Evolutionary Robotics

F. Sanfilippo [1]

[1] Department of Maritime Technology and Operations, Aalesund University College, Aalesund, Norway, fisa@hials.no

Supervisors: Kristin Ytterstad Pettersen and Houxiang Zhang

Topics in System and Control Theory (TK8111)

June 19, 2014

NTNU – Trondheim
Norwegian University of Science and Technology

HØGSKOLEN I ÅLESUND
Aalesund University College

Summary

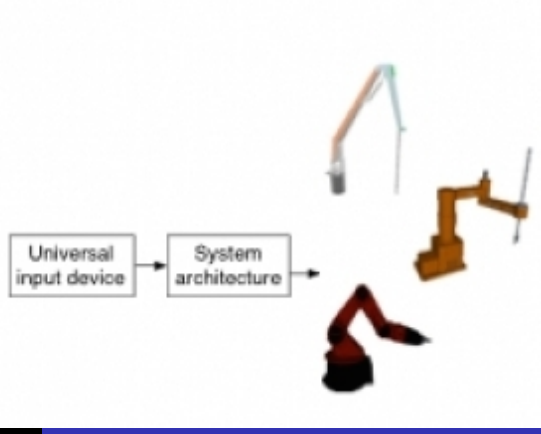
Motivations

Trigger: "A Flexible and Common Control Architecture for Rolls-Royce Marine Cranes and Robotic Arms"

Motivations

- Control different cranes, with dissimilar kinematics, by using a common input device
- Different kinematic models are needed in order to control different arms
- Is it possible to develop alternative control algorithms that are able to scale and handle different manipulator configurations, from simple ones with few DOFs to the most complex with a considerable number of DOFs?

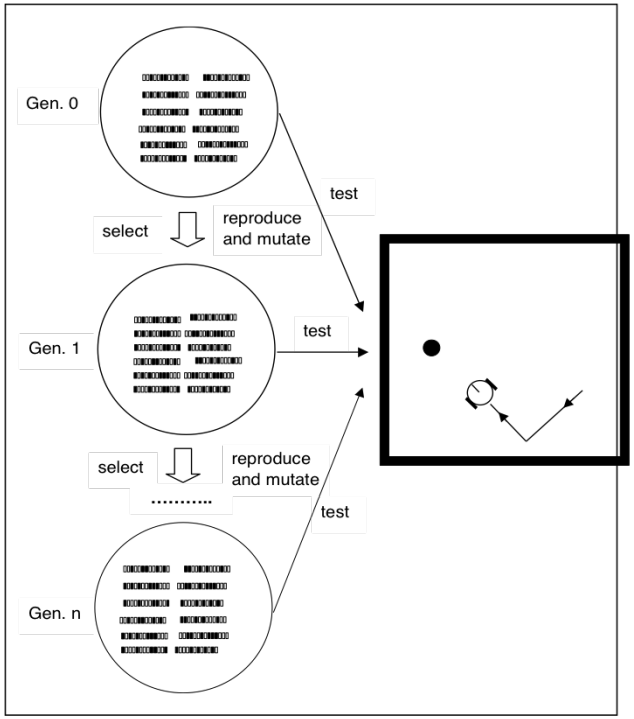
An alternative solution: no a priori knowledge for the IK model of the arm, a method that derives its kinematic properties from evolutionary robotics



The Idea

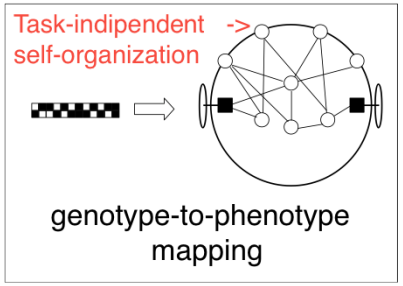
Evolutionary Robotics

- Initial random population
- Genotype-to-phenotype mapping and fitness assessment
- Reproduction



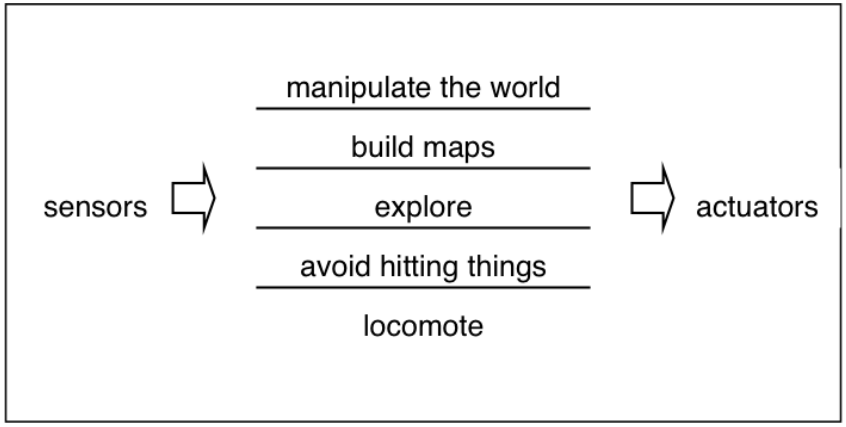
$$\Phi = V\left(1-\sqrt{\Delta v}\right)\left(1-i\right)$$

fitness function



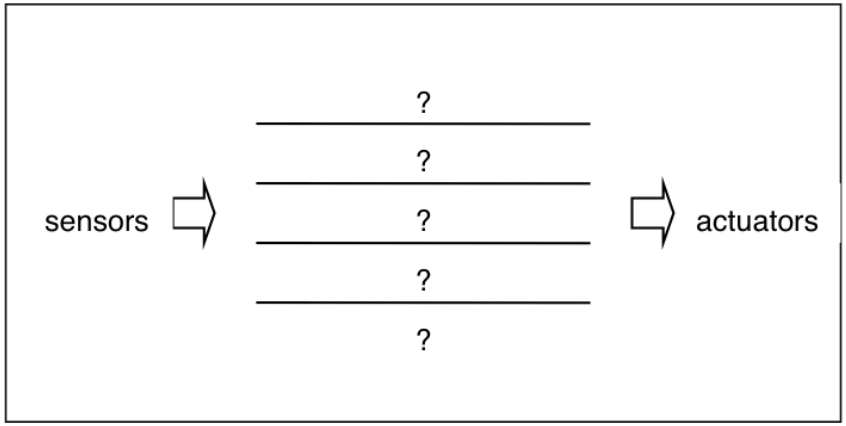
Stefano Nolfi & Dario Floreano, 2000

Behavior-Based Robotics Vs ER



behavior-based robotics

[Brooks, 1986]

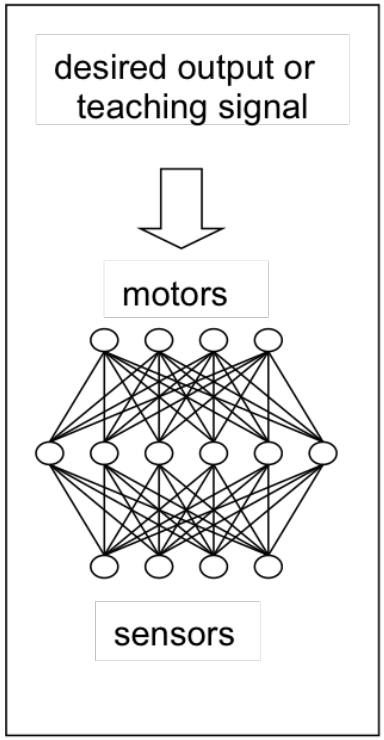


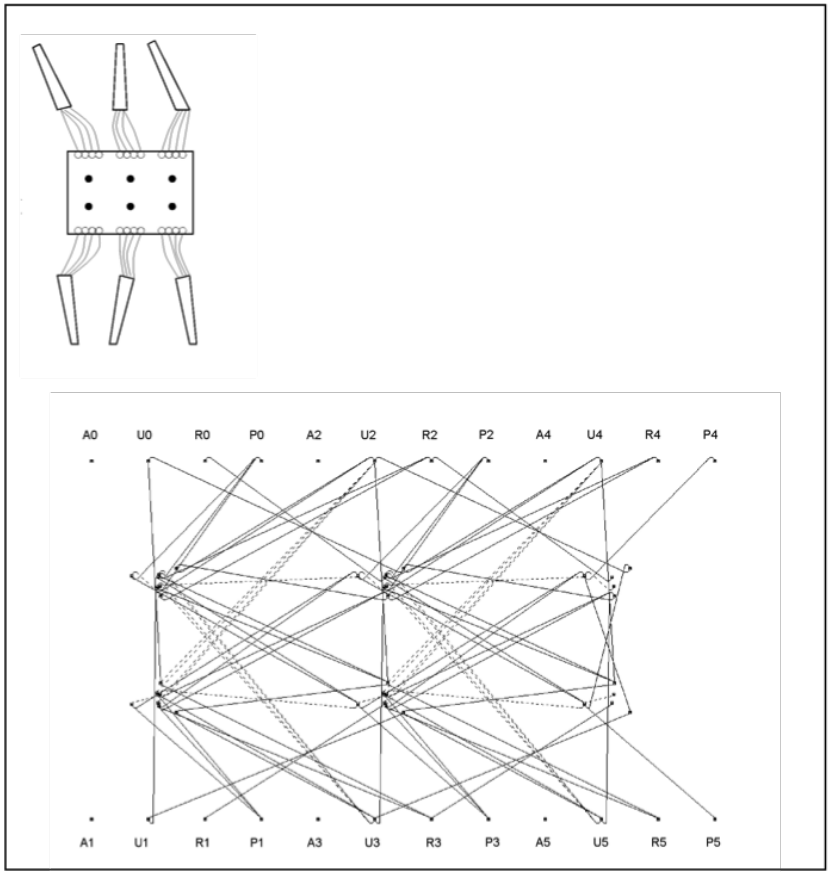
evolutionary robotics

Stefano Nolfi & Dario Floreano, 2000

Learning Robotics Vs ER

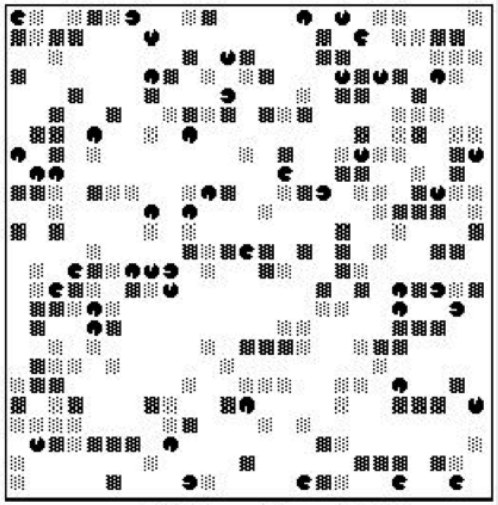
[Kodjabachian & Meyer, 1999]



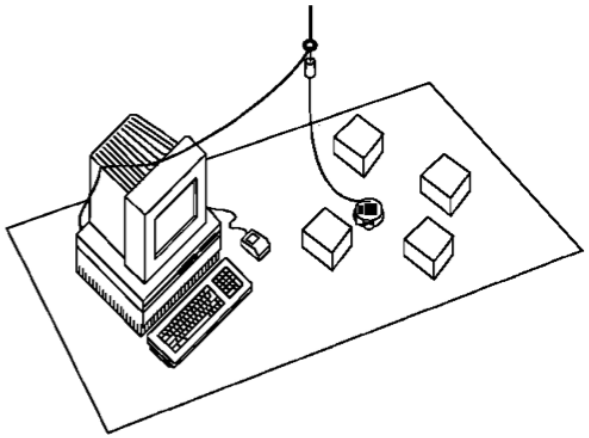


Stefano Nolfi & Dario Floreano, 2000

Artificial Life Vs ER



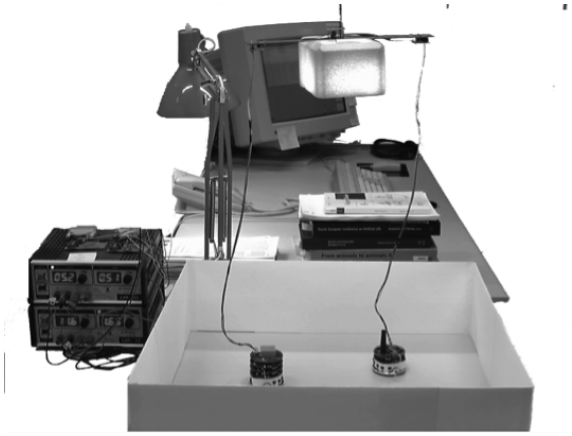
[Menczer and Belew, 1997]

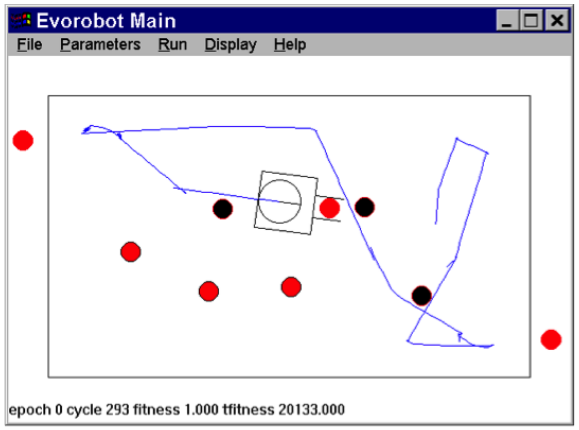


[Floreano and Mondada 1994]

Stefano Nolfi & Dario Floreano, 2000

How to Evolve Robots





evolution on the real world

[Floreano and Nolfi, 1998]

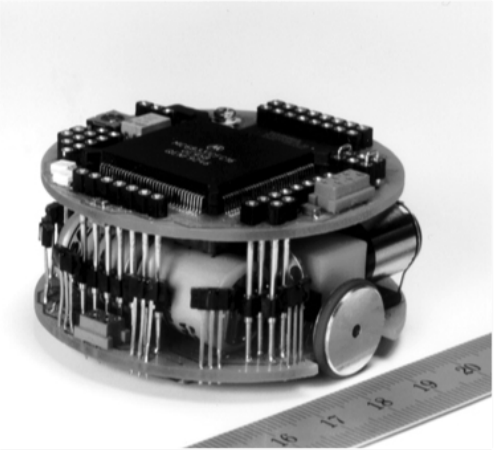
evolution on simulation

+ test on the real robot

[Nolfi, Floreano, Miglino, Mondada 1994]

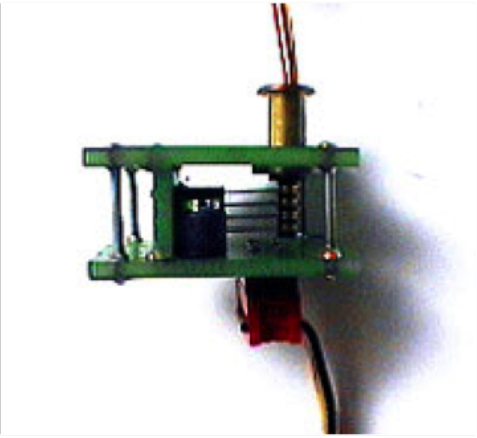
Evolution in the Real World

mechanical robustness



[© K-Team SA]

energy supply



[© K-Team SA]

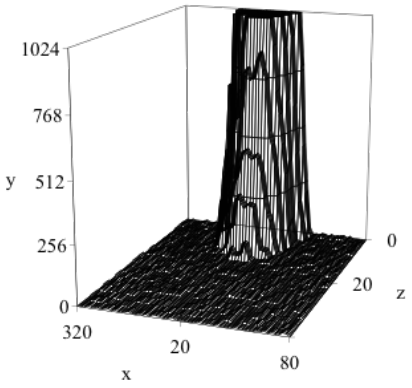
analysis

[© K-Team SA]

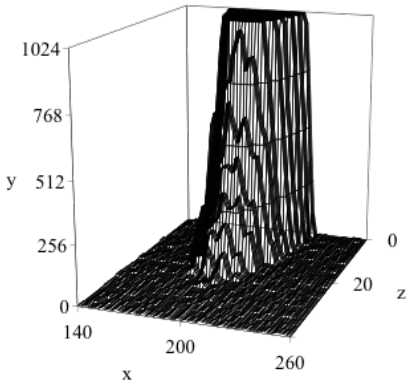
[Floreano and Mondada, 1994]

Evolution in Simulation

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



4th IF sensor



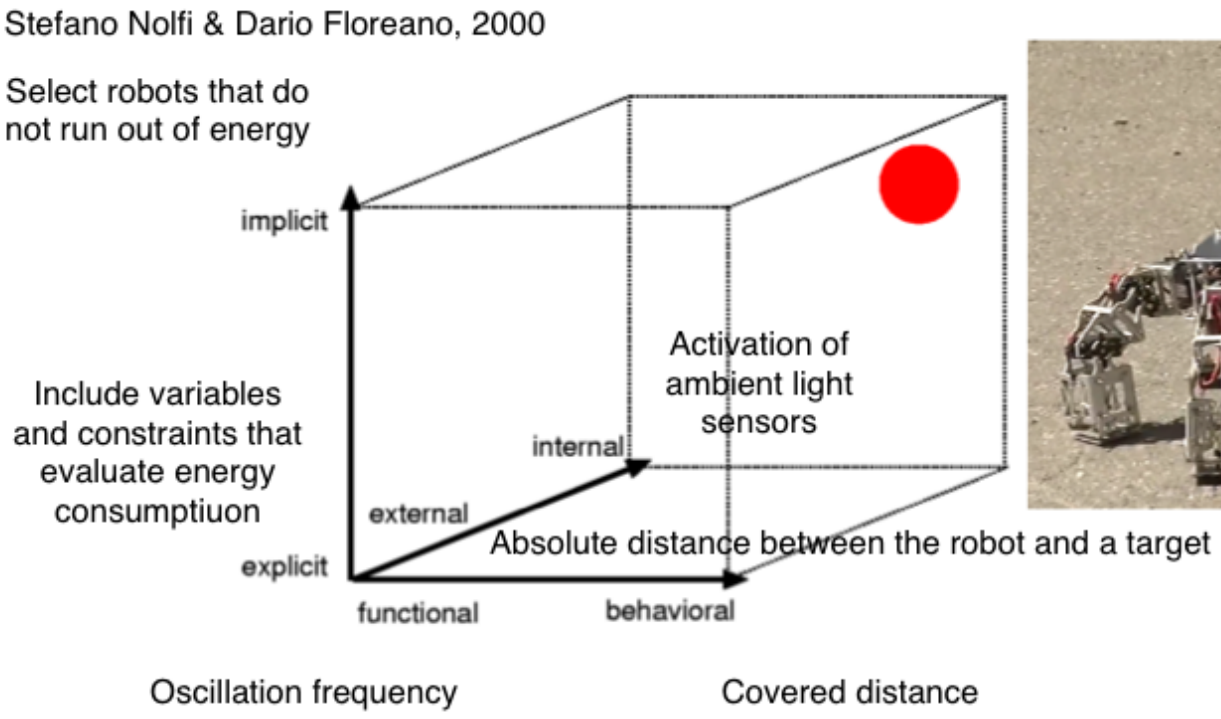
8th IF sensor

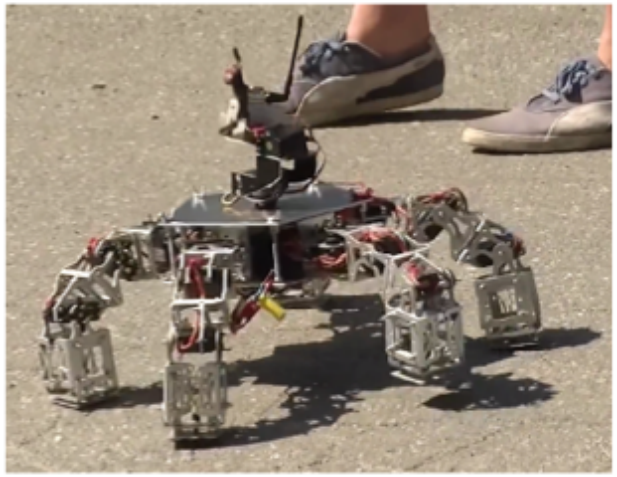
Simulating the robot

- Physical sensors deliver uncertain values and commands to actuators have uncertain effects
- The body of the robot and the environment should be accurately reproduced in the simulation

Designing the Fitness Function

Fitness space provides a framework for describing fitness functions of autonomous systems.

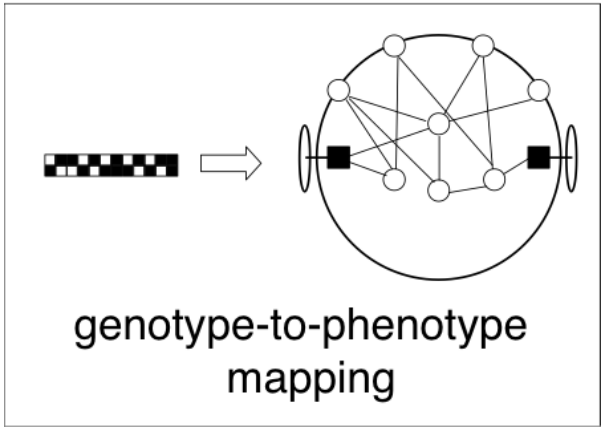




Genetic Encoding

Requirements for efficient genetic encoding

- Evolvability
- Expressive power
- Compactess
- Simplicity



Stefano Nolfi & Dario Floreano, 2000

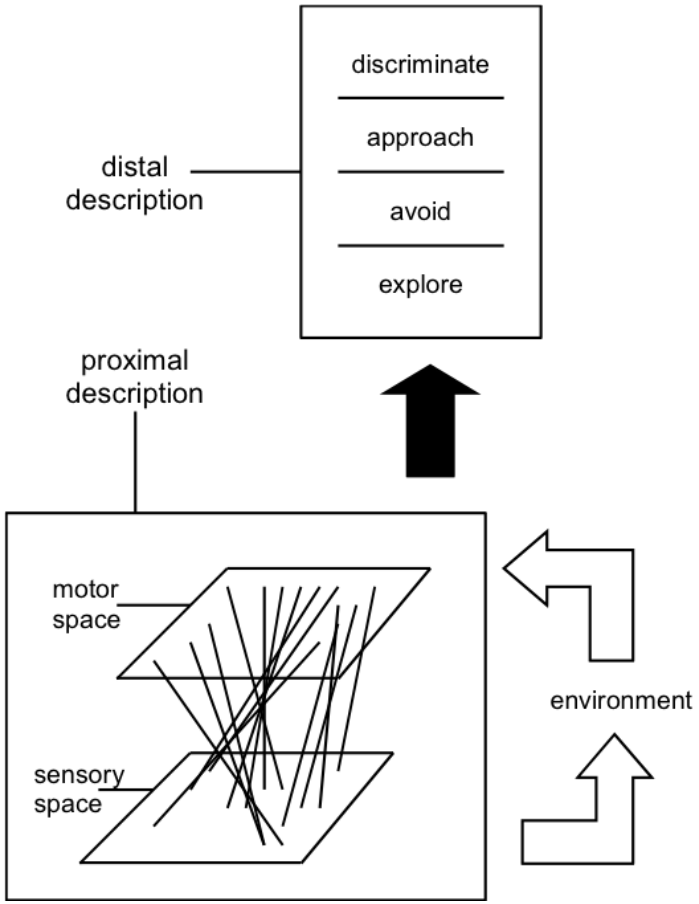
Adaptation Vs Decomposition and Integration

The main strategy followed to develop robots has been that of Divide and Conquer

- Divide the problem into a list of hopefully simpler sub-problems
- Build a set of modules or layers able to solve each sub-problem
- Integrate the modules so to solve the whole problem

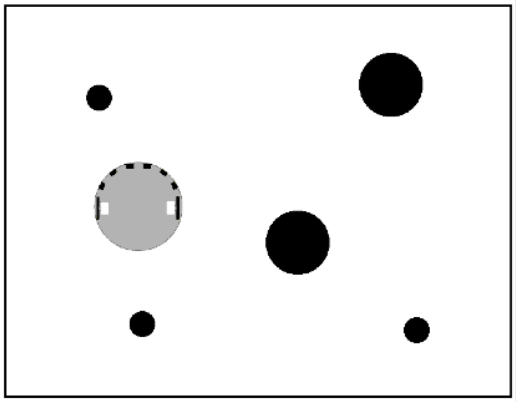
Unfortunately, it is not clear how a desired behavior should be broken down. The main reason for which it is difficult to break down a desired behavior into simpler pieces is that behavior is not only the result of the robot's control system but the result of the interaction between the robot and the environment

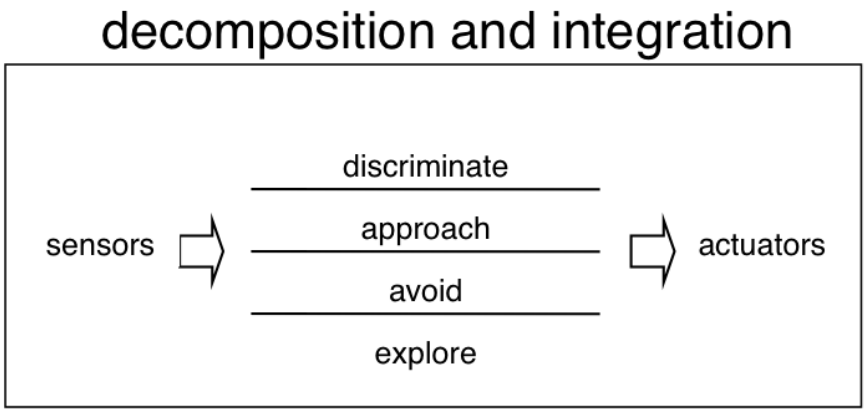
Proximal and Distal Descriptions of Behaviors

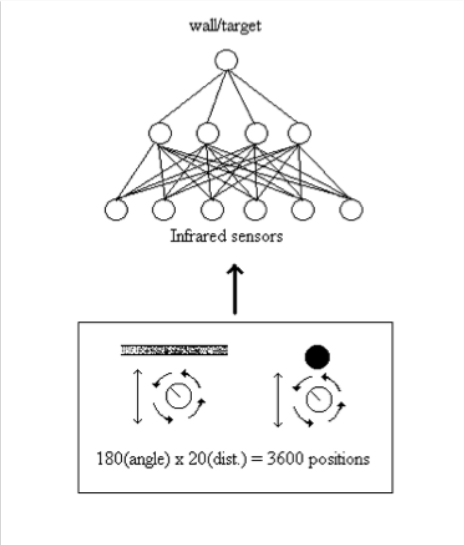


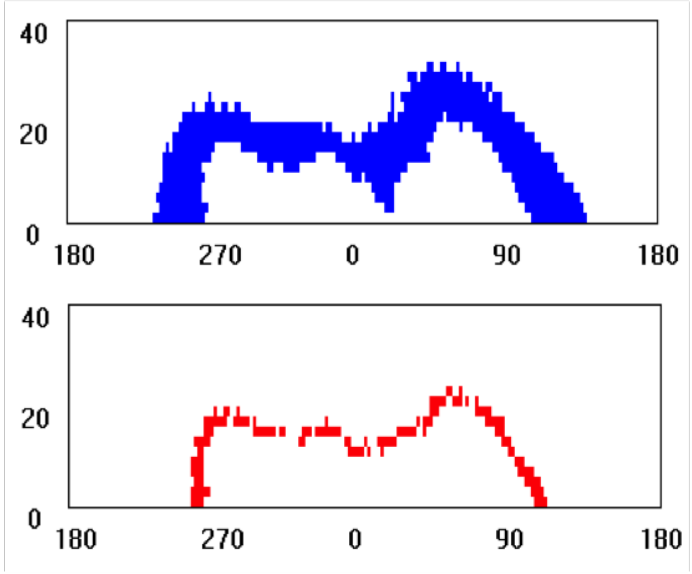
[Nolfi, 1997]

Discrimination Task







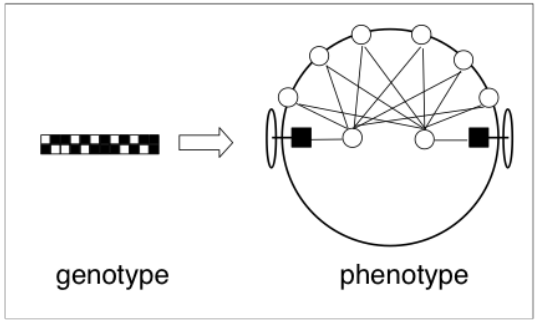


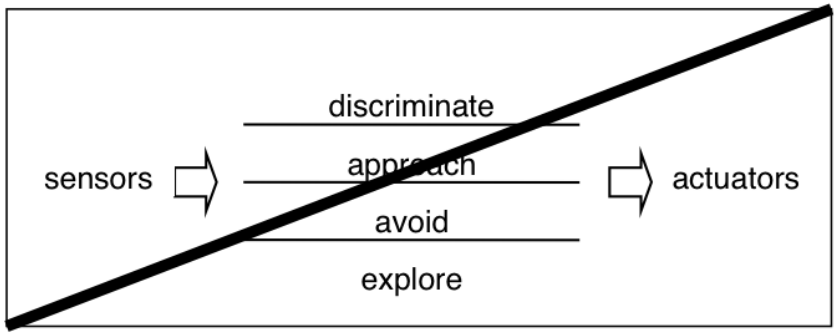
walls and cylinders

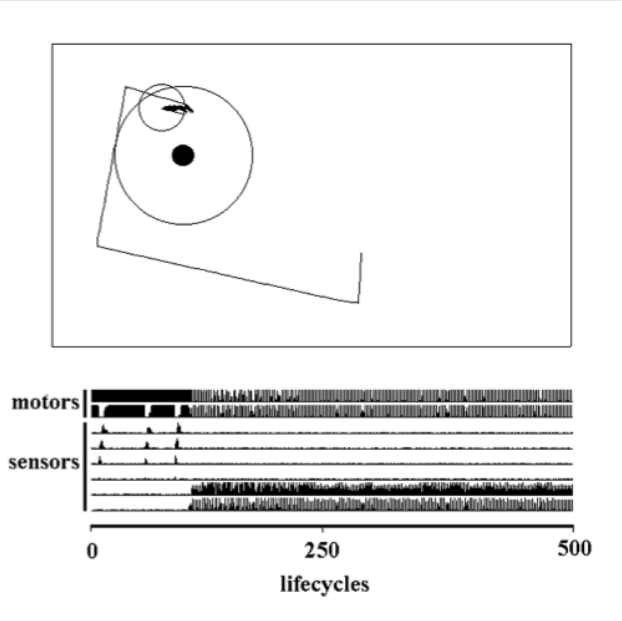
small and large cylinders

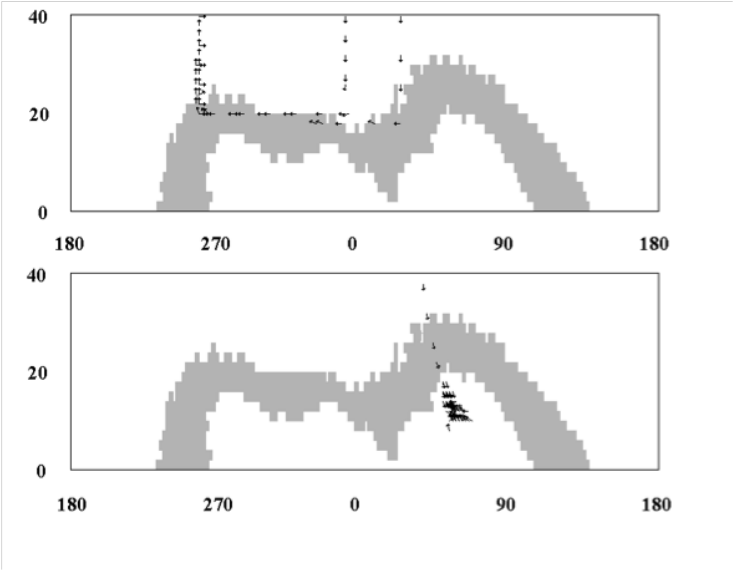
[Nolfi, 1996,1999]

Discrimination Task



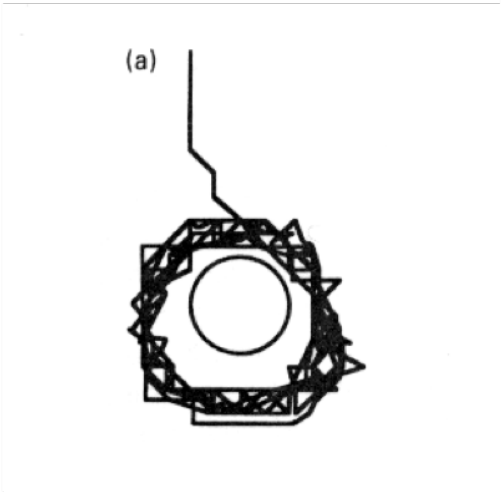


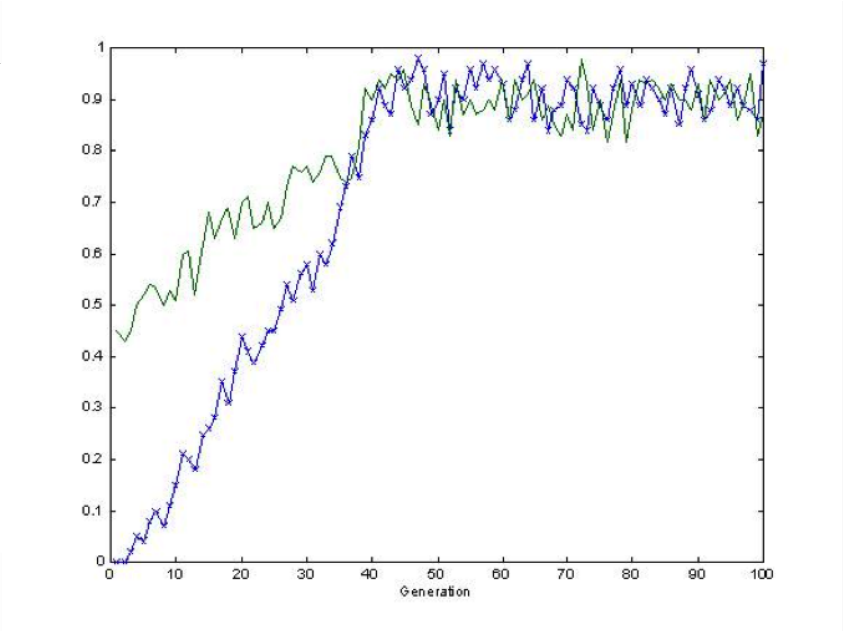




[Nolfi, 1996]

Discrimination Task

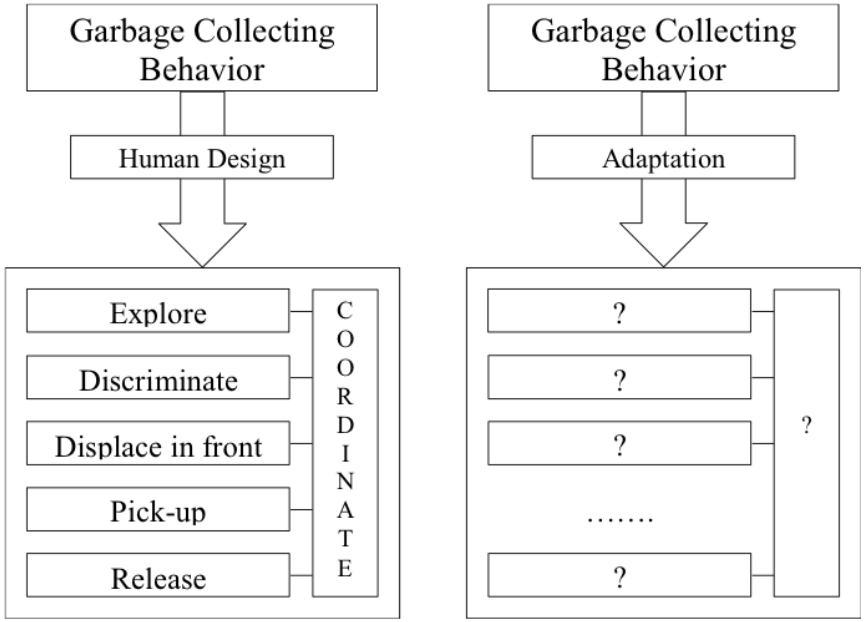




Evolved robots act so to select sensory patterns that are easy to discriminate

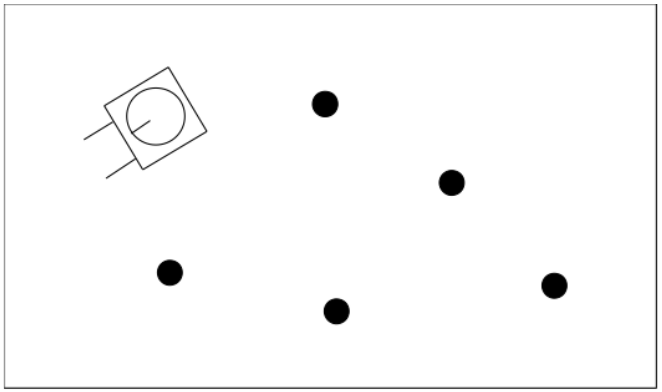
[Scheier, Pfeifer, and Kuniyoshi, 1998]

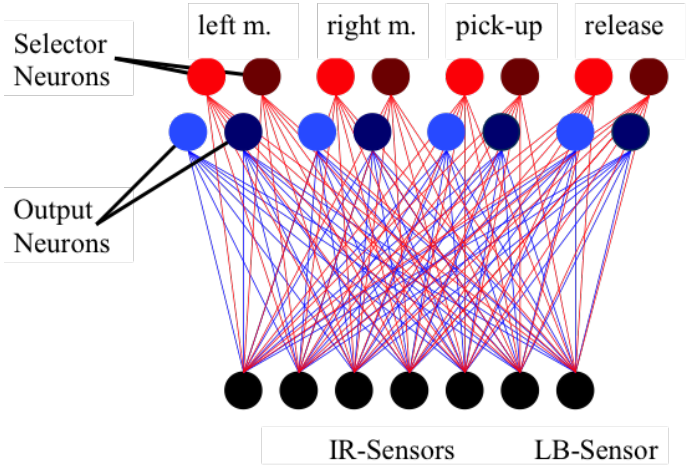
Is Modularity Useful in ER?

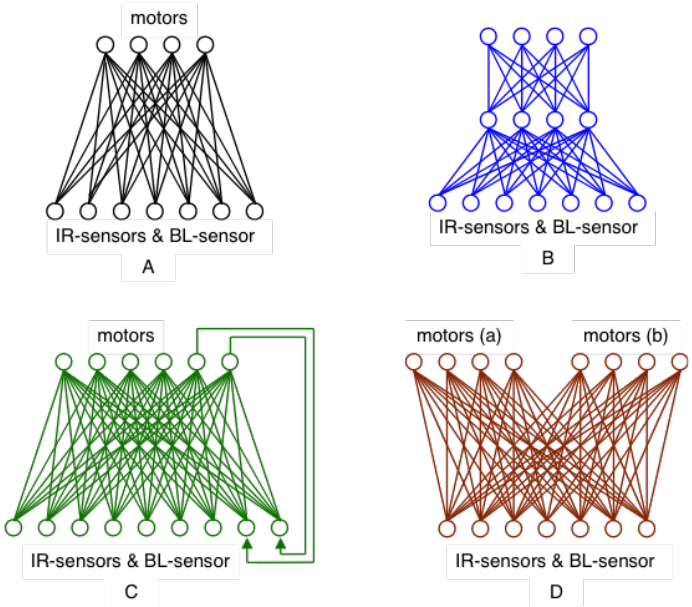


- Is modularity useful in ER?
- What is the relation between self-organized neural modules and behaviors?

The Garbage Collecting Task

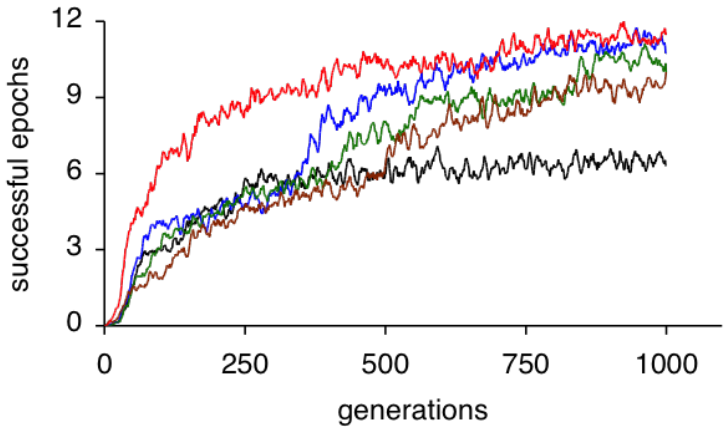




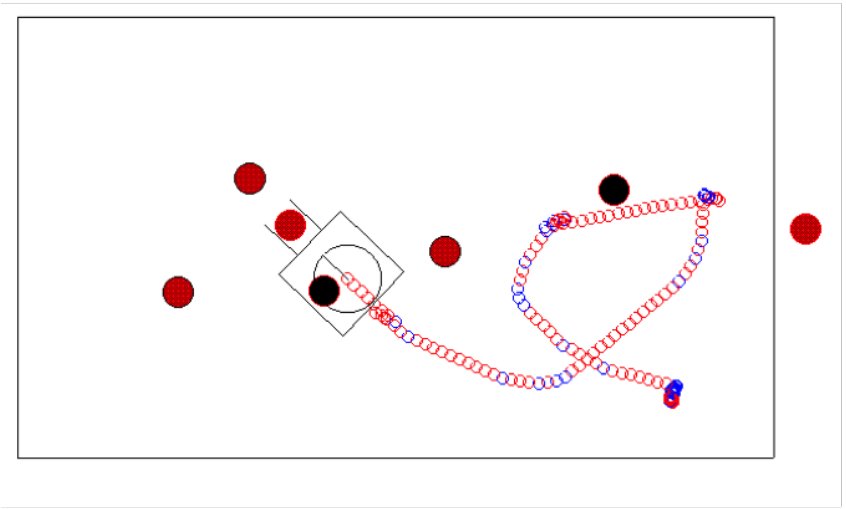


[Nolfi, 1997]

The Garbage Collecting Task



Modular neural controller able to self-organize outperform other architectures



There is not a correspondence between self-organized neural modules and sub-behaviors

[Nolfi, 1997]

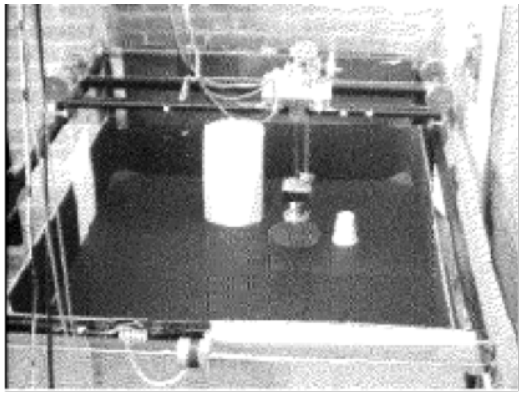
Incremental Evolution

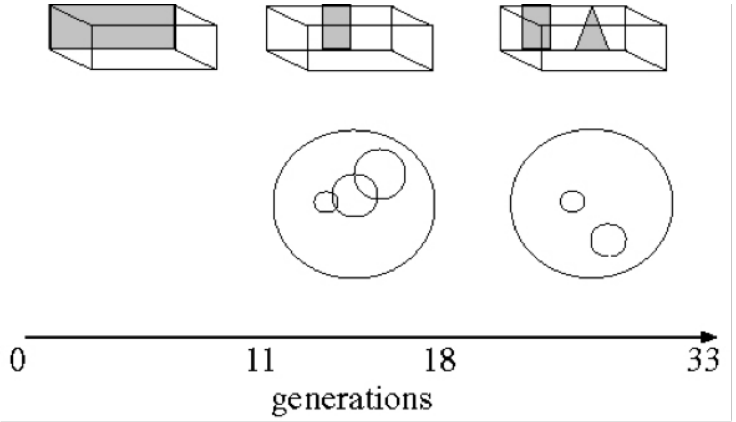
Selecting individuals directly for their ability to solve a task only works for simple tasks

Incremental Evolution: starting with a simplified version of the task and then progressively increasing complexity

- Including in the selection criterion also a reward for sub-components of the desired behavior
- Start with a simplified version of the task and then progressively increase its complexity by modifying the selection criterion

Visually-Guided Robots





[Cliff et al. 1993; Harvey et al. 1994]

Resilient Machines through Continuous Self-Modeling

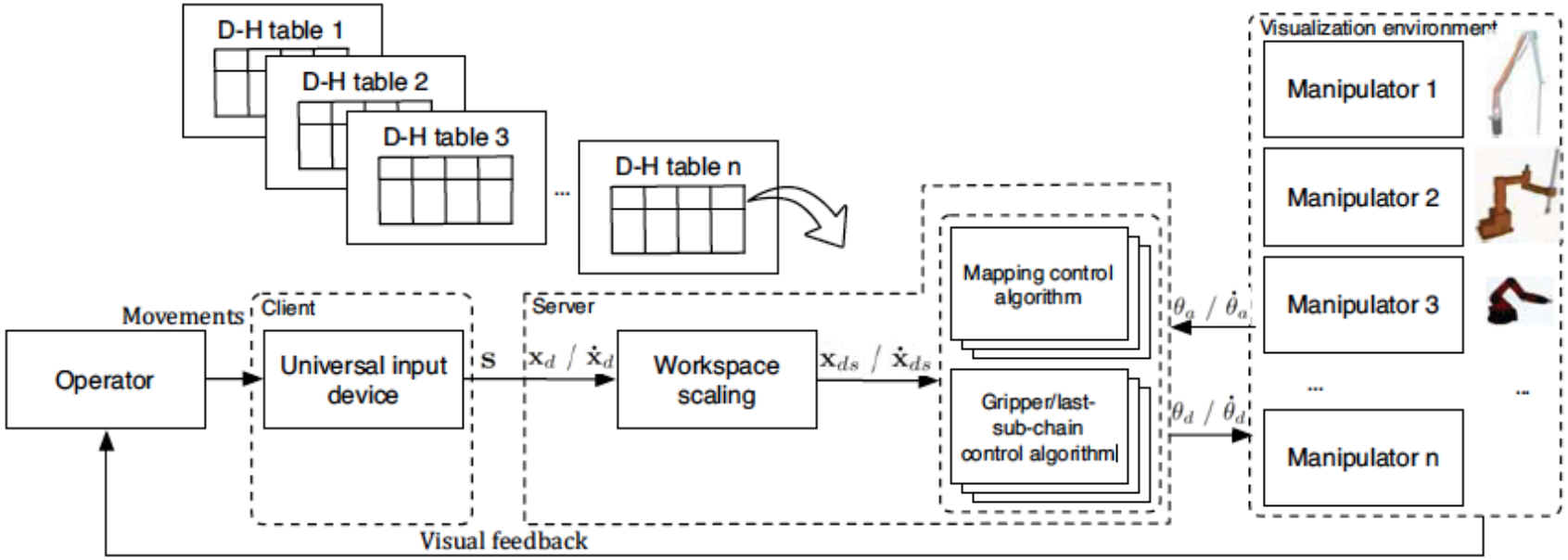
Background

Controlling robotic arms

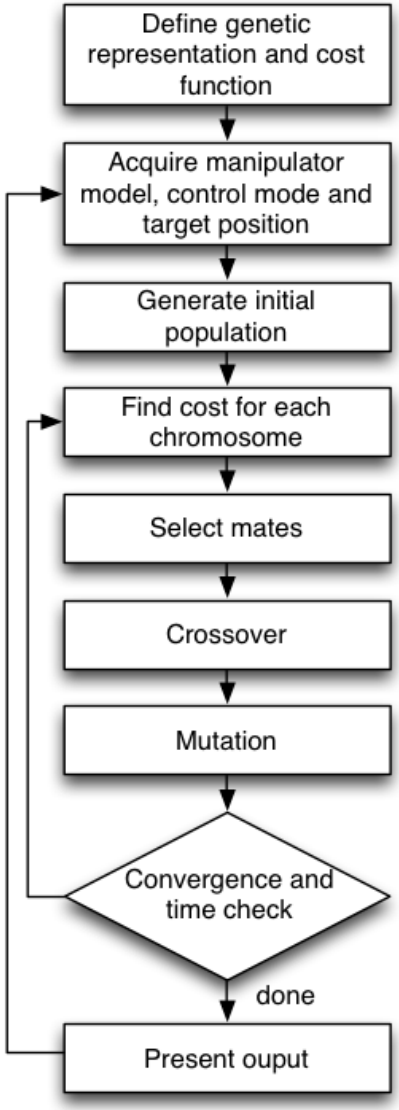
- A common assumption in most of the previous works is that the IK model of the arm to be controlled is a priori knowledge
- When considering arms with redundant degrees of freedom, the inverse kinematics can have multiple solutions, and therefore singularity problems could arise
- This method is not very flexible, especially when planning to control different arms using a universal input device because several IK models are needed

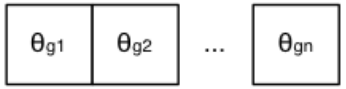
An alternative solution to the problem might consist of using methods that do not assume a priori knowledge for the IK model of the arm: a solution that derives its kinematic properties from a machine learning procedure

Background


D-H table 1
D-H table 2
D-H table 3
...
D-H table n
Movements
Operator
Client
Universal input device
s
Server
$\mathbf{x}_d$ / $\dot{\mathbf{x}}_d$
Workspace scaling
$\mathbf{x}_{ds}$ / $\dot{\mathbf{x}}_{ds}$
Mapping control algorithm
Gripper/last-sub-chain control algorithm
θ_a / $\dot{\theta}_a$
θ_d / $\dot{\theta}_d$
Visualization environment
Manipulator 1
Manipulator 2
Manipulator 3
...
Manipulator n
Visual feedback

Flowchart





$$d(\mathbf{x}_t, \mathbf{x}_a) = |\mathbf{x}_t - \mathbf{x}_a|$$

$$\mathbf{x}_t = \mathbf{x}_{ds}$$

$$\mathbf{x}_t = \mathbf{x}_a + \dot{\mathbf{x}}_{ds}\Delta t$$

Selection, Crossover and Mutation

- Select mates: the selection of candidates to be used as parents in the crossover process is obtained by using the stochastic universal sampling method [1], which is a fitness proportionate selection method.
- Crossover: the crossover function is defined as a hybrid function that stochastically switches - with a 50% crossover probability - between using a single-point and a uniform crossover method, to create new offspring from the selected parent chromosomes
- Mutation: mutation may occur in a chromosome by stochastically adding a random value of 5% to the value of its genes. In particular, there is a 0.5% mutation chance for each gene. Additionally, a form of elitism is also used and 10% of the fittest chromosomes survives unaltered between generations

[1] J. E. Baker, "Reducing bias and inefficiency in the selection algorithms", 1985

Convergence and Time Check

- The GA population stops evolving and the fittest chromosome is returned when the cost drops below 0.01 or when the overall time spent evolving the population exceeds 100 ms. Note that, since the target position is normalised according to the workspace, a value of 0.01 results in being weighed and proportionate to the workspace. A time limit of 100 ms allows the population to reach a good level of evolution in the first few steps of iterations without effecting the operator experience in terms of perception.
- After the first few iterations, the time limit is stochastically seldom reached for target positions that are located inside the boundaries of the workspace

Present Output

The genes of the fittest chromosome are then presented as output. In particular, denoting these genes as θ_f and according to the operation scenario, the output is obtained as:

$$\theta_d = \theta_f,$$

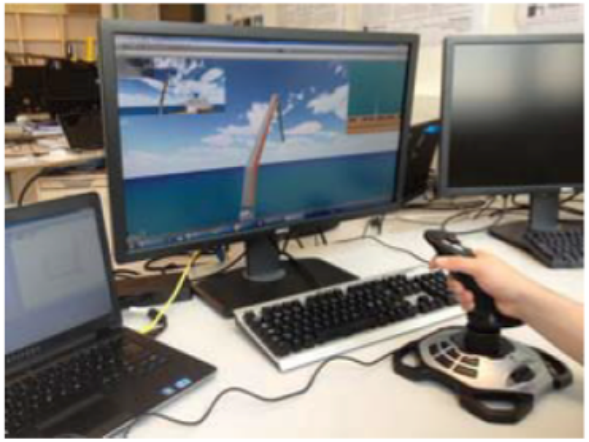
when operating in position control mode, or as:

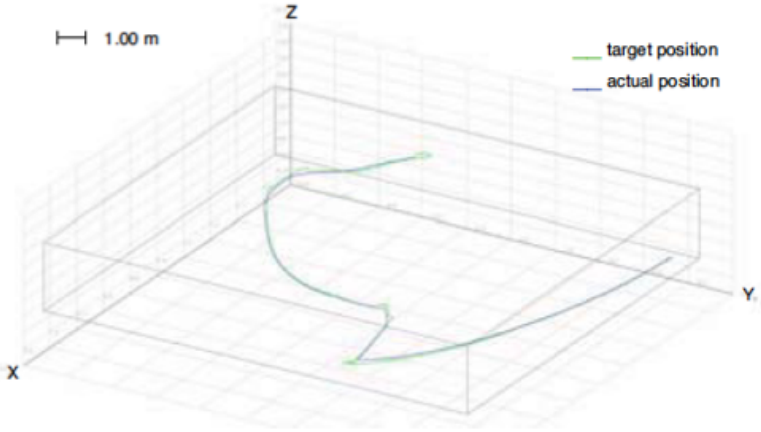
$$\dot{\theta}_d = \frac{\theta_f - \theta_a}{\Delta t},$$

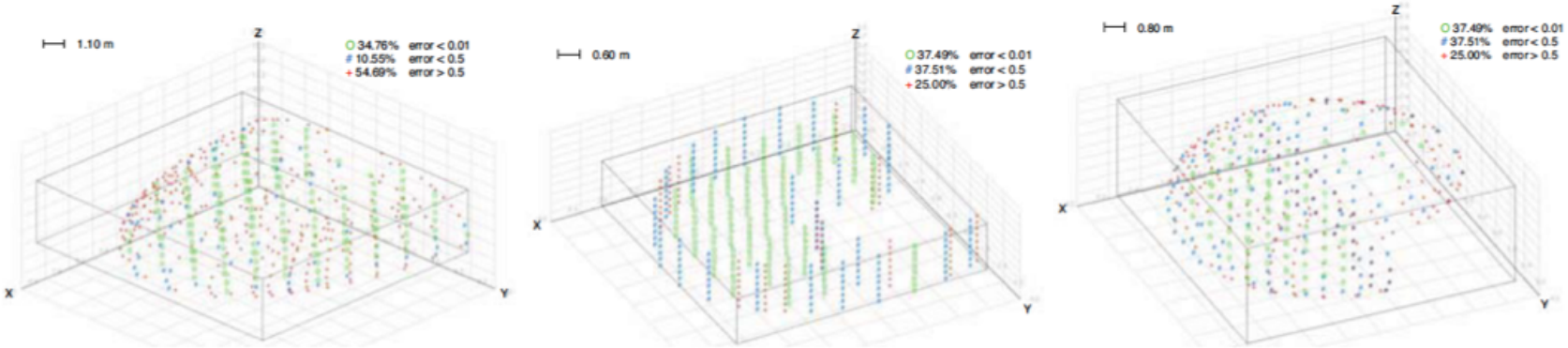
when operating in velocity control mode.

Simulations and Results

The simulation of a knuckle boom crane model and the trajectory tracking of its Cartesian paths in X, Y and Z coordinates.







Error distribution for 512 equally-spaced target positions for a the knuckle boom crane model, a SCARA robot and a KUKA youBot robot.

Simulations and Results

Conclusions

To summarize what I have learned in this course

- Evolutionary prospective and importance of self-organization
- Self-organization does not mean that an evolutionary system is entirely free of human design (fitness functions, genotype-to-phenotype mapping, evolutionary conditions, control architecture, electronics and hardware)
- Evolutionary robotics can be seen as a tool for studying behavioral systems
- Design principles may help us to design the characteristics which are not subjected to evolutionary process

Thank you for your attention