defeated by individuals that were born 30 generations earlier.

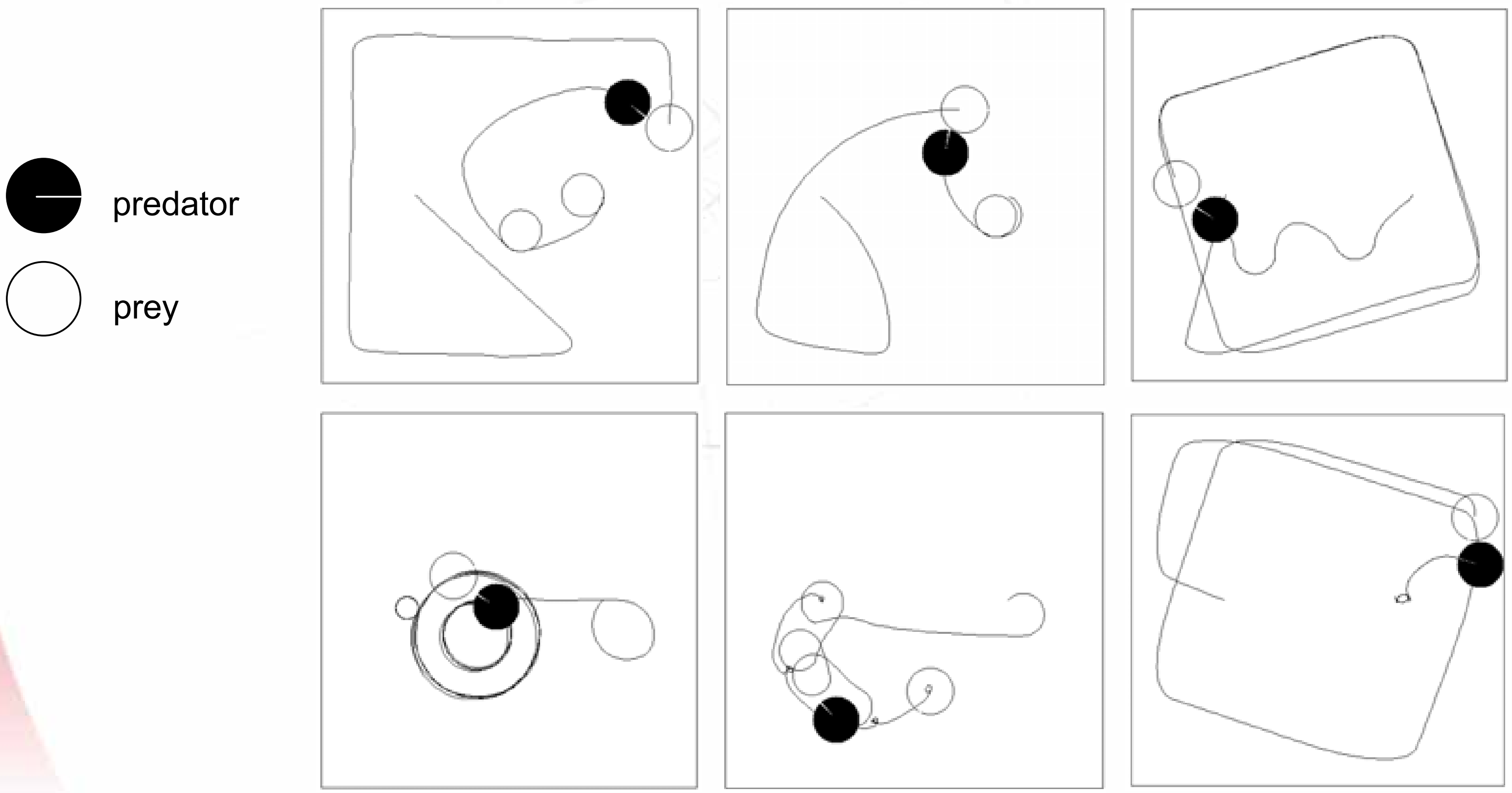
These data indicate that co-evolution may have developed into re-cycling dynamics after 20 generations.



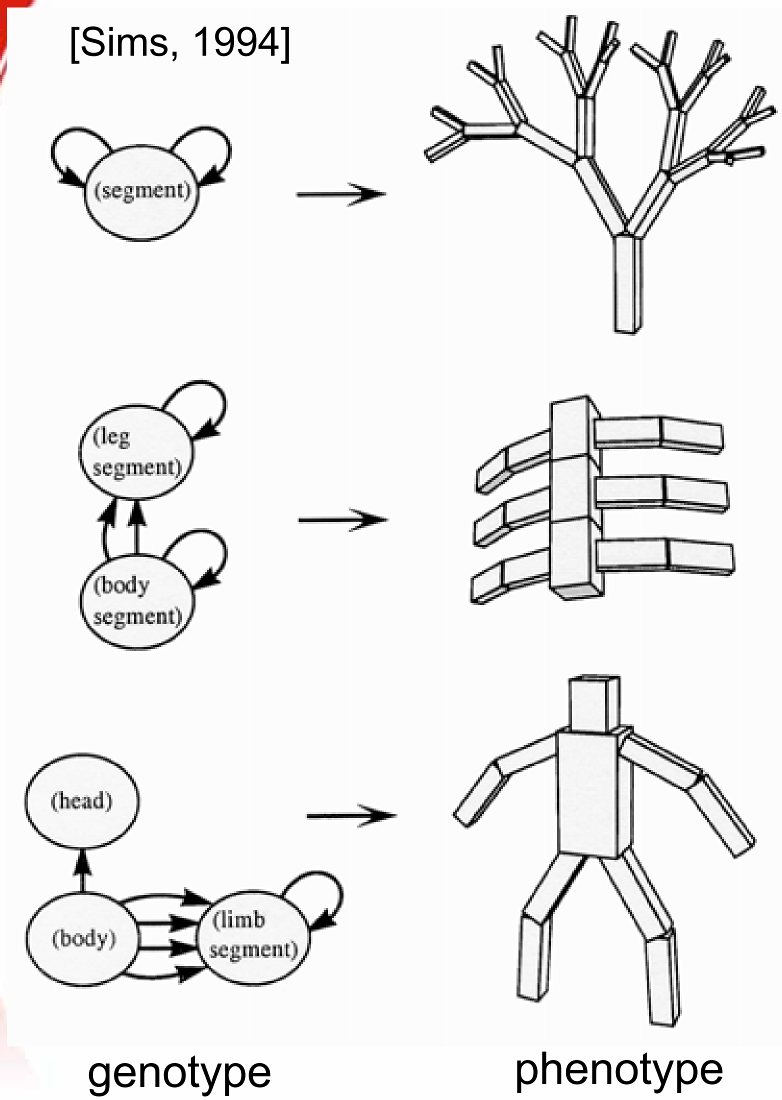
CIAO data are even less capable of revealing progress.

Emerging strategies

Despite lack of progress measured against previous opponents, co-evolved individuals display highly-adapted strategies against their opponents and a large variations of behaviors.
Each tournament shows individuals belonging to the same generation.



Virtual Creatures



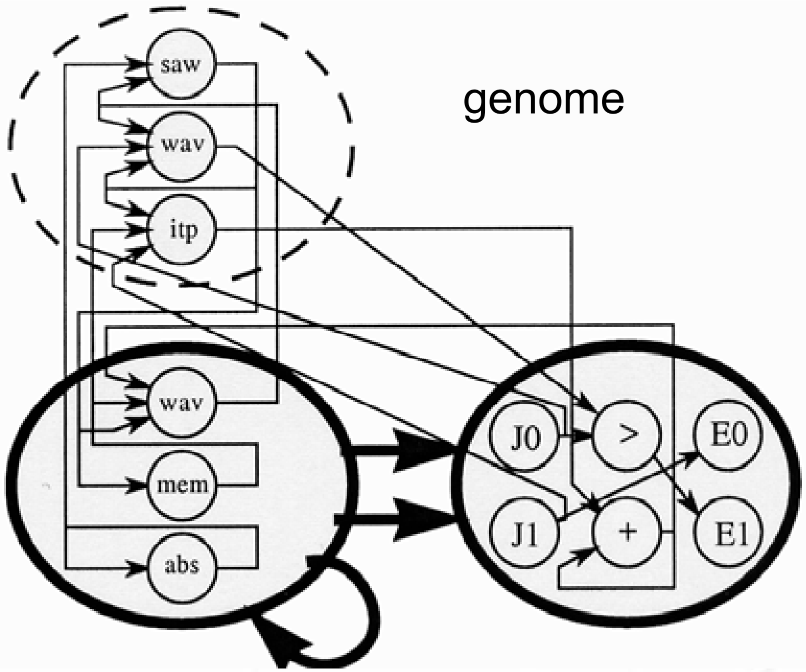
Body representation is directed graph. Nodes have properties:

- dimension
- joint type (rigid, twist, revolute, ...)
- recursive-limit
- connection (position, orientation, scale, reflection)
- terminal
- neural circuit

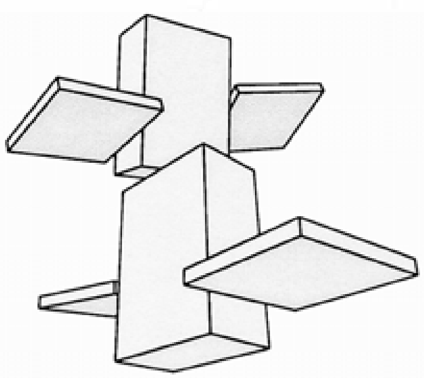
Neural circuit representation is directed graph. Nodes have properties:

- sensor
  - joint sensor
  - contact sensor
  - photosensor
- neuron (math type)
  - sum
  - memory
  - oscillator
  - max, etc.
- effector (force on muscle)
  - positive/negative (push/pull)

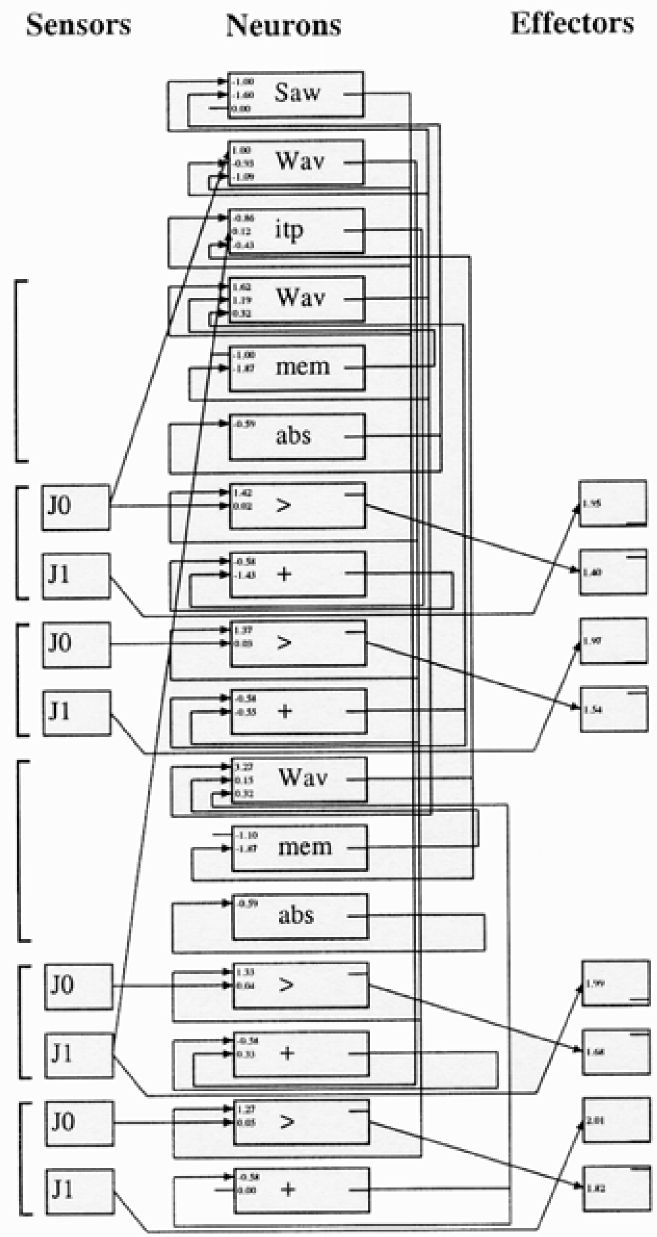
Example



genome



body



brain

[Sims, 1994]

Legged Robots

Co-evolving neural control and body parameters (Floreano et al., 2000)

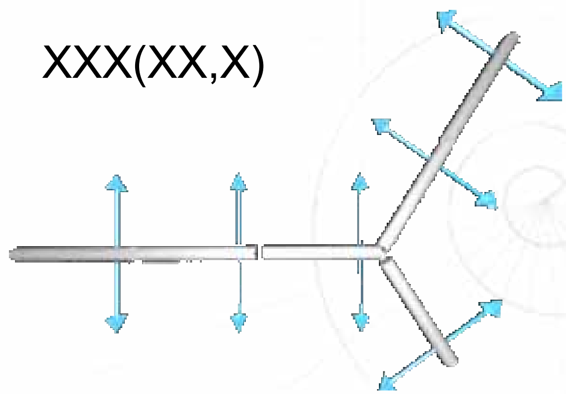
1) Co-evolution of body/leg ratio and of control system in simulation.

2) Physical realisation of evolved body-plan and transfer of evolved controller.

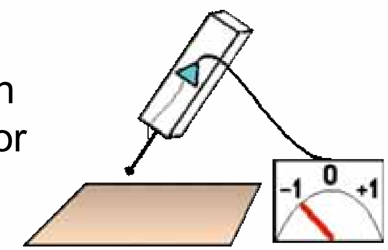
Framstick [Komosinski & Ulatowski, 1999]

Primitives are joined sticks. Sticks can host sensors and neurons. Joints are actuated by muscles. Fast simulation using finite element method (only force effects on few parts of the system are computed).

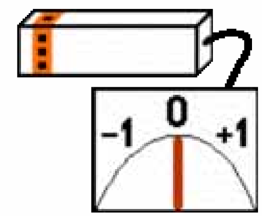
XXX(XX,X)

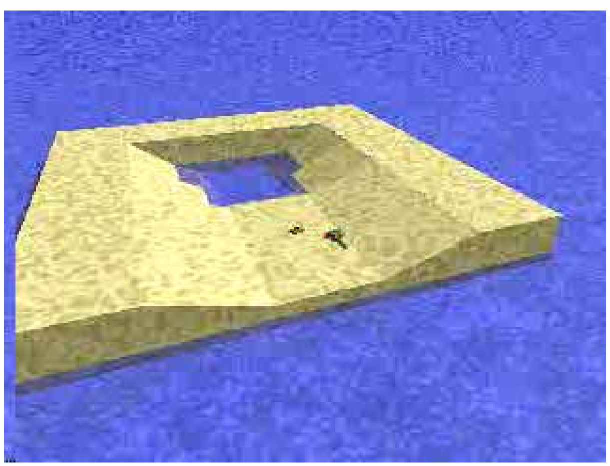


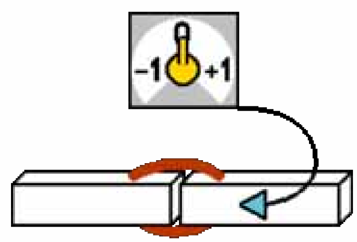
touch sensor



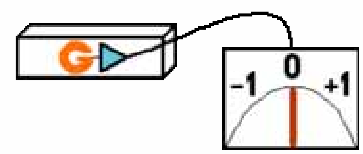
food sensor







muscle



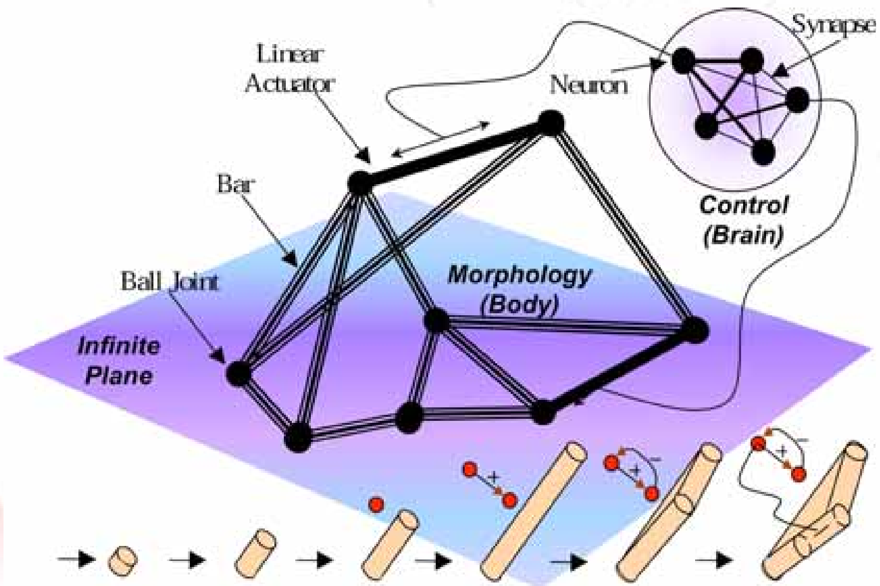
gyroscope

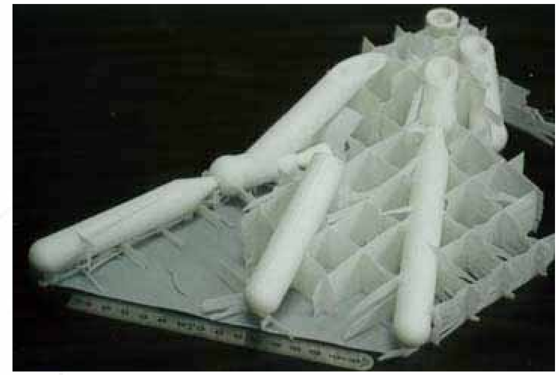
www.frams.poznan.pl

The Golem project

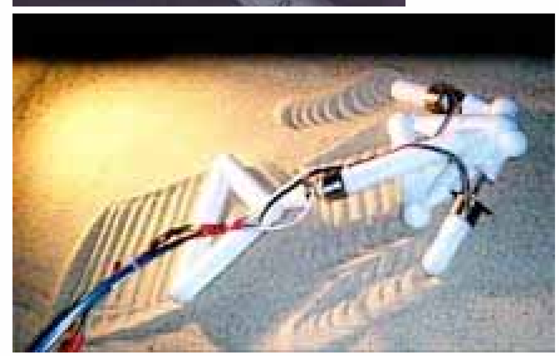
Lipson & Pollack (2000) added the physical construction of the creatures by using a 3D thermoplastic printer and extensible bars.

- Evolution takes place in simulation
- Fitness = distance covered by the robot
- Selected individuals are built by:
  - printing the bars
  - fitting joints and motors
  - downloading neural network in PIC controller



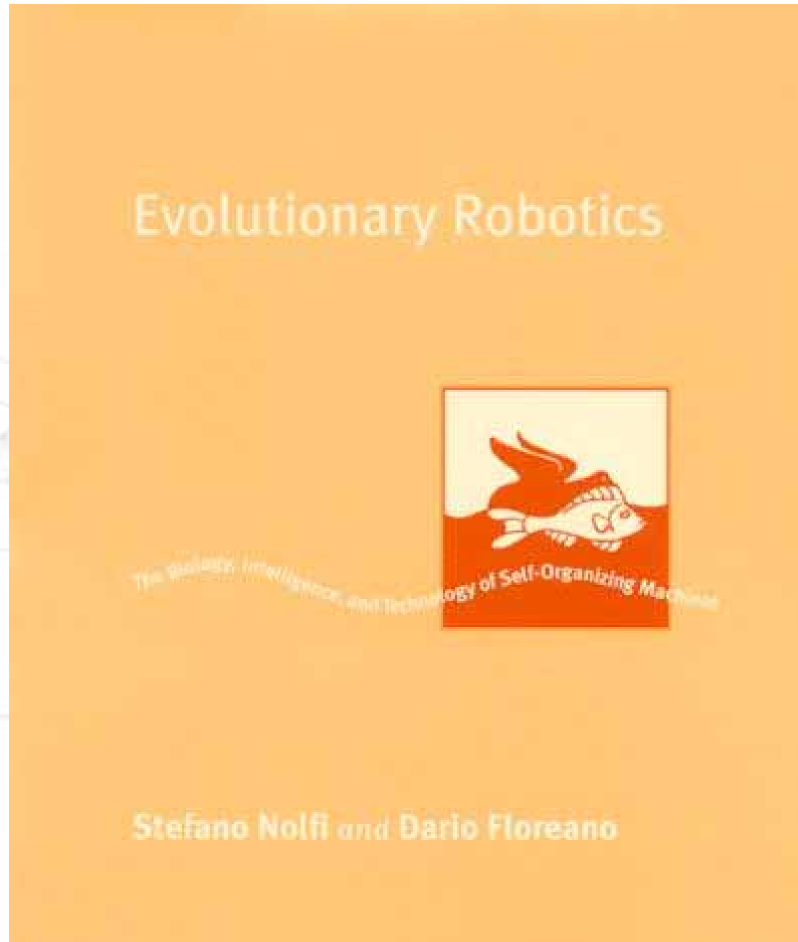


video clip



http://golem03.cs-i.brandeis.edu/

For more info…



MIT Press
Hardcover, 2000, 2001
Paperback, 2004

Free Software
http://gral.ip.rm.cnr.it/evorobot/simulator.html
http://asl.epfl.ch/resources/evo/index.php