Evolutionary Re-Adaptation of Neurocontrollers in Changing Environments

Dario Floreano
Centre for Neuro-Mimetic Systems
Swiss Federal Institute of Technology in Lausanne
LAMI-INF-EPFL, CH-1015 Ecublens
E-mail: floreano@di.epfl.ch
http://diwww.epfl.ch/lami/team/floreano/

Abstract

Evolutionary robotics is an interesting novel approach to shape the control system of autonomous robots. This explores issues related to re-adaptation in changed environments of a population of evolved individuals. Experimental studies are reported for genetic evolution of neurocontrollers that have developed the ability to perform homing navigation for battery recharge of miniature mobile robot. It is shown that re-adaptation to important changes in the environment is very rapid and does not disrupt previously acquired knowledge. The results are discussed in relation to the internal representation of the neurocontroller and to the variability within the population.

1 Evolutionary Shaping of Autonomous Robots

An autonomous robot can be seen as an artificial organism capable of self-organising its own behaviour according to environmental constraints in order to maintain its own viability without human intervention. In recent years, the analogy between autonomous robots and biological organisms has generated a novel approach, also known as "Behaviour–Based Robotics" [10], to program and understand robot behaviour in unknown and unpredictable environments. This approach is quite different from the classical AI approach that attempted to emulate human reasoning by building large and complex planning systems which failed to deliver the expected results [1]. In Behaviour–Based Robotics emphasis is put on speed, robustness, low-cost, and incremental development of modular controllers. Here, researchers seek inspiration from biology, trying to understand and reproduce the smart and simple mechanisms that allow animals to survive in their own environment [13].

Typically, the control system of such robots is based on parallel and distributed processes interconnected by modifiable links. One example is the well-known subsumption architecture [2]. Several successful results have been also reported using neural networks; e.g., see [15, 14, 3, 11] for an overview of the field. Neural networks, in particular, offer interesting advantages if one is interested in biological inspiration and adaptation: **a)** They facilitate knowledge-tranfer

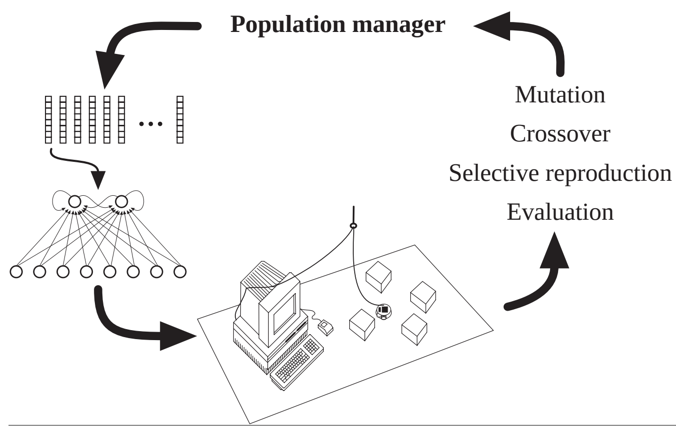


Figure 1: Genetic evolution on a single robot. Between one individual and the next the robot is allowed to perform a random motion for 5 seconds.

and comparison between biological and artificial organisms; **b)** Several learning algorithms are available for modifying internal processing parameters; **c)** They display intrinsic generalisation abilities. A large number of people have used genetic algorithms for evolving the appropriate internal parameters of robot neurocontrollers (see [12] for a comprehensive overview). Genetic algorithms [9] represent a family of optimisation techniques inspired upon natural evolution.

A genetic algorithm operates on a population of strings, each of them representing a solution of the problem, that is the parameters defining a neurocontroller for the robot (Figure 1). It is necessary to define a *fitness function*, that is a measure of robot performance, and a set of decoding rules for mapping the genotypic string into the neural network phenotype. Initially, all the strings are randomly generated and each is in turn decoded, downloaded on the robot, and tested while the fitness value is measured. Then, three genetic operators –selective reproduction, crossover, and mutation– are applied in order to create a novel generation of strings. Selective reproduction allocates a number of copies for each string proportional to its fitness value, crossover swaps parts of the genetic material between randomly paired strings, and mutation randomly changes the contents of each gene with a certain low probability. The process is repeated until the average performance of the population and the performance of the best individual reaches reaches a satisfactory value (see [7] for an introduction to genetic algorithms). An interesting feature of genetic algorithm is that it is not necessary to specify in detail the desired actions for every possible sensory stimulation; rather, only a general description of the expected behaviour can be used to design the fitness function. This is of great advantage in autonomous robotics where correct motor actions are not known in advance making thus supervised learning algorithms unsuitable.

To date, most of the experiments reported in the literature have been done using simulated robots. Although those findings are certainly relevant, we believe that computer models of robots cannot capture the complexity of the interaction between a real robot and a physical environment where mechanical and physical laws (such as wearing of the components, changing light conditions, friction, etc.), non-white noise at all levels, and various types of potential hardware malfunctioning play a major role.[1] Here, I shall focus on what is now recognised as the first experiment where the evolutionary process has been entirely carried out on a real mobile robot without human intervention [5]. In section 2 I shall briefly describe the basic setup and results; in section 3 I shall focus on novel results where important environmental conditions are changed; finally, in section 4 I shall discuss the results and shortly describe current directions.

2 Homing for Battery Charge: A case-study

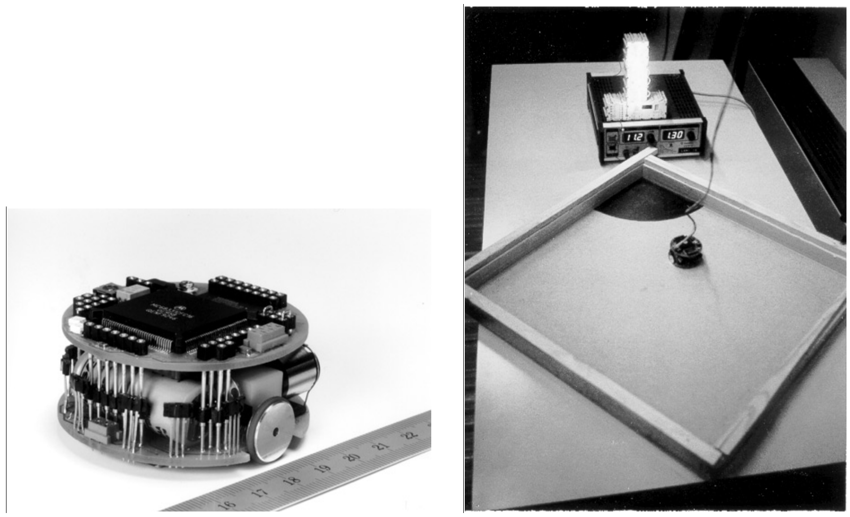
In the experiment here described we attempted to evolve a homing behaviour for battery recharge using the miniature mobile robot Khepera. Our goal was that of of testing the hypothesis that complex behaviours do not necessarily have to be specified in the fitness function, but can rather develop in order to satisfy a more general constraint [5]. Therefore, neither the fitness function nor the neurocontroller incorporated explicit knowledge about the presence and location of the battery charger.

In this experiment we employed the miniature mobile robot Khepera [17], which has a diameter of 55 mm, it is 30 mm high, and its weight is 70 g (Figure 2a). The robot is supported by two wheels and two small Teflon balls placed under its platform. The wheels are controlled by two DC motors with an incremental encoder (12 pulses per mm of robot advancement) and can rotate in both directions. In the basic version used here, it is provided with eight infrared proximity sensors placed around its body (six on one side and two on the opposite one) which are based on emission and reception of infrared light. Each receptor can measure both the ambient infrared light (which in normal conditions is a rough measure of the local ambient light intensity) and the reflected infrared light emitted by the robot itself (for objects closer than 4-5 cm in our experiments). An additional sensor was placed under the robot platform to detect floor brightness. The robot was also provided with a simulated battery characterized by a fast linear discharge rate (max duration: approx. 20 seconds), and with a simulated sensor giving information about the battery status (we could have used the on-board batteries, but the evolutionary procedure would have lasted 6 years, instead of 10 days).

The environment employed for the evolutionary training consisted of a 40x45 cm arena delimited by walls of light-blue polystyrene and the floor was made of thick gray paper (Figure 2b). A 25 cm high tower equipped with 15 small DC lamps oriented toward the arena was placed in one corner. The room did not have other light sources. Under the light tower, a circular portion of the floor at the corner was painted black. The painted sector, that represented the recharging area, had a radius of approximately 8 cm and was intended to simulate the platform of a prototype of battery charger under construction. When the robot happened to be over the black area, its simulated battery became instantaneously recharged.

A multilayer perceptron of continuous sigmoid units was used to map sensor inputs into motor outputs. 12 input units clamped to 8 infrared sensors, 2

[1] But see [16] for a clever methodology that bridges the gap between simulation and implementation in certain conditions.



a) b)

Figure 2: a) Khepera, the miniature mobile robot. b) The environment with the light tower and the robot (there were no other light sources in the room).

ambient light sensors (one on the front and one on the back side of the robot), 1 floor-brightness detector, and one battery charge sensor, were fully connected to 5 hidden units with recurrent connections [4]; hidden units were fully connected to 2 output units, each controlling the speed and rotation direction of the two wheels. Each robot (corresponding to one string of the population) started its life with a fully charged battery which was discharged by a fixed amount at each time step: a fully charged battery allowed a robot to move for 50 time steps. If the robot happened to pass over the black area the battery was instantaneously recharged and, thus, its life prolonged. An upper limit of 150 steps was allowed for each individual, in order to eventually terminate the life of robots that remained on the recharging area or that regularly passed over it.

Each genetic string coded real values of synaptic strengths and neuron thresholds of the neural network (a total of 102 genes per string). The population size was kept constant to 100 individuals per generation, the crossover and mutation probability were 0.1 and 0.2 respectively. Each motor action lasted 380 ms (including time for serial communication with the workstation). Each decoded individual was evaluated during its life according to the following fitness function Φ,

$$\Phi = V(1-i), \qquad 0 \leq V \leq 1,\ 0 \leq i \leq 1 \tag{1}$$

where V is a measure of the average rotation speed of the two wheels and i is the activation value of the proximity sensor with the highest activity. The function Φ has two components: the first one is maximized by speed and the second by obstacle avoidance. The accumulated fitness value of each individual (which depended both on the performance of the robot and on the length of its life) was then divided by the maximum number of steps (150) and stored away

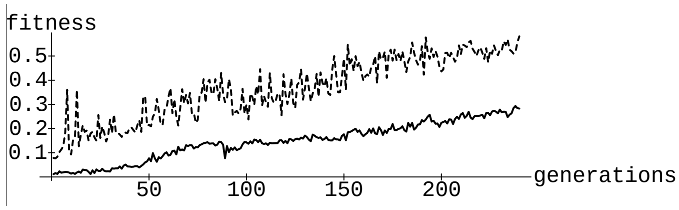


Figure 3: Average population fitness (continuous line) and fitness of the best individual (dotted line) at each generation.

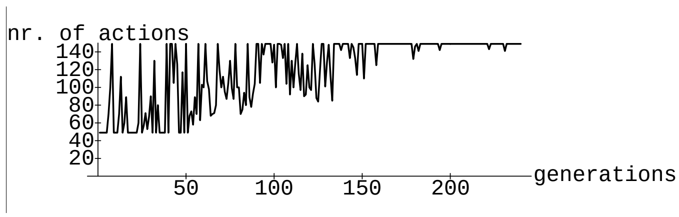


Figure 4: Number of actions during life for the best individual at each generation. 50 actions (approximately 20 seconds) represent the minimum life length because each individual starts with a full battery.

for the genetic operators. It should be noted that locating and passing over the recharging area is not treated as one of the main goals that the robot should achieve, but only as a possible behavioral strategy that could emerge to exploit the characteristics of the robot and of the environment.

The robot was left alone in a dark room lit only by the small light-tower while we monitored its evolution on our workstation in another room for the next 10 days. Both the population average-fitness and the fitness of the best individual steadily increased along the corresponding 240 generations (Figure 3). Accordingly, the number of steps (actions) taken by the best neurocontroller increased along generations (Figure 4). By combining the data in Figure 3 and Figure 4, it is possible to notice that, especially in the last 90 generations, increased their own life duration and spent shorter periods of time over the recharging area (no fitness was returned when the robot was on the charging area!). An analysis of the behaviour and of the neurocontroller internal dynamics showed that the best individual of the last generation spent most of his energy wandering around the environment without hitting the walls; when the battery was almost discharged (exactly *when* depended on robot location and remaining energy), the robot quickly returned to the charging station and rapidly returned to the environment once the battery had been recharged (see [5] for further details and analysis).

3 Changing Environmental Layout

On the left side of Figure 5 one can see the trajectory of the best individual of generation 240 for the initial 50 actions. The robot starts with a fully charged

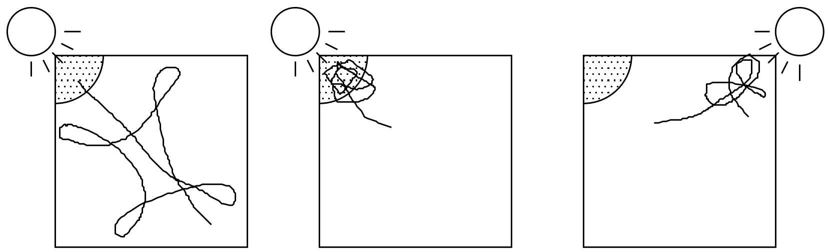


Figure 5: Trajectories of best individual of generation 240 in three environmental conditions. Left: Test in training conditions. The robot starts with a full battery in the bottom right corner (only the first 50 actions are displayed). Centre: The battery is not automatically recharged when the robot arrives on the charging area. The robot starts in the centre of the environemnt with an almost discharged battery. Right: The light source is positioned on the top right corner, but the charging area remains at the original location.

battery in the lower right corner of the environment and moves around the environment avoiding the charging area (where it does not receive any fitness value) until the battery reaches a minimum level; then, it heads straight to the charging area where it always arrives with approximately 2% residual energy. Discovery of the charging area location, straight paths, and calculation of residual energy as function of robot location are abilities that were not specified in the fitness function, but emerged in order to maximise the selective reproduction criterion of keeping as long as possible the wheels rotating without hitting the walls (see [5] for a description of the internal world map developed by the neurocontrollers). Therefore, these abilities can be seen as sub-goals finalised to the satisfaction of the final goal described in the fitness function.

In this experiment, the position of the light source is the only landmark that can be exploited to locate the charging area. The robot has learned to associate certain values of light intensity and direction with the charging area location. Two simple tests show this. If the battery is not automatically re-charged when the robot arrives to the area, it will still stay on it exploring the it thoroughly until all the residual energy is exhausted (Figure 5, centre). Similarly, if the light source is moved to the top right corner (but the charging area is not moved), the robot will head toward that corner and stay in the surroundings until all the energy is exhausted. However, now exploration of the corner will consist of larger trajectories, probably because the neurocontroller is actively looking for the black surface (Figure 5, right).

Although these behaviours are interesting, it would be desirable for an autonomous robot to re-adapt to changing environmental conditions in order to maintain its own viability. Adaptation in the original environment required 240 generations, corresponding to approximately 250 hours of continuous operation. How long will it take –if possible at all– to re-adapt to important environmental changes? Since here the most important environmental variable is the position of the light source relative to the charging area, evolutionary training was continued in three different environmental conditions, each with the light source positioned in a different corner of the environment (top-right, bottom-right, and

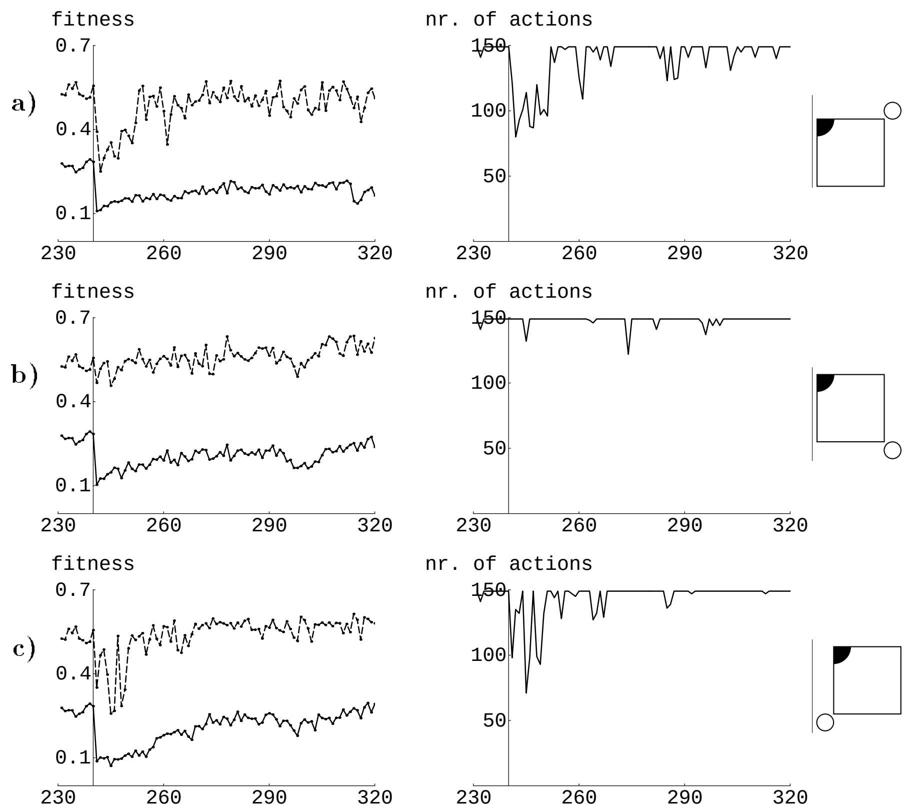


Figure 6: Re-adaptation in environments with a new light position. Each row (a, b, and c) plots –respectively– the average population fitness (continuous line) and the fitness of the best individual (dotted line) across generations, the number of actions during life for the best individual at each generation, and a sketch of the light position (small circle) in the environment (the black sector represents the charging area). For sake of comparison, each plot includes data for the last ten generations of the original run (see Figs. 3 and 4); therefore, data on re-training start at the origin of the y-axis.

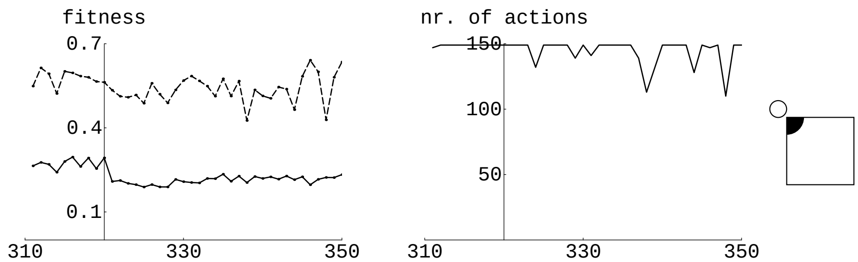


Figure 7: Re-training on the original environment of the last population of individuals adapted to the light spot positioned in the low-left corner (Fig. 6c).

bottom-left; called *a*, *b*, and *c*, respectively). For each condition, the genetic algorithm was restarted on the population of generation 240 and continued for 80 additional generations with the same parameters already described in section 2. Data for all conditions are displayed in Figure 6. A comparison of these data with the data recorded from the original training session puts in evidence three main results. The initial drop in performance, both at the level of the population and of the best individual, is not dramatic; this is probably due to the presence of several individuals which were "sub-optimal" in the original environment, but resulted indeed fitter in the new environment. Re-adaptation took place relatively quickly; in the worst cases (Figure 6, *a* and *c*), 20 generations (less than 10% of the time required for initial training) were sufficient to create an individual perfectly adapted to the new environment which reported the same performances already measured for the best individual of generation 240 in the original environment. Re-adaptation was extremely rapid when the light source was positioned in the corner opposite to charging area (Figure 6, *b*); this indicates that the mirror symmetry of the new environment does not require a drastic change in the internal representation developed by the neurocontrollers. In other words, whereas conditions *a* and *c* require a rotational translation of the internal world map involving changes in several synaptic weights, condition *b* can be successfully solved by changing few important synaptic weights which result in a complete reversal of the internal map. If this is the case, then one expects that only the best individuals (i.e., those which already developed a correctly oriented map in the original environment) would benefit from mirror symmetry of the new environment; this is indeed shown by the sharp contrast between the performance of the best individual and the average performance of the population which displays the same initial drop and recovery rate as in conditions *a* and *c*.

Another question is whether re-adaptation in a changed environment has cancelled the behavioural strategies acquired in the original environment and –if this is the case– what is the amount of additional training necessary to restore them. The genetic algorithm was thus restarted on the original environment from the population of the last generation of individuals re-adapted to condition *c*; and continued for 30 generations; the population of condition *c* was chosen because it was one of the two cases which required longer re-adaptation. All evolutionary parameters were kept constant. As it can be seen in Figure 7, the initial drop in performance is minimal both at the level of the best individual and at the level of the population. This result indicates that internal behavioural knowledge acquired during the original adaptation phase has not been considerably disrupted by re-adaptation to a quite different environmental layout. Although here behavioural knowledge is stored at the level of the population, the system as a whole is capable of rapid reconfiguration while preserving previously acquired representations.

4 Discussion

The results discussed above raise a number of issues. Perhaps, the most important one concerns variability of the population along generations. Variability of individuals within the population represents the combustion material that drives genetic adaptation. When internal variation is exhausted, there cannot be any further adaptation. The reason why the population of generation 240 could re-adapt to the new environments is that it had not yet converged (average fitness performance had not yet reached a plateau level). It is reasonable to assume that further 80 generations of retraining reduced the population variability; thus, the possibility to have further re-adaptation after generation 320 (240 plus 80) is smaller than after generation 240. Indeed, this is what happens when the population of generation 320 is retrained in the original environment (Figure 7). Here, the average performance of the population does not increase along generations. The reason why the best individuals can achieve the task is that previous knowledge about the charging area location was not erased from the population. These consideration make one think that re-adaptation is not an open-end property of traditional genetic algorithms where single-task optimisation is the main objective. It is thus necessary to look for different approaches. One possibility is that of using variable-length genotypes for continuously evolving populations [8]. A different approach is that of evolving learning rules that modify the internal parameters of the neural network while the robot interacts with its own environment [6]. Initial experiments in the latter case have generated learning networks capable of configuring the robot behaviour in few seconds according to the goal described in the fitness function and to the actual properties of the environment. These results seem promising because they could provide a way to reduce the adaptation time requested by the traditional evolutionary approach. However, these experiments were carried out in static environments and require further tests in order to assess the real potentials of the approach.

References

[1] R. A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15, 1990.

[2] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–59, 1991.

[3] D. Cliff, P. Husbands, J.-A. Meyer, and S. W. Wilson. *From Animals to Animats 3 – SAB94*. MIT Press – Bradford Books, Cambridge, MA, 1994.

[4] J. L. Elman. Finding Structure in Time. *Cognitive Science*, 14:179–211, 1990.

[5] D. Floreano and F. Mondada. Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26:396–407, 1996.

[6] D. Floreano and F. Mondada. Evolution of plastic neurocontrollers for situated agents. In P. Maes, M. Mataric, and J-A. Meyer, editors, *From Animals to Animats IV: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA, 1996.

[7] D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Reading, MA, 1989.

[8] I. Harvey. Species Adaptation Genetic Algorithms: A basis for a continuing SAGA. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life.* MIT Press-Bradford Books, Cambridge, MA, 1992.

[9] J. H. Holland. *Adaptation in natural and artificial systems.* The University of Michigan Press, Ann Arbor, 1975.

[10] P. Maes. Behavior-based artificial intelligence. In J. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA, 1993.

[11] P. Maes, M. Mataric, and J-A. Meyer. *From Animals to Animats 4 – SAB96.* MIT Press – Bradford Books, Cambridge, MA, 1996.

[12] M. Mataric and D. Cliff. Challenges in Evolving Controllers for Physical Robots. *Robotics and Autonomous Systems*, 1996. In press.

[13] D. J. McFarland and T. Boesser. *Intelligent Behavior in Animals and Robots.* MIT Press/Bradford Books, Cambridge, MA, 1993.

[14] J.-A. Meyer, H. L. Roitblat, and S. W. Wilson. *From Animals to Animats 2 – SAB92.* MIT Press – Bradford Books, Cambridge, MA, 1993.

[15] J.-A. Meyer and S. W. Wilson. *From Animals to Animats – SAB90.* MIT Press – Bradford Books, Cambridge, MA, 1991.

[16] O. Miglino, H. H. Lund, and S. Nolfi. Evolving Mobile Robots in Simulated and Real Environments. *Artificial Life*, 1996. in press.

[17] F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturization: A tool for investigation in control algorithms. In *Proceedings of the Third International Symposium on Experimental Robotics*, Kyoto, Japan, 1993.