Evolutionary Neurocontrollers for Autonomous Mobile Robots

Dario Floreano[1] and Francesco Mondada[1,2]
[1]Laboratory of Microcomputing (LAMI),
Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne
[2]K-Team SA, CH-1028 Préverenges, Switzerland

Abstract

In this article we describe a methodology for evolving neurocontrollers of autonomous mobile robots without human intervention. The presentation, which spans from technological and methodological issues to several experimental results on evolution of physical mobile robots, covers both previous and recent work in the attempt to provide a unified picture within which the reader can compare the effects of systematic variations on the experimental settings. After describing some key principles for building mobile robots and tools suitable for experiments in adaptive robotics, we give an overview of different approaches to evolutionary robotics and present our methodology. We start reviewing two basic experiments showing that different environments can shape very different behaviors and neural mechanisms under very similar selection criteria. We then address the issue of incremental evolution in two different experiments from the perspective of changing environments and robot morphologies. Finally, we investigate the possibility of evolving plastic neurocontrollers and analyze an evolved neurocontroller that relies on fast and continuously changes synapses characterized by dynamic stability. We conclude by reviewing the implications of this methodology for engineering, biology, cognitive science, and artificial life, and point at future directions of research.

Keywords: Artificial Evolution, Autonomous Mobile Robots, Neural Networks, Evolution and Learning, Robot Learning, Machine Learning, Bio-Inspired Machines, Robot Navigation.

1 Introduction

In this article we give an overview of our approach to artificial evolution of neural controllers for autonomous mobile robots. The presentation, which spans from technological and methodological issues to experimental results on evolution of physical mobile robots, covers both previous and recent work in the attempt to provide a unified picture within which the reader can compare the effects of systematic variations on the experimental settings.[1]

Autonomous mobile robots are machines that are expected to operate in partially unknown and unpredictable environments. Therefore, in contrast to robots used in highly controlled and constrained environments, autonomous robots cannot be fully pre-programmed to carry out a predefined set of actions because one does not know in advance the universe of sensorimotor mappings required by all the situations that the robot might encounter.

Biological organisms can be a source of inspiration for design and control of autonomous mobile robots (Steels, 1995). Animals and humans are indeed autonomous agents that display robust adaptation and stable behavior in changing environments with minimal, or without, external supervision and control (McFarland & Boesser, 1993). In several circumstances, it might be desirable to create autonomous robots which possess features similar to those of biological autonomous agents. Without going as far as modeling robots directly on animals or humans (Brooks, Breazeal (Ferrell), & Irie, 1998), biological organisms can inspire the development of autonomous robots with respect to a set of basic principles. These include the nature of the adaptation mechanisms, such as philogenetic evolution and ontogenetic learning, the preference for behavioral stability and robustness over precision, self-organization and self-selection of goals and values, and adaptation while interacting with an environment.

Along similar lines, but from a different standpoint, autonomous robots represent both a tool and a metaphor for developing and testing models of adaptive behavior and cognitive abilities (McFarland & Boesser, 1993; Pfeifer & Scheier, 1998). Autonomous robotic agents, just as biological agents, act in a physical environment where the consequences of their own actions constantly affect their sensory input and therefore their future actions (Parisi, Cecconi, & Nolfi, 1990), effectively opening the way to a new ecological approach to the study of artificial intelligence (Pfeifer & Scheier, 1998).

[1] Video clips of all the experimental results described in this article are accessible from the WWW page `http://diwww.epfl.ch/lami/team/floreano`.

Developing adaptive autonomous robots requires a careful methodology based on integration of control issues, hardware design, and technological choices. In the next section we shall describe some key issues in design and technologies for adaptive autonomous robots that are at the basis of our work. We will then introduce a methodology for applying artificial evolution to physical robots and describe an array of experiments on mobile robots that capitalize on the basic principles of biological adaptation mentioned above. We shall show that a variety of complex behaviors can autonomously emerge from the interaction between the robot and the environment without human intervention and with few externally-imposed constraints. This paper does not aim at providing definitive answers and proofs on the mechanisms of adaptive behaviors, but simply at illustrating what we think are promising directions of research from an engineering, biological, and cognitive perspective. Our goal is to show that artificial evolution is not only a powerful technique for engineering adaptive autonomous robots, but also a rich tool for investigating new hypothesis and gaining insights into the biology and technology of adaptive behavior.

2 Mobile Robots

Simulation studies have for long time been considered a valid investigation methodology for developing autonomous robot controllers. Despite recent widespread agreement on the importance of using physical robots for development and test of new models (Brooks, 1986, 1991; Dorigo & Schnepf, 1993; Franceschini, Pichon, & Blanes, 1991; Pfeifer, 1995; Steels, 1994), real mobile robots are used only by a minority of researchers. Some of the reasons might be practical. For example, most of the researchers working in the field of adaptive behavior come from various disciplines, such as biology, psychology, neurophysiology, and mathematics, and are not always very keen to fiddle with mechanical and electronical problems that mobile robots might give. Furthermore, physical robots are usually large and fragile. Therefore, they require large dedicated spaces and could be badly damaged by bio-inspired controllers that go through an adaptation phase based on trials and errors.

———————————

INSERT FIGURE 1 ABOUT HERE

———————————

From this perspective, mobile robots to be used for research and development of bio-inspired adaptive systems should be robust, relatively compact, reliable, easily interfaceable to standard computer platforms and software, and equipped with solutions for power supply and behavioral monitoring. The Khepera miniature mobile robot was originally developed at our laboratory by Franzi, Guignard, and Mondada as a tool to carry out research in adaptive and bio-inspired control (Mondada, Franzi, & Ienne, 1993). In its basic configuration (figure 1), Khepera consists of two boards: the sensorimotor board and the CPU board. The motor system uses two lateral wheels and supporting pivots in the front and back. This configuration allows rotation of the body without lateral displacement. The sensory system uses eight active infrared-light sensors distributed around the body, six on one side and two on the other (this asymmetry can be used to establish the front and back of the robot). These sensors can detect the presence of objects by emitting and measuring reflected light (the closer the object, the stronger the response), but can also be used to measure the infrared component of ambient light. Four rechargeable NiCd batteries with a total autonomy of approximately 30-40 minutes are secured on the sensorimotor board. The CPU board encloses the robot's main processor (a Motorola MC68331 with 128 Kbytes of EEPROM and 256 Kbytes of static RAM), an A/D converter for the acquisition of analog signals coming from the sensorimotor board, and an RS232 serial-line miniature connector which can support data transmission and power supply from an external computer.

In order to meet the criteria detailed above, the Khepera robot has been designed on the basis of the following criteria: miniaturization, modular open architecture, expandibility, interface, and compatibility with larger robots.

Robot *miniaturization* can bring several advantages, if realized in the appropriate proportions. The experimenter can build complex environments on a limited surface. For a miniature robot like Khepera which measures 55 mm in diameter, a standard office desk of 0.9 m x 1.8 m represents a working surface equivalent to that of a tennis court for a standard-size robot with a diameter of 55 cm. A compact working surface also allows an easier monitoring of the robot behavior. Fundamental laws of physics give higher mechanical robustness to a miniature robot. In order to intuitively understand this physical phenomenon, compare a robot of 50 mm in diameter crashing against a wall at 50 mm/s with a robot of 1 m in diameter crashing against the same wall at 1 m/s. The miniature robot will resist the collision, the other robot will probably report serious damages. However, miniaturisation brings also some drawbacks, such as the difficulty of mounting large devices on the robot (ultrasonic sensors, laser range finders, etc.).

———————————

INSERT FIGURE 2 ABOUT HERE

A *modular open architecture* enables different possible configurations and experiments using the same basic components. It also means possibilities for extensions that are developed for specific research goals, such as bio-morphic sensors or actuators. To this end, Khepera has an extension bus that makes it possible to add turrets on the top of the basic configuration, depending on the needs of the experiments which one wishes to perform (figure 2). This modularity is based on a parallel and a serial bus. The parallel bus can be used for simple extensions directly under control of the main Khepera processor. The serial bus implements a local network for inter-processor communication. Using this second bus, other processors can be connected to the main one in order to build a multi-processor structure centred on the Khepera main processor. This kind of structure has the advantage that one can employ additional computational devices on extension modules, thus keeping the main processor free for global management of the robot behaviour.

This is the case, for example, of the localization module which can compute the robot istantaneous position and communicate the result to the main processor on the robot. Evolved controllers cannot fully understood by isolating it from the environment and looking at their structures because several solutions are intimately related to the physical interactions between the robot and the environment. In order to facilitate analsys, an external positioning system was developed for monitoring the behavior of the robot and correlating it with the dynamics of the neurocontroller being tested. A device emitting laser beams at predefined angles and frequencies was positioned on the top of the environment and the Khepera was equipped with an additional turret capable of detecting laser beams and computing in real-time the robot displacement. This computation was carried out on a private processor placed on the additional turret. Every 300 ms robot position and controller variables were sent to a programmable acquisition software (Cheneval, 1995) on the workstation which instantaneously processed and visualized data while the robot freely interacted with the environment.

In order to facilitate its *interface*, the Khepera robot has been designed so that it can be attached to any computer through a serial connection and rotating contacts. The serial connection provides electrical power and supports fast data communication (up to 57kBaud) between the robot and the workstation. The user can easily control the robot interactively or from a program while it moves nearby on the desktop as if it was a simulated robot. Alternatively, one can download the code on the microcontroller of the robot and use the cable only for power supply or data reports, thus exploiting the CPU power and disk storage of the workstation to record all the necessary data while the robot freely interacts with its own environment, and later analyze them. This setup is quite useful for testing adaptive controllers that require extended periods of time, such as in the evolutionary approach.

Once a model has been developed and tested on a miniature robot, it might be desirable to test it on a different platform. For example, on might wish to extend the architecture to open environments, rough terrains, or simply incrementally evolve competencies for increasingly more complex robots. Crossplatform *compatibility* in mobile robotics is a rare feature, especially between robots of very different sizes. Software compatibility alone is not sufficient to guarantee a good transfer of algorithms between different robots. Mechanical structure, sensor caracteristics, and many other aspects of the physical platform also play an important role in the feasibility of the transfer.

INSERT FIGURE 3 ABOUT HERE

With this goal in mind, the Koala robot (figure 3) has been designed to support transfers from the Khepera robot. Despite the different size, shape, and sensorimotor configuration, the Koala robot is very similar to the Khepera in many essential aspects. The six wheels of Koala are driven by two motors, as for Khepera, each controlling one side of the platform. The central wheels on the two sides are lower than the others, so that rotation of the whole platform is very similar to that of the Khepera. The proximity sensors of the Koala are based on the same concept used for those of the Khepera, but the measurement range is scaled up to the larger size of the Koala. Also, the number of sensors has been changed from 8 on the Khepera to 16 on the Koala. The Khepera hardware modularity described above has also been both supported and scaled up in the Koala. In addition to the serial extension bus of the Khepera, the Koala is equipped with a fast inter-processor parallel communication bus to support transfer of larger amounts of data. At the software level, the two robots are compatible, having the same low-level BIOS software and the same communication facilities.

3 Artificial Evolution

In the experiments described in this paper, we have employed a combination of evolutionary computation and neural networks. The emphasis here is not on reproducing specific performances of living organisms or detailed biological functions, but rather on investigating how behavioral autonomy and free interaction can support the development of complex, robust, and stable behaviors in partially unknown and changing environments.

An evolutionary algorithm (e.g., Holland, 1975; Koza, 1992; Schwefel, 1995) searches the space of solutions (*phenotypes*) encoded on strings (*genotypes*) of finite length by selecting, reproducing, and recombining the best strings in a population for several generations until an optimal solution is found. Typically, evolutionary algorithms are used to find optimal parameter values for a function, where the value returned by the function of the encoded parameters is translated into the probability of selecting and reproducing the best strings (Goldberg, 1989).

In *evolutionary robotics*, genetic strings encode some characteristics of the controller, and the selection criterion (*fitness function*) is based solely on the behavior of the robot. This means that the feedback required by artificial evolution could be very simple and sparse, for example only whether a mobile robot has reached a target location. Since artificial evolution is based on selective reproduction and random variations of encoded strings, various characteristics of the controller (or even the morphology of the robot (Lund, Hallam, & Lee, 1997)) can be genetically encoded and co-evolved. Only behavioral success drives the progressive discovery of solutions that ultimately depend on the interactions between the robot and its environment. Other adaptive techniques, such as supervised learning, gradient descent, or implementation of dedicated circuits, require more information on the environment, the robot, and the desired behavior. Therefore, although they might generate powerful controllers, they are limited to situations where this information is available.

3.1 Related work

The idea of representing the control system of a robot as an artificial chromosome subject to the laws of genetics and of natural selection dates back to the end of the 80's, but appeared in the scientific press only a couple of years later (Beer, 1990; Parisi et al., 1990; Cliff, 1991; Floreano, Miglino, & Parisi, 1991) when the first simulated robots with a sensorimotor system began evolving on computer screens. However, only during this decade, with the appearance of several new mobile robots designed according to the principles outlined in section 2, several researchers began evolving controllers for physical robot.

Brooks (1992) proposed the idea of evolving primitives of a behavior language for autonomous mobile robots, effectively combining a subsumption architecture (Brooks, 1986) with a genetic programming approach (Koza, 1992). The term *Evolutionary Robotics* has been coined by a group of researchers at the University of Sussex (Cliff, Harvey, & Husbands, 1993) whose approach is based on a combination of simulations and physical robots guided by evolutionary Dynamical-Recurrent-Neural-Networks (Harvey, Husbands, Cliff, Thompson, & Jakobi, 1997). The Sussex group has developed a new evolutionary paradigm called Species Adaptation Genetic Algorithm (Harvey, 1992, 1993) for incrementally evolving neurocontrollers and patterns of logical gates and connections for reconfigurable circuits (Thompson, Harvey, & Husbands, 1996). A research group at the Italian Research Council in Rome has introduced the concept of *Ecological Neural Networks* (Parisi et al., 1990) within the framework of evolutionary sensorimotor organisms and carried out a set of experiments that combine new simulation tools (Nolfi, Floreano, Miglino, & Mondada, 1994b) and physical robots. The Rome group has put particular emphasis on evolution of network architectures (Nolfi & Parisi, 1995; Nolfi, 1997), evolution and learning (Nolfi, Elman, & Parisi, 1994a; Nolfi & Parisi, 1996), and investigations of cognitive models of animal behavior with evolving robots (Miglino, Denaro, Tascini, & Parisi, 1998). Within a similar approach, Lund has explored issues on the co-evolution of controllers and robot morphologies (body plans) in simulations (Lund et al., 1997) and is currently looking at interactive evolution of robots for educational and entertainment applications (Lund, Miglino, Pagliarini, Billard, & Ijspert, 1998). Interactive selection of robot controllers by humans, also named subjective fitness, has been tested also by Gruau, whose work is characterized by evolution of modifiable neural architectures compactly encoded on the genotype as developmental rules (Gruau, 1996; Gruau & Quatramaran, 1997). Almost all the approaches employed in the research described above are based on evolution of neural controllers. A number of other persons have evolved other types of controllers, for example machine code (Nordin & Banzhaf, 1996), Boolean functions on reconfigurable circuits for vision-based navigation (Keymeulen, Iwata, Konaks, & Suzuki, 1998), and classifier systems (Dorigo & Colombetti, 1998). Several broad surveys of this field within the perspective of robotic applications are provided by Gomi (1996, 1997, 1998).

4 Environmental Shaping

Our approach is characterized by two main features: online evolution carried out entirely on physical mobile robots and a simple fitness function that emphasizes the role of environmental interactions in the development of new behaviors.

———————————

INSERT FIGURE 4 ABOUT HERE

———————————

The robot is attached to a workstation through a serial cable which provides power supply and communication of sensory values and motor commands every 300 ms. The evolutionary algorithm and robot controllers run on the workstation (figure 4). In the following sections we shall review several experiments that use the same or similar fitness function. It will be shown that different behaviors and/or controller dynamics emerge from the interaction between the types of robots, environments, and genetic encodings considered, without being explicitly programmed.

In this section we shall briefly review two experiments in (more details can be found in Floreano & Mondada, 1996a) that show how different environments can shape drastically different behaviors in neurocontrollers evolved under similar selection criteria. These data will be useful for evaluating and comparing more recent results described in the next two sections.

4.1 The looping maze

In one of our first experiments, we tested the hardware and software methodology described above for a simple reactive navigation task. The Khepera robot was placed in a looping maze environment (figure 5).

———————————

INSERT FIGURE 5 ABOUT HERE

———————————

A genetic algorithm (Goldberg, 1989) was used to evolve the synaptic strengths of a neural network composed of eight sensory units and two motor units. Each sensory unit was clamped to one of the eight active infrared sensors whose reading was updated every 300 ms. Each motor unit received weighted signals from the sensory units and from the other motor unit, plus a recurrent connection with itself with a 300 ms delay. The net input of the motor units was offset by a modifiable threshold and passed through a logistic squashing function. The resulting outputs, in the range $[0, 1]$, were used to control the two motors so that an output of 1 generated maximum rotation speed in one direction, an output of 0 generated maximum rotation speed in the opposite direction, and an output of 0.5 did not generate any motion in the corresponding wheel. A population of 80 individuals, each coding the synaptic strengths and threshold values of the neural controllers was initialized with all weights set to small random values centered around zero. Each individual was tested on the physical robot for 80 sensorimotor cycles (approximately 24 seconds) and evaluated at every cycle according to the following fitness function

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

where V is the average rotation speed of the two wheels, Δv is the absolute value of the algebraic difference between the signed speed values of the wheels (positive is one direction, negative the other) and i is the normalized activation value of the infrared sensor with the highest activity. The first component is maximized by speed, the second by straight motion, and the third by distance from objects.

———————————

INSERT FIGURE 6 ABOUT HERE

———————————

Each generation took approximately 40 minutes, including 5 seconds of random motion between tests of new individuals. The whole evolutionary process was fully automated, and statistics were collected on the hard disk of the workstation. During the first 100 generations both average and best fitness values grew steadily, as shown in figure 6.

———————————

INSERT FIGURE 7 ABOUT HERE

———————————

A fitness value of 1.0 could be achieved only by a robot moving straight at maximum speed in an open space. In the environment shown in figure 5, where some of the sensors were almost always active and where several turns were necessary to navigate, 0.3 was the maximum value attained (even when continued for further 100 generations; data not shown). Figure 7 shows the trajectory of the best individual of the last generation (data collected and plotted using the laser positioning device described in section 2).

————————

INSERT FIGURE 8 ABOUT HERE

————————

Although, the fitness function did not specify in what direction the robot should navigate (given that it is perfectly circular and that the wheels can rotate in both directions), after a few generations all the best individuals moved in the direction corresponding to the side with the higher number of sensors (figure 8). Individuals moving in the other direction had higher probability of colliding into corners without detecting them and thus disappeared from the population.

————————

INSERT FIGURE 9 ABOUT HERE

————————

Another way of describing the adaption process is the “state-space approach” proposed by ethologists for quantitatively measuring adaptation of biological autonomous agents (McFarland & Houston, 1981). The activity of an animal depends on its state, which is characterized by a set of variables such as its energy level, the perception of the environment, etc. These state variables define an n-dimensional state space, where the axes are provided by n state variables considered. Adaptation is then described as a transition from an initial oscillatory state towards a sub-region of this space. This region, which is compact and bounded, represents the equilibrium conditions of the animal (Sibly & McFarland, 1974). Within this framework, the robot can be described as a point in a three-dimensional space defined by three state variables corresponding to the three fitness components. Figure 9 shows the position of the best individuals at each generation within this graph. The best individuals of the last 20 generations remain within the same compact subregion of the space, despite the constant perturbations of the recombination and mutation operators, which represents the stability condition for the evolved controllers.

————————

INSERT FIGURE 10 ABOUT HERE

————————

Similarly, on the time scale of a single individual, when a controller (figure 10) is pulled away from the equilibrium region, for example by positioning it close to two walls, it will return to the stability region (in fact, implementing obstacle avoidance and forward navigation).

4.2 Battery recharge

In a second set of experiments, we modified the environment by positioning a battery charger on a corner of a square arena and a light tower over it (figure 11).

————————

INSERT FIGURE 11 ABOUT HERE

————————

The sensory system of the robot (and consequently the set of input units of the neural controller) was extended by enabling two ambient-light sensors (one on each side), fitting one infrared sensor beneath the platform of the robot (capable of telling apart black from white floor), and adding a sensor of battery charge. When the robot happened to pass over the recharging area, delimited by a black-painted area on the floor, its battery became instantaneously recharged. The battery was characterized by a fast linear discharge, allowing only 50 sensorimotor loops. The neural network under evolution was augmented by including five fully-connected hidden units. The same fitness function described above was used (except for the middle term which had been used to encourage straight navigation in the previous experiment) and accumulated at every time step.

INSERT FIGURE 12 ABOUT HERE

A population of 100 individuals was encoded and randomly initialized as in the experiment described in the previous section. Each individual started with a fully charged battery; individuals that happened to pass, or stay over the recharging zone could perform a higher number of sensorimotor loops during which their fitness points were accumulated. The maximum number of sensorimotor steps was set at 150. The robot evolved for 10 days in a room where the only illumination was provided by the small light tower over the recharging zone. All fitness indicators constantly increased (figure 12).

INSERT FIGURE 13 ABOUT HERE

The best individuals also managed to achieve an increasing number of sensorimotor steps (figure 13). After approximately 200 generations they were capable of navigating around the environment, covering long trajectories and avoiding walls and the recharging area; only when the battery was almost discharged they initiated a straight navigation towards the recharging zone and exited it once their battery was charged. These individuals always arrived at the recharging area with only 2 or 3 sensorimotor steps (approximately 1 second) of residual energy. Initiation of the trajectory for recharging depended on the location of the robot within the environment.

INSERT FIGURE 14 ABOUT HERE

A set of behavioral and “neuroethological” analyses were performed using the external positioning device described in section 2. By correlating the robot behavior and position with the activation of the neurons in real time while the robot freely moved in the environment, it was possible to show that some units specialized for reactive behaviors, such as obstacle avoidance, forward motion, battery monitoring, light following, while others displayed more complex activation patterns. One of them revealed a pattern of activation levels that depended on whether the robot was oriented facing the light tower or facing the opposite direction (figure 14). In the former case, the activation reflected zones of the environment and paths typically followed by the robot during navigation. For example, one can see a path frontally leading to the recharging area which is marked by specific neural activity. In the latter case, the same neuron displays a gradient field orthogonally aligned with the recharging area. This gradient provides an indication of the distance from the recgarging area. This pattern of activity is not significantly affected by the charge level of the battery, which is integrated elsewhere in the neural network. More details and other behavioral and neural measures can be found in (Floreano & Mondada, 1996a).

4.3 Discussion

The differences between the behavioral outcomes of these two experiments are mainly due to the characteristics of the environment in which the robot has evolved, rather than to the neurocontroller architectures or to the fitness functions. An environment is not a set of objective features that exist independently of the robot, but *the* set of features that can be perceived by the robot and that play some role in its mission, which here is to accumulate fitness points for selective reproduction. From this ecological perspective, the characteristics of the robot shell and of its sensorimotor system are part of the environment. The main difference between artificial evolution, as it has been used here, and other approaches to adaptive behavior is that the former capitalizes on the interactions between the robot and the environment seen from the robot’s perspective, whereas the latter methods often impose architectural and adaptive constraints that derive from the experimenter’s view of the environment (for further discussion of this issue, see also Nolfi, 1997). The emergent direction of motion in the experiment with the looping maze is a very simple example of how evolution develops solutions that match the environmental characteristics from the perspective of the agent.

Self-selection of information and of suitable solutions is relevant also for the value system of an autonomous robot. In simple words, a value system provides the agent with a way to judge what is good and what is not. A value system can be used for regulating behavior and modulating learning (McFarland & Boesser, 1993; Pfeifer & Scheier, 1998). Typically, in reinforcement learning approaches (e.g., Kaelbling, Littman, & Moore, 1996) and other adaptive models, such as map learning (e.g., Burgess, Recce, & O'Keefe, 1994), the value system is externally imposed by the experimenter. For example, certain sensory configurations or locations of the environment are associated with positive rewards or can trigger synaptic changes. This strategy might make sense from an engineering point in simple environments, but it might fail in more complex settings where multiple values should be developed, integrated, and appropriately weighted depending on the circumstances. Artificial evolution could potentially be a way for developing autonomous agents that self-select their value system. For example, in the experiment with the battery charger the neurocontroller autonomously discovered the battery charger and learned to periodically return to it because this behavior allowed it to be fitter in the evolutionary task of keeping as much as possible away from obstacles. Battery charging, and all the neural machinery that was evolved in order achieve this self-selected "goal", was not directly specified in the fitness function, but emerged as a sub-goal functional to the maximization of a more general fitness criterion defined from the perspective of the robot: keep the wheels moving and the sensor activations low.

5 Re-adaptation

The results from the two experiments described above indicate that artificial evolution can develop efficient neurocontrollers exploiting relevant features of the environment that were not explicitly defined in advance. From an engineering perspective, the price to pay is the amount of time required by the evolutionary process carried out on physical robots. The question, then, is to what extent the system can generalize and/or re-adapt to modified environmental conditions without re-training it from scratch. Artificial evolution of neurocontrollers offers generalization at two levels: the individual and the population. At the individual level, generalization capitalizes on the feature invariants encoded by the neural network. At the population level, generalization is provided by diversity of the individuals. Depending on the selection pressure, a population will sooner or later converge to a single solution, but the mutation operator will still maintain a number of different solutions distributed around the best individual.

5.1 Environmental re-arrangement

Consider the experiment with the battery charger described in the subsection 4.2. One way of testing the generalization properties of the evolved neurocontrollers consists in manipulating the arrangement of the environment.

———————————

INSERT FIGURE 15 ABOUT HERE

———————————

The first frame of figure 15 shows a typical trajectory of the best neurocontroller after 240 generations. If the battery is not automatically re-charged when the robot arrives to the charging area, the robot will circulate over the black area until all the residual energy is exhausted (figure 15, center). If the light source is moved to the top right corner (but the charging area is not moved), the robot will head toward the light and move in its surroundings until all the energy is exhausted (figure 15, right). Therefore, the position of the light tower represents an important perceptual cue.[2]

———————————

INSERT FIGURE 16 ABOUT HERE

———————————

In order to test the generalization properties of the population and the re-adaptation dynamics, the population of neurocontrollers of generation 240 was tested and incrementally evolved in three new environmental conditions, each one identified by positioning the light tower in a different corner of the environment (top-right, bottom-right, and bottom-left). For each condition, the genetic algorithm was restarted on the population of generation 240 and continued for 80 additional generations. Data for all conditions are displayed in Figure 16, including data from the previous ten generations for the sake of comparison. The presence of several individuals which were "sub-optimal" in the original environment, but resulted fitter in the new environment prevented a dramatic drop of the both fitness indicators. Re-adaptation of the population to each new condition took place relatively quickly; approximately 20 generations (10% of the time required for evolution from scratch) were sufficient to generate an individual perfectly adapted to the new environment which reported the same performances already measured for the best individual of

[2] Indeed, in experiments reported elsewhere (Floreano & Mondada, 1996a) on the best evolved individual, it was observed that if the light was switched off during the return to the charger, the robot performed wide circular trajectories looking for gradient information.

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 16, *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 90-degree rotation of the mapping from sensors to actions, involving changes in several synaptic weights, condition *b* can be successfully solved by changing few synaptic weights which result in a 180-degree rotation of the sensorimotor mapping. If this is the case, then one would expect that only few best individuals (i.e., those which already developed a correctly oriented map in the original environment) could 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*.

5.2 Crossplatform evolution

Consider now the simple experiment described in subsection 4.1 where the robot learns to navigate in a looping maze (figure 5). The evolved neurocontroller has developed a pattern of synaptic weights that matches the morphology of the Khepera, its sensorimotor layout, and the response properties of the infrared sensors. However, this platform-specific solution does not allow a successful transfer of the evolved controllers onto a robot with a different morphology. Rather than starting evolution from scratch on a new robot, one might wish to continue evolution on the new robot incrementally. From the point of view of the neurocontroller, changing the sensorimotor characteristics of the robot is just another way of modifying the environment.

————————

INSERT FIGURE 17 ABOUT HERE

————————

The Koala robot described in section 2 was placed within a scaled-up version of the looping maze already employed for the Khepera robot (figure 17). Only eight of the sixteen infrared sensors available on the Koala were selected as input to the neurocontroller (see figure 3; from left clockwise: L4, L3, L1, R1, R3, R4, R7, L7). As for the Khepera, the response of each infrared sensor was linearly compressed within the range $[0, 1]$.

————————

INSERT FIGURE 18 ABOUT HERE

————————

After 106 generations of evolution on the Khepera robot, the last population of neurocontrollers was incrementally evolved on the Koala robot until generation 150 (figure 18) without any further change to the software parameters. After a partial drop in performance, in approximately thirty generations the best individuals report fitness values similar to those recorded during the last generations on the Khepera and are capable of performing full laps of the maze. In this experiment, re-adaptation was mainly concerned with a new body shape and a different layout of the sensors. For example, the passage on the top right corner of the environment requires a rotation in place for the Koala whereas the Khepera—in its original maze—could go around it without stopping.

5.3 Discussion

The incremental evolutionary approach described in this section works as long as there is enough diversity in the population of individuals. The amount of diversity required depends on many factors, such as the genetic encoding used, the type of changes made to the environment, and other evolutionary parameters, such as the selection pressure, the mutation rate, the population size, etc. However, there is trade-off between the amount of diversity that must be maintained and the ability of the system to discover stable solutions. We think that in the long run this approach is not likely to be a viable solution to open-ended evolution. Open-ended evolution will require several variations that go beyond a traditional genetic algorithm, such as a genetic encoding that can capitalize on previously evolved behaviors, genetic operators that can keep diversity while limiting damage to highly fit genes, and an evolutionary program that is more inspired upon natural evolution rather than upon algorithms for function optimization.

However, this simple type of incremental evolution is a viable solution to the bootstrapping problem, that is the attempt to evolve from scratch a very complex behavior which might correspond to a very tiny and peaked region of the fitness landscape. In this case, it is very likely that all the individuals of the initial generation will report zero fitness and no progress might take place. A solution consists of starting to evolve the population on simple versions of the desired task, and incrementally evolve the same population with increasingly more complex environments and fitness functions (e.g., see Harvey, Husbands, & Cliff, 1994).

6 Evolution of Plastic Controllers

In nature, adaptation to the environment takes place at multiple levels and time scales, ranging from the long-term dynamics of *phylogenetic evolution* to the fast process of *ontogenetic learning.* Most of the work in Evolutionary Robotics, including the experiments described above, are inspired upon the mechanisms of phylogenetic evolution. From a theoretical perspective, Hinton and Nowlan (1987) showed that learning might ameliorate and speed up artificial evolution by exploring the space surrounding an individual on the fitness landscape. The combination of artificial evolution and learning has recently attracted much interest (e.g., see Belew & Mitchell, 1996; Turney, Whitley, & Anderson, 1996), but most computational approaches are limited to a combination of genetic algorithms and traditional supervised learning algorithms (e.g., see Parisi et al., 1990; Ackley & Littman, 1992).

In this section we describe a different approach based on the assumption that the neural mechanisms underlying ontogenetic learning are themselves developed and shaped by the evolutionary process. Once connected to the robot, the evolved neurocontrollers continuously modify their synaptic strengths according to the plasticity rules specified in the genotype. Each genotype encodes a set of parameters that describes static and dynamic characteristics of the synapses, but not synaptic strengths. As soon as a neurocontroller is decoded from the genotype, its synapses are initialized to small random values which will begin to change depending on the interactions between the robot and the environment; once the maximum life-time for an individual neurocontroller has expired, the final synaptic strengths are *not* written back into the chromosome. Therefore, rather than using artificial evolution as a surrogate of learning, this approach is concerned with exploring the evolution of learning. For the sake of analysis and comparison with the experiments already described above, the same looping maze and fitness function described in section 4 were used also here and all the experiments were carried out on the Khepera robot.

————————

INSERT FIGURE 19 ABOUT HERE

————————

The architecture of the neurocontroller consisted of three neurons—one hidden neuron and two motor neurons—each receiving synaptic connections from the eight infrared sensors and from the hidden neuron itself (27 synapses); as in the previous experiments, this architecture could not be modified by the evolutionary algorithm. Synaptic connections could have a *driving* or a *modulatory* effect on the postsynaptic neuron; afferent signals were combined in a two-component activation function (Phillips, Kay, & Smyth, 1995) which gave an output between 0 and 1 (figure 19). Driving signals determined whether the unit activity was below or above 0.5 (which, when transformed into the range ±0.5 for motor control, was the point of inversion of wheel rotation), whereas modulatory signals could enhance or dampen the unit response, but could not change the direction of wheel rotation. Each synapse was individually coded on binary chromosomes by a set of four properties: whether driving or modulatory (1 bit), whether excitatory or inhibitory (1 bit), its learning rule (2 bits), and its learning rate (four values encoded on 2 bits: 0.0, 0.33, 0.66, and 1.0). Four Hebbian learning rules were taken into consideration: simple Hebb, postsynaptic, presynaptic, and covariance (see Willshaw & Dayan, 1990).

These rules were modified in order to satisfy the following constraints. Synaptic strength could not grow indefinitely, but was kept in the range $[0, 1]$ by means of a self-limiting mechanism which depended on synaptic strength. Because of this self-limiting factor, a synapse could not change sign, which was genetically specified, but only strength. Synaptic strength change Δw depended solely on presynaptic and postsynaptic activations x, y.

The simplest learning mechanism—*plain Hebb*— allows only synaptic strengthening

$$\Delta w = (1 - w)\, xy \tag{2}$$

The *postsynaptic rule* keeps the strengthening conditions of the Hebb rule, but also adds the possibility of synaptic weakening when the postsynaptic unit is active and the presynaptic unit is not

$$\Delta w = w\,(-1 + x)\, y + (1 - w)\, xy \tag{3}$$

Conversely, in the *presynaptic rule* synaptic weakening takes place when the presynaptic unit is active, but the postsynaptic unit is inactive

$$\Delta w = wx\,(-1 + y) + (1 - w)\, xy \tag{4}$$

It should be noticed that the postsynaptic and presynaptic rules, once simplified, are respectively very similar to the "instar" and "oustar" learning rules described by Grossberg (1980, 1982). Finally, the *covariance rule* strengthens or weakens a synapse depending on the difference between the activation levels of the presynaptic and postsynaptic units: if the presynaptic and postsynaptic activation levels differ by more than half the maximum node activation ($|x-y| > 0.5$), synaptic strength is reduced in proportion to that difference, otherwise it is increased in proportion to the difference

$$\Delta w = \begin{cases} (1-w)\mathcal{F}(x,y) & \text{if } \mathcal{F}(x,y) > 0 \\ (w)\mathcal{F}(x,y) & \text{otherwise} \end{cases} \tag{5}$$

where $\mathcal{F}(x,y) = \tanh(4(1-|x-y|)-2)$ is a measure of the difference between the presynaptic and postsynaptic activity. $\mathcal{F}(x,y) > 0$ if the difference is bigger or equal to 0.5 (half the maximum node activation) and $\mathcal{F}(x,y) < 0$ if the difference is smaller than 0.5.

As soon as a neurocontroller is decoded and attached to the sensors and motors of the robot, each synapse is assigned its sign (excitatory or inhibitory), role (driving or modulatory), learning rate, and its strength is initialized to a new random value in the range $[0.0, 0.1]$; all synaptic strengths are updated every 300 ms (the duration of one sensorimotor loop) according to the following discrete-time equation

$$w^t = w^{t-1} + \eta\Delta w^t \tag{6}$$

where η is the learning rate.

Three different runs of this experiment were made. In all cases the best individual fitness reached a maximum value around the 50th generation ($\Phi = 0.23, \pm 0.09$). As compared to the evolution of synaptic strengths described in section 4.1, the neurocontroller reached the same performance level much faster (almost twice as fast), but displayed more inter- and intra-generation fitness variability. The evolved behaviors displayed smooth paths around the arena (figure 7). It should be noticed that this behavior was developed by each individual neurocontroller during the first few sensorimotor loops, independently of the initial random values assigned to the synapses. As for the experiment described in section 4, all the best individuals across the three runs advanced in the direction corresponding to the higher density of sensors.

INSERT FIGURE 7 ABOUT HERE

The evolved neurocontrollers displayed different behavioral strategies and learning modalities both within a single population and across the three evolutionary runs. From an observer perspective, these behaviors were variations of two main strategies that could be labeled as "reactive obstacle avoidance" and "active wall following". Reactive-obstacle-avoidance looked very similar to the behavioral strategy evolved in the experiment of section 4.1, except for initial bumps into walls that disappeared after a few sensorimotor loops. Here, we shall analyze an active-wall-following behavior displayed by one of the best individuals.

The robot was positioned facing a corner of the inner wall, the synapses were initialized to small random values, and the robot was let free to move (see figure 7, left; initial position corresponds to the set of superimposed bars in the lower portion of the environment). During the initial 6-7 sensorimotor loops, the robot alternated backward and forward motions until it found a wall on its right side. It then began moving forward (counterclockwise) keeping the wall at a distance of 2 cm on the right side. Every second or third action, it slightly turned toward the wall and then resumed forward navigation. This turning behavior was gradually reduced when the robot moved along a straight long wall (e.g., along the north and east walls). When the robot encountered a wall frontally, it stopped, backed shortly while rotating to the right, and then resumed forward motion in the new direction. After the first lap of the maze, the path became smoother with less trajectory adjustements (Figure 7, right).

INSERT FIGURE 21 ABOUT HERE

Figure 21 plots the strengths of all the *active synapses* in the network during the first 100 sensorimotor loops visualized in figure 7. All the synapses to the internal node remain close to zero, except for those from the rear sensors and the self-recurrent connection. As a consequence, the internal unit is maintained always moderately active, without regard to the external sensory information, providing constant signal to the motor wheels also in the absence of sensory information. Since the synapses conveying these signals are both driving and excitatory, the internal unit generates a constant forward rotation of both wheels. Similarly, all the synapses from the sensors on the left side of the body to the two wheels remain close to zero (except for one—not plotted here—which uses the postsynaptic rule and can rapidly strengthen if the robot finds something on the left side). All the remaining synapses to the right-wheel neuron are excitatory: since most of them are modulatory, they can accelerate the rotation speed when the sensors on the right side become more active. The excitatory driving synapse from the central sensor to the right wheel causes fast accelerations and decelerations which, combined with the inhibitory effect of the corresponding synapse to the left-wheel neuron, can generate backward rotations when something is frontally detected. Synapses connecting the three sensors on the front-right side to the left wheel are responsible for the wall-following behavior. The synapse corresponding to the rightmost sensor (nearly always highly active) quickly learns to transmit constant and high excitation to the left wheel. This excitation—that would lead the robot against a wall—is counterbalanced by the driving inhibitory signals that come from the other two front-right sensors: the corresponding synapses display an oscillatory pattern that is responsible for the frequent small turns toward the wall. These turns are functional because they provide information about the wall curvature by allowing all the three right-front sensors to receive information on distance from the wall. the gradual stabilization of behavior displayed by the robot trajectory is reflected by the weakening of the oscillations displayed by the inhibitory synapse from the center-right sensor to the left motor.

————————

INSERT FIGURE 22 ABOUT HERE

————————

A key observation from this experiment is that behavioral stability is not expressed by a final stable state of the synaptic configuration, as in most connectionist models (Hertz, Krogh, & Palmer, 1991), but rather by dynamic equilibrium in an n-dimensional state-space (where n is the number of synapses). Figure 22 plots the trajectory of the neurocontroller in the reduced state-space of the first three principal components of the recorded synaptic activity during the first 100 actions of the individual (displayed in figure 7). During the first 6 actions the neurocontroller moves toward a subregion of space where there is little change along the first and second principal components. During the remaining actions, almost all the residual variation takes place in the dimension of the third principal component and is gradually reduced as the robot performs the second lap around the maze.

The evolved individuals could not store genetic information on the fine details of the environment in the form of precise synaptic strengths, as in the experiment described in section 4, but the pattern of active synapses and their sign was sufficient to store some basic behaviors. For example, in the case of the neurocontroller analyzed above, at the beginning the robot always moved backward until it found a wall and then it positioned itself so that the wall happened to be on its right side. This sort of reflex, that is genetically inherited and immutable behavior, effectively puts the the neurocontroller in a position from where it can effectively engage in a dynamic adaptation process (other analyses can be found in Floreano & Mondada, 1996b).

6.1 Discussion

The genetic encoding used in this experiment was motivated by the desire to keep the number of independent variables as small as possible while capturing some of the biological elements relevant in the evolution of adaptive nervous systems. Therefore, the genotype encoded only gross properties of the synapses, such as their postsynaptic effects (driving or modulatory) (Hirsch & Gilbert, 1993) and their signs, but not synaptic strengths. The choice of encoding four variations of the Hebb rule, and the learning rates, was motivated by neurophysiological evidence for these plasticity rules in biological nervous systems (Kelso, Ganong, & Brown, 1986; Stent, 1973; Singer, 1987; Stanton & Sejnowski, 1989; Yang & Faber, 1991), and inspired upon biological findings of genes regulating the expression of NMDA receptors (Hollman & Heinemann, 1993) which are thought to be the most likely mechanism responsible for activating different types of Hebbian learning. Artificial evolution selected some of the available features and discarded others. For example, the internal unit of the neurocontroller analyzed above was not used for processing sensory information, and the plain Hebb rule was used very sparsely (as in most individuals of the last generations). Since the Hebb rule does not allow weakening of a synapse, it was used only for synapses responsible for establishing basic competencies, such as motion generation from the internal unit.

The stable behavior achieved during "lifetime" by the controllers was regulated by fast and continously changing synapses characterized by *dynamic stability.* In the conventional view, synapses are relatively (statically) stable and slow components of the nervous system. In most connectionist models, synaptic change takes place on a slower time scale than neuron activations. Consequently, synaptic changes are identified with learning of new skills or acquisition of new knowledge, while neuron activations are identified with the expression of behavior and of existing knowledge.[3] Within this perspective, the neurocontrollers evolved in this experiments offer a complementary view of adaptive behavior where fast-changing synapses are responsible for both learning and behavior regulation. From a computational perspective, this functioning modality adds an extra degree of freedom to the neurocontroller. Preliminary investigations in dynamic environments where two robots, a predator and a prey, are co-evolved in the same arena have shown that predators equipped with evolved plastic controllers can adapt their behavior on-the-fly and catch a higher number of prey displaying different escape strategies (Floreano & Nolfi, 1997) than predators equipped with evolved controllers that have the same architecture, but fixed synaptic weights.

Life-time adaptation of the evolved neurocontroller analyzed above is supported by an evolved reflex—backing against a wall—that increases the probability of the controller to be in a situation from where it can learn a behavior that is useful for selective reproduction, in this case straight navigation without collision. Evolved reflexes can be seen as an implicit value system because they increase the probability of getting explicit values (Pfeifer & Scheier, 1998), that is fitness points.

7 Conclusion

We have described a methodology for building, evolving, and analyzing autonomous mobile robots. The characteristics of the robots employed in these experiments, combined with a set of tools for monitoring the evolutionary progress, have allowed us to run all the experiments on physical robots for extended periods of time and perform "neuroethological" analyses in order to link behavioral performance with neural dynamics. In all the experiments described in this article we have employed very similar fitness functions and neural architectures in the attempt to study how the evolved controllers are affected by the environment and by the encoding strategy employed.

The design of the fitness function is one of the most delicate steps involved in setting up an evolutionary system. Our approach described here consists in choosing simple functions, designed from the perspective of the robot, that use only sensory information locally available to the robot and capitalize on the interaction dynamics between the robot and its environment. In this way, evolution can gradually build and combine solutions that perfectly match the physical characteristics of the robot and of the environment without externally imposed constraints based on an observer's perspective. The evolution of a preferential direction of motion (section 4.1) and of a battery-charging behavior (section 4.2) are two examples of solutions that have not been explicitly imposed by the experimenter.

From an engineering point of view, one might be tempted of accelerating evolution by, for example, devising fitness functions that explicitly select individuals with desired macro-behaviors or by evolving higher-level primitives. Although this strategy can indeed generate complex controllers very quickly (Urzelai, Floreano, Dorigo, & Colombetti, 1998), the self-organizing potentialities of the evolving controllers are limited by the constraints imposed by the experimenter. Conversely, if one wishes to use artificial evolution for investigating cognitive and biological aspects of autonomous behavior, it would be more fruitful to concentrate efforts on devising increasingly more challenging environments and situations—rather than fitness functions—that could implicitly push the evolutionary individuals towards more complex abilities.

Artificial evolution can also be a tool for exploring alternative forms of adaptation and gaining insights on mechanisms that might (or might not) be used by biological forms of life, but have not yet been discovered. The experiment on the evolution of plastic controllers described in the last section goes in this direction. The evolved controllers make use of fast and continuously changing synapses that are dynamically stable, suggesting new ways in which a network of interconnected adaptive elements might achieve adaptive behavior. From an Artificial Life perspective, evolutionary robots are embodied and embedded systems suitable for exploring life-as-it-is and life-as-it-could-be (Langton, 1990).

Independently of the point of view adopted, we believe that in the years to come it will be necessary to invest much research efforts in *incremental open-ended* evolution in order to evolve artificial agents displaying the complexity and adaptivity of biological autonomous systems. From this broader perspective, the experiments described here are initial explorations in the artificial evolution of autonomous robots aimed at investigating the methodology and evaluating the potentials. Despite some initial theoretical work in this direction (Harvey, 1992, 1993), pursuing an incremental open-ended evolutionary program will require starting several new research projects looking from a different perspective at issues such as genetic encoding, evolvable primitives, developmental programs, selection

[3] This view has been partially challenged by Yamauchi and Beer (1995) , who have evolved and analyzed continuous-time recurrent neural networks that give the external appearance of performing reinforcement learning while, in fact, these networks have fixed connection weights and use only internal node dynamics.

criteria, evolutionary operators, and principles of fitness design for autonomous artificial organisms that interact with partially unknown and unpredictable physical environments.

Reference

Ackley, D. H., & Littman, M. L. (1992). Interactions between learning and evolution. In Langton, C., Farmer, J., Rasmussen, S., & Taylor, C. (Eds.), *Artificial Life II: Proceedings Volume of Santa Fe Conference*, Vol. XI. Addison Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, Redwood City, CA.

Beer, R. D. (1990). *Intelligence as adaptive behavior: an experiment in computational neuroethology.* Academic Press, San Diego, CA.

Belew, R. K., & Mitchell, M. (Eds.). (1996). *Adaptive Individuals in Evolving Populations: Models and Algorithms.* Addison-Wesley, Redwood City, CA.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Robotics and Automation*, *RA-2*, 14–23.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, *47*, 139–59.

Brooks, R. A. (1992). Artificial Life and real robots. In Varela, F. J., & Bourgine, P. (Eds.), *Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life.* The MIT Press/Bradford Books, Cambridge, MA.

Brooks, R. A., Breazeal (Ferrell), C., & Irie, R. (1998). Alternative Essences of Intelligence. In Gomi, T. (Ed.), *Evolutionary Robotics. From Intelligent Robots to Artificial Life.* AAI Books, Kanata, Canada.

Burgess, N., Recce, M., & O'Keefe, J. (1994). A model of hippocampal function. *Neural Networks*, *7*, 1065–1081.

Cheneval, Y. (1995). Packlib, an interactive environment to develop modular software for data processing. In Mira, J., & Sandoval, F. (Eds.), *From Natural to Artificial Neural Computation, IWANN-95*, pp. 673–682 Malaga. Springer Verlag.

Cliff, D., Harvey, I., & Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, *2*, 73–110.

Cliff, D. T. (1991). Computational neuroethology: a provisional manifesto. In Meyer, J. A., & Wilson, S. W. (Eds.), *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA.

Dorigo, M., & Colombetti, M. (1998). *Robot shaping: An experiment in behavior engineering.* MIT Press, Cambridge, MA.

Dorigo, M., & Schnepf, U. (1993). Genetic-based machine learning and behavior based robotics: a new synthesis. *IEEE Transactions on Systems, Man and Cybernetics*, *23*, 141–154.

Floreano, D., Miglino, O., & Parisi, D. (1991). Emergent complex behaviours in ecosystems of neural networks. In Caianiello, E. (Ed.), *Parallel Architectures and Neural Networks.* World Scientific Press, Singapore.

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

Floreano, D., & Mondada, F. (1996b). Evolution of plastic neurocontrollers for situated agents. In Maes, P., Matarić, M., Meyer, J., Pollack, J., Roitblat, H., & Wilson, S. (Eds.), *From Animals to Animats IV: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pp. 402–410. MIT Press-Bradford Books, Cambridge, MA.

Floreano, D., & Nolfi, S. (1997). Adaptive behavior in competing co-evolving species. In Husbands, P., & Harvey, I. (Eds.), *Proceedings of the 4th European Conference on Artificial Life* Cambridge, MA. MIT Press.

Franceschini, N., Pichon, J.-M., & Blanes, C. (1991). Real time visuomotor control: From flies to robots. In *Proceedings of the Fifth International Conference on Advanced Robotics*, pp. 91–95 Pisa.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Redwood City, CA.

Gomi, T. (1996). Non-cartesian robotics. *Robotics and Autonomous Systems*, *18*, 169–184.

Gomi, T. (1997). Robotics and Emerging Business Principles. In Gomi, T. (Ed.), *Evolutionary Robotics. From Intelligent Robots to Artificial Life* Kanata, Canada. AAI Books.

Gomi, T. (1998). Non-Cartesian Robotics - the first 10 years. In Gomi, T. (Ed.), *Evolutionary Robotics. From Intelligent Robots to Artificial Life*, pp. 245–306 Kanata, Canada. AAI Books.

Grossberg, S. (1980). How does the brain build a cognitive code?. *Psychological Review*, *87*, 1–51.

Grossberg, S. (1982). *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control.* Reidel Press, Boston, MA.

Gruau, F. (1996). Artificial cellular development in optimization and compilation. In Sanchez, E., & Tomassini, M. (Eds.), *Towards Evolvable Hardware. The Evolutionary Engineering Approach.* Springer-Verlag, Berlin, Heidelberg.

Gruau, F., & Quatramaran, K. (1997). Cellular encoding for interactive evolutionary robotics. In Husbands, P., & Harvey, I. (Eds.), *Proceedings of the 4th European Conference on Artificial Life* Cambridge, MA. MIT Press.

Harvey, I. (1992). Species Adaptation Genetic Algorithms: A basis for a continuing SAGA. In Varela, F. J., & Bourgine, P. (Eds.), *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pp. 346–354. MIT Press-Bradford Books, Cambridge, MA.

Harvey, I. (1993). Evolutionary robotics and SAGA: the case for hill crawling and tournament selection. In Langton, C. (Ed.), *Artificial Life III*, pp. 299–326. Addison Wesley, Redwood City, CA.

Harvey, I., Husbands, P., & Cliff, D. (1994). Seeing The Light: Artificial Evolution, Real Vision. In Cliff, D., Husbands, P., Meyer, J., & Wilson, S. W. (Eds.), *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA.

Harvey, I., Husbands, P., Cliff, D., Thompson, A., & Jakobi, N. (1997). Evolutionary Robotics: The Sussex Approach. *Robotics and Autonomous Systems*, *20*, 205–224.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation.* Addison-Wesley, Redwood City, CA.

Hinton, G. E., & Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, *1*, 495–502.

Hirsch, J. A., & Gilbert, C. D. (1993). Long-term changes in synaptic strength along specific intrinsic pathways in the cat visual cortex. *Journal of Physiology*, *461*, 247–262.

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

Hollman, M., & Heinemann, S. (1993). Cloned glutamate receptors. *Annual Review of Neuroscience*, *17*, 31–108.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kay, J., & Phillips, W. A. (1996). Activation Functions, Computational Goals and Learning Rules for Local Processors with Contextual Guidance. *Neural Computation*, *9*, 895–910.

Kelso, S., Ganong, A., & Brown, T. (1986). Hebbian synapses in hippocampus. *Proceedings of the National Academy of Sciences USA*, *83*, 5326–5330.

Keymeulen, D., Iwata, M., Konaks, K., & Suzuki, R. (1998). Off-Line and On-Line Model-Based Evolution for Tracking Navigation Using Evolvable Hardware. In Husbands, P., & Meyer, J. (Eds.), *Proceedings of the First European Workshop on Evolutionary Robotics* Berlin. Springer Verlag.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection.* MIT Press, Cambridge, MA.

Langton, C. G. (1990). Artificial life. In Langton, C. (Ed.), *Artificial Life*. Addison-Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, Redwood City, CA.

Lund, H. H., Hallam, J., & Lee, W.-P. (1997). Evolving robot morphology. In *Proceedings of the IEEE 4th International Conference on Evolutionary Computation*. IEEE Press.

Lund, H. H., Miglino, O., Pagliarini, L., Billard, A., & Ijspert, A. (1998). Evolutionary Robotics—A Children's Game. In *Proceedings of the IEEE 5th International Conference on Evolutionary Computation*. IEEE Press.

McFarland, D. J., & Boesser, T. (1993). *Intelligent Behavior in Animals and Robots*. MIT Press/Bradford Books, Cambridge, MA.

McFarland, D. J., & Houston, A. (1981). *Quantitative Ethology: the state-space approach*. Pitman Books, London.

Miglino, O., Denaro, D., Tascini, G., & Parisi, D. (1998). Detour behaviours in evolving robots: Are internal representations necessary?. In Husbands, P., & Meyer, J. (Eds.), *Proceedings of the First European Workshop on Evolutionary Robotics* Berlin. Springer Verlag.

Mondada, F., Franzi, E., & Ienne, P. (1993). Mobile robot miniaturization: A tool for investigation in control algorithms. In Yoshikawa, T., & Miyazaki, F. (Eds.), *Proceedings of the Third International Symposium on Experimental Robotics*, pp. 501–513 Tokyo. Springer Verlag.

Nolfi, S. (1997). Using emergent modularity to develop control system for mobile robots. *Adaptive Behavior*, *5*, 343–364.

Nolfi, S., Elman, J. L., & Parisi, D. (1994a). Learning and evolution in neural networks. *Adaptive Behavior*, *3*, 5–28.

Nolfi, S., Floreano, D., Miglino, O., & Mondada, F. (1994b). How to evolve autonomous robots: Different approaches in evolutionary robotics. In Brooks, R., & Maes, P. (Eds.), *Proceedings of the Fourth Workshop on Artificial Life*, pp. 190–197 Boston, MA. MIT Press.

Nolfi, S., & Parisi, D. (1995). Genotypes for neural networks. In Arbib, M. A. (Ed.), *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA.

Nolfi, S., & Parisi, D. (1996). Learning to adapt to changing environments in evolving neural networks. *Adaptive Behavior*, *5*, 75–98.

Nordin, P., & Banzhaf, W. (1996). An online method to evolve behavior and to control a miniature robot in real time with genetic programming. *Adaptive Behavior*, *5*, 107–140.

Parisi, D., Cecconi, F., & Nolfi, S. (1990). Econets: Neural networks that learn in an environment. *Network*, *1*, 149–168.

Pfeifer, R. (1995). Cognition – perspectives from autonomous agents. *Robotics and Autonomous Agents*, *15*, 47–70.

Pfeifer, R., & Scheier, C. (1998). *Understanding Intelligence*. MIT Press, Cambridge, MA.

Phillips, W. A., Kay, J., & Smyth, D. (1995). The discovery of structure by multi-stream networks of local processors with contextual guidance. *Network*, *6*, 225–246.

Schwefel, H. P. (Ed.). (1995). *Evolution and Optimum Seeking*. John Wiley, Chichester.

Sibly, R. M., & McFarland, D. J. (1974). A state-space approach to motivation. In McFarland, D. J. (Ed.), *Motivational Control Systems Analysis*. Academic Press, London.

Singer, W. (1987). Activity-dependant self-organisation of synaptic connections as a substrate of learning.. In Changeux, J. P., & Konishi, M. (Eds.), *The Neural and Molecular Bases of Learning*. Wiley, London.

Stanton, P. K., & Sejnowski, T. J. (1989). Associative long-term depression in the hippocampus induced by hebbian covariance. *Nature*, *339*, 215–218.

Steels, L. (1994). The Artificial Life Roots of Artificial Intelligence. *Artificial Life*, *1*, 75–110.

Steels, L. (Ed.). (1995). *The Biology and Technology of Intelligent Autonomous Agents.* Springer-Verlag – NATO ASI Series, Berlin.

Stent, G. (1973). A physiological mechanism for hebb's postulate of learning. *Proceedings of the National Academy of Sciences USA*, *70*, 997–1001.

Thompson, A., Harvey, I., & Husbands, P. (1996). Unconstrained evolution and hard consequences. In Sanchez, E., & Tomassini, M. (Eds.), *Towards Evolvable Hardware. The Evolutionary Engineering Approach.* Springer Verlag, Berlin.

Turney, P., Whitley, D., & Anderson, R. (1996). Special Issue on Evolution, Learning, and Instincts: 100 Years of the Baldwin Effect. *Evolutionary Computation*, *4*, iii–viii.

Urzelai, J., Floreano, D., Dorigo, M., & Colombetti, M. (1998). Incremental robot shaping. In Koza, J., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M., Goldberg, D. E., Iba, H., & Riolo, R. (Eds.), *Genetic Programming 1998: Proceedings of the Third Annual Conference* San Francisco, CA. Morgan Kaufmann.

Willshaw, D., & Dayan, P. (1990). Optimal plasticity from matrix memories: What goes up must come down. *Neural Computation*, *2*, 85–93.

Yamauchi, B., & Beer, R. D. (1995). Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior*, *2*, 219–246.

Yang, X.-D., & Faber, D. S. (1991). Initial synaptic efficacy influences induction and expression of long-term changes in trasmission. *Proceedings of the National Academy of Science USA*, *88*, 4299–4303.

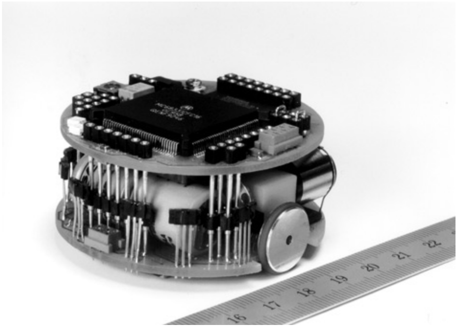

Figure 1: The miniature mobile robot Khepera beside a ruler (in cm). Black pucks around the body are active infrared sensors. Rechargeable batteries are sandwiched between the sensorimotor board and the motherboard; the latter hosts the microcontroller (black square), EEPROM, RAM, and communication ports.

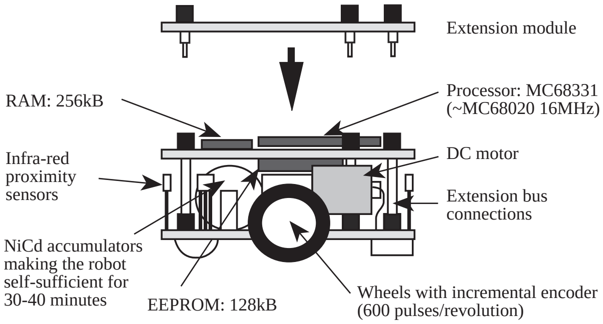

Figure 2: The open modular architecture of the Khepera robot.

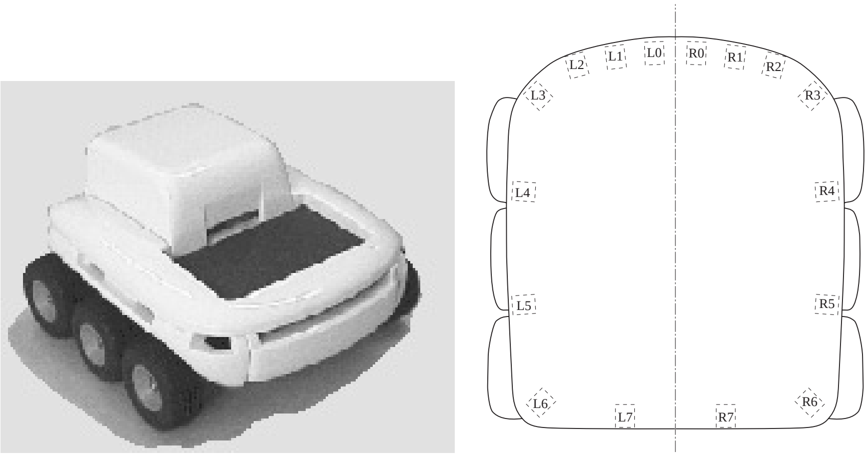

Figure 3: **Left:** The Koala robot, despite its shape and size (31 x 32 x 18 cm), has operating features similar to those of the Khepera robot with which it is fully compatible. **Right:** Layout of infrared sensor positions. L=left; R=right.

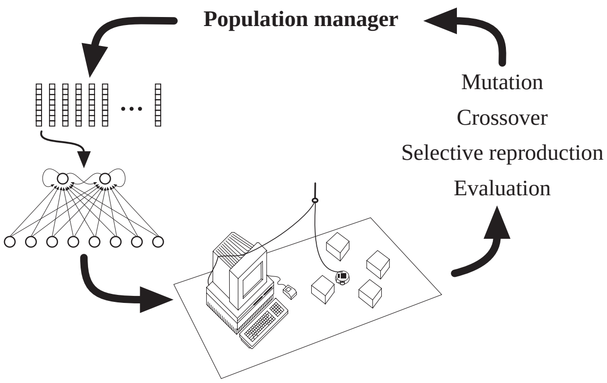

Figure 4: Evolutionary experiments on a single robot. Each individual of the population is decoded into a corresponding neurocontroller which reads sensory information and sends motor commands to the robot every 300 ms while its fitness is automatically evaluated and stored away for reproductive selection.

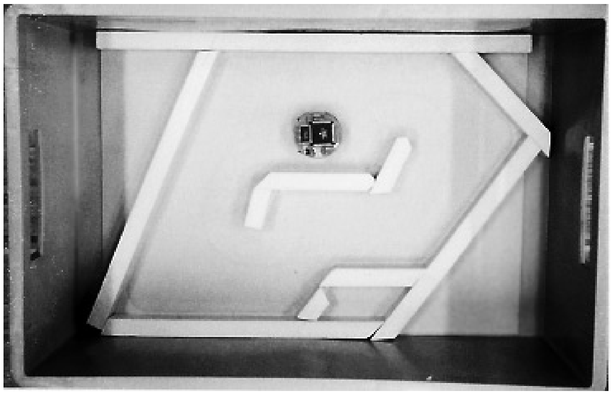

Figure 5: Bird's-eye view of the looping maze and the Khepera.

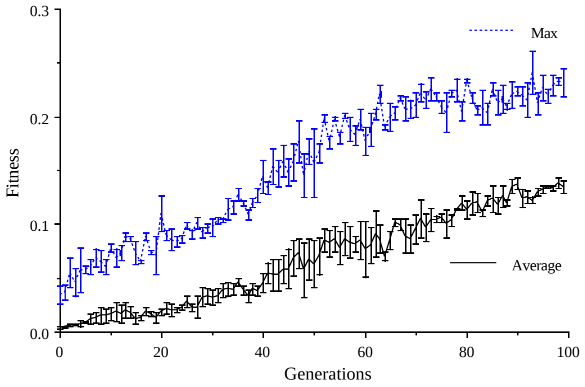

Figure 6: Average fitness of the population and fitness of the best individual at each generation (error bars show standard error over three repeated runs).

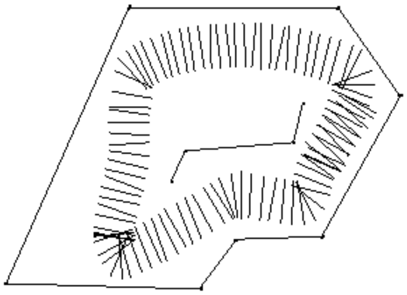

Figure 7: Trajectory of the robot controlled by the best individual of the last generation. Data recorded and plotted every 300 ms using the laser positioning device described in section 2 above. Segments represent the axis between the two wheels.

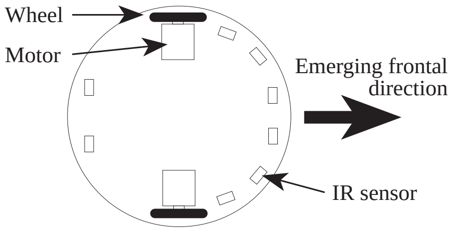

Figure 8: The direction of motion evolved by all best neurocontrollers corresponds to the side with the highest number of infrared sensors.

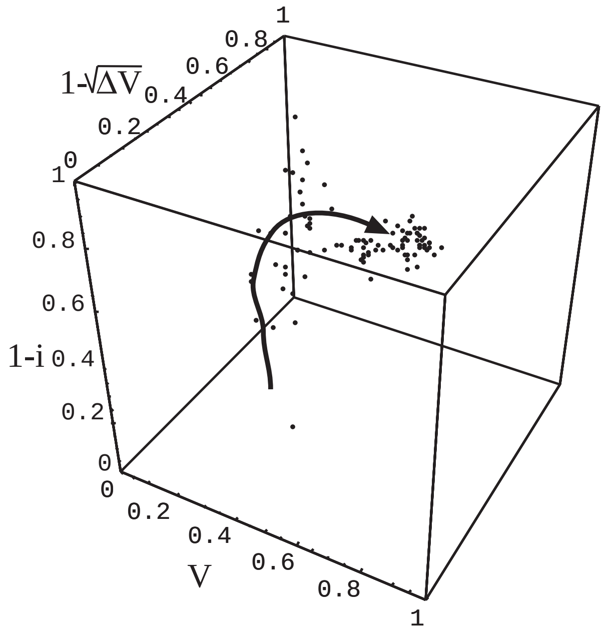

Figure 9: State-space representation of evolutionary adaptation. Each dot is the state of the best individual of a generation. The arrow shows the direction of motion during evolution. The dots concentrate in a sub-space indicated by the arrow tip in the last 20 generations ($x = 0.6 \pm 0.07$, $y = 0.6 \pm 0.05$, $z = 0.4 \pm 0.15$).

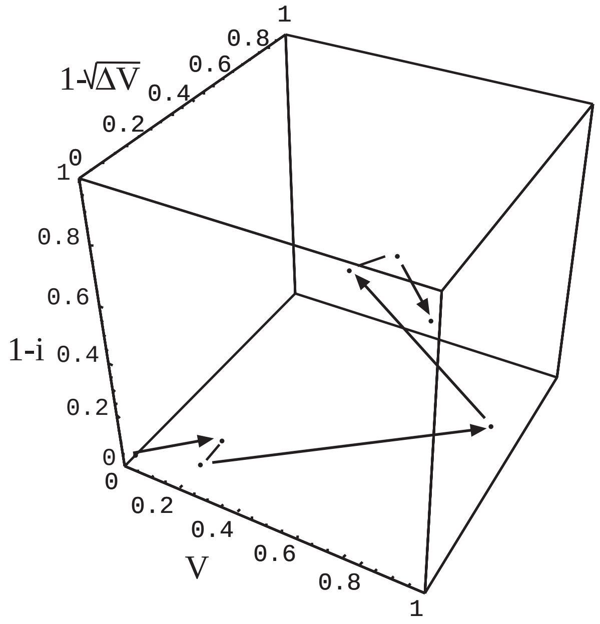

Figure 10: State-space representation of the behavior of the best individual of the last generation when positioned close to two walls. By moving away from the walls and initiating the navigation behavior, the controller in fact returns to the equilibrium zone ($x = 0.65$, $y = 0.6$, $z = 0.5$).

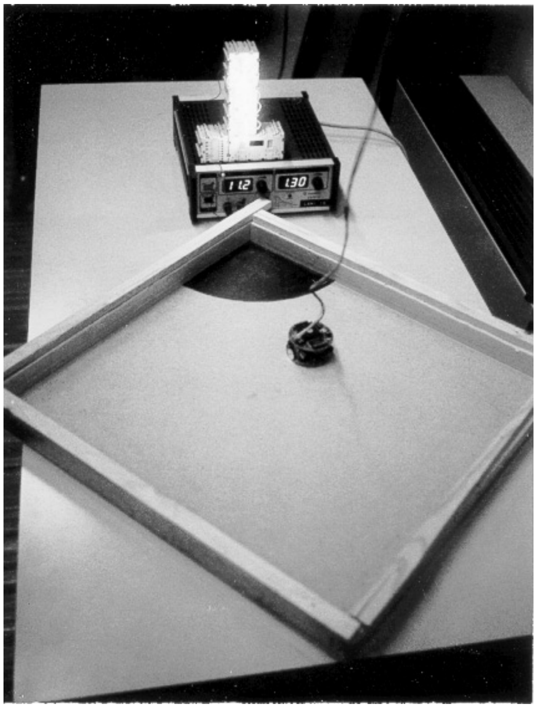

Figure 11: Bird's-eye view of the arena, the light tower, and the Khepera.

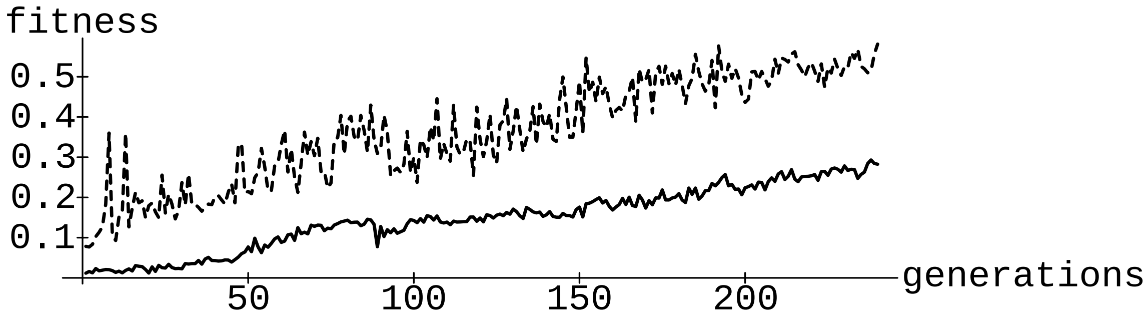

Figure 12: Average population fitness (continuous line) and fitness of the best individual (dotted line) at each generation in the experiment with the battery charger.

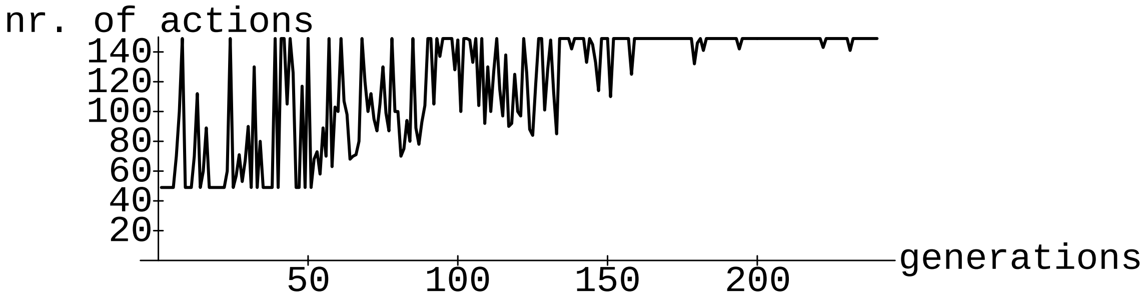

Figure 13: Number of sensorimotor steps achieved by the best individuals in the experiment with the battery charger. Each individual starts with a full battery which lasts 50 sensorimotor steps, if not recharged. The maximum number of steps allowed is 150.

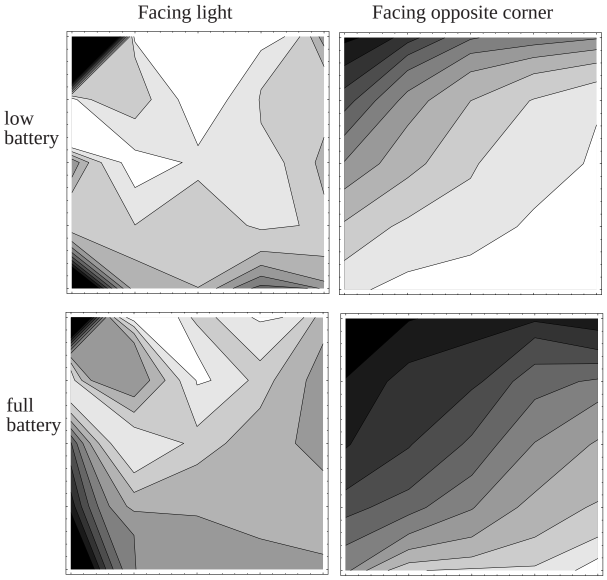

Figure 14: Contour map of the activation levels of an internal unit while the robot was positioned at various locations in the environment for four different conditions depending on orientation and battery charge. The recharging area is located at the top left corner of each map.

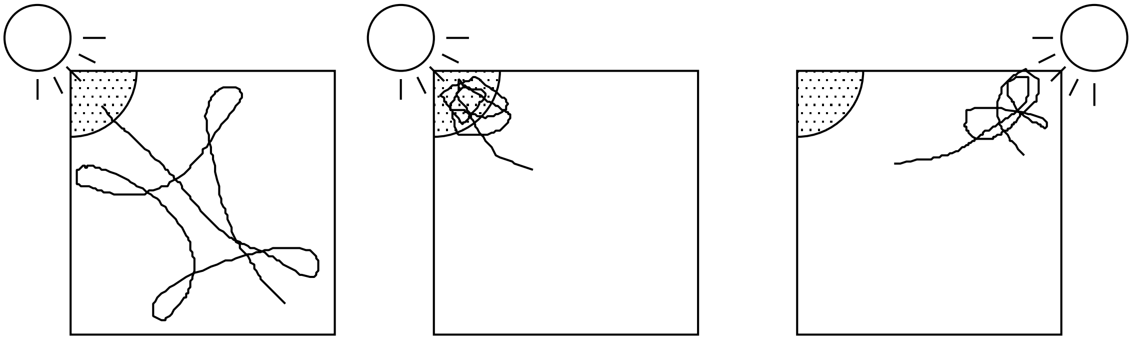

Figure 15: Trajectories of best individual of generation 240 in three environmental conditions. **Left**: Test in training conditions. The robot starts with a full battery in the bottom right corner (only the first 50 actions are displayed). **Center**: The battery is not automatically recharged when the robot arrives on the charging area. The robot starts in the centre of the environment 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.

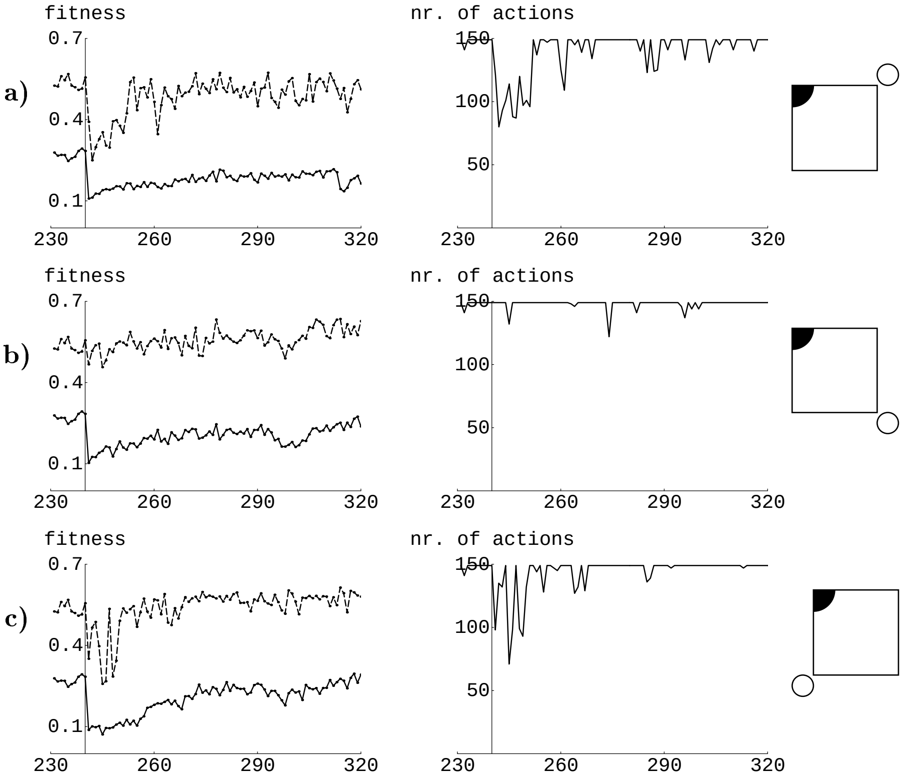

Figure 16: Re-adaptation in environments with a new light position. Each row (a, b, and c) plots—respectively—the average population fitness (continuous line) and the fitness of the best individual (dotted line) across generations, the number of actions during life for the best individual at each generation, and a sketch of the light position (small circle) in the environment (the black sector represents the charging area). For the sake of comparison, each plot includes data for the last ten generations of the original run (see figures 12 and 13) shown here to the left of the y axis.

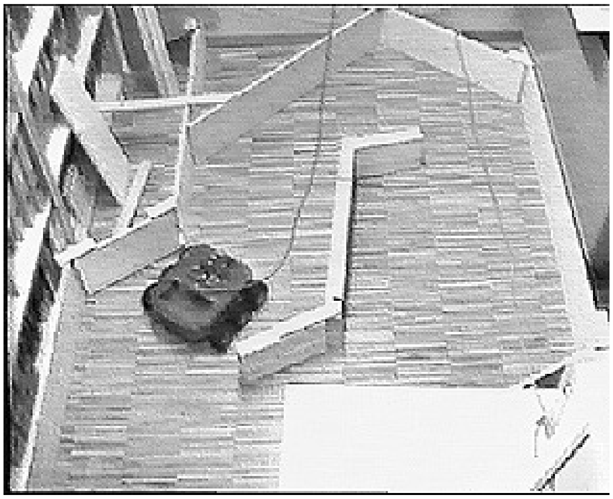

Figure 17: The Koala robot, without the protective white shell, in a scaled-up version of the looping maze used for evolution on the Khepera (see also figure 5).

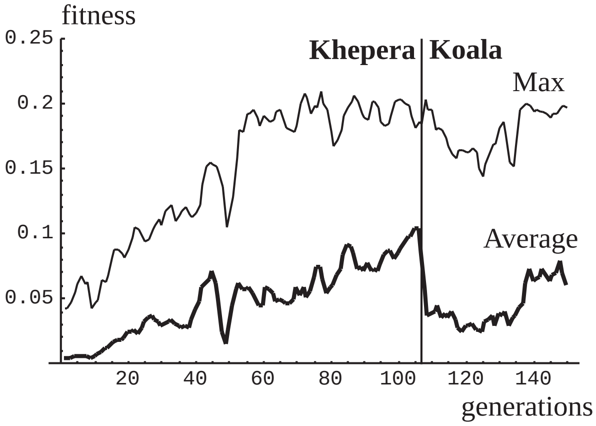

Figure 18: Average population fitness and fitness of the best individual across generations. Evolution begins on the Khepera; from generation 107 it continues on the Koala. Data from a single run smoothed using rolling averages (window size=3).

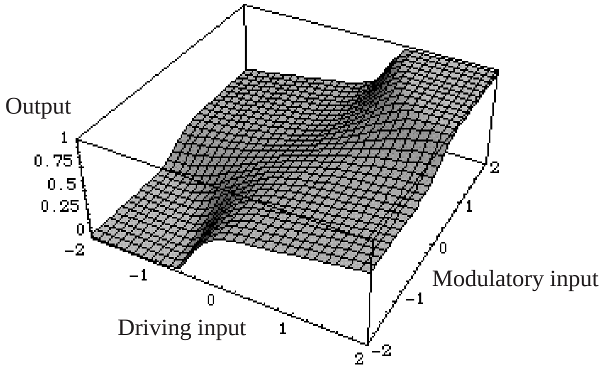

Figure 19: Activation function combining driving and modulatory signals (Kay & Phillips, 1996). 0.5 is the point of inversion of wheel rotation. Modulatory signals can enhance or dampen the unit response, but not change the direction of wheel rotation.

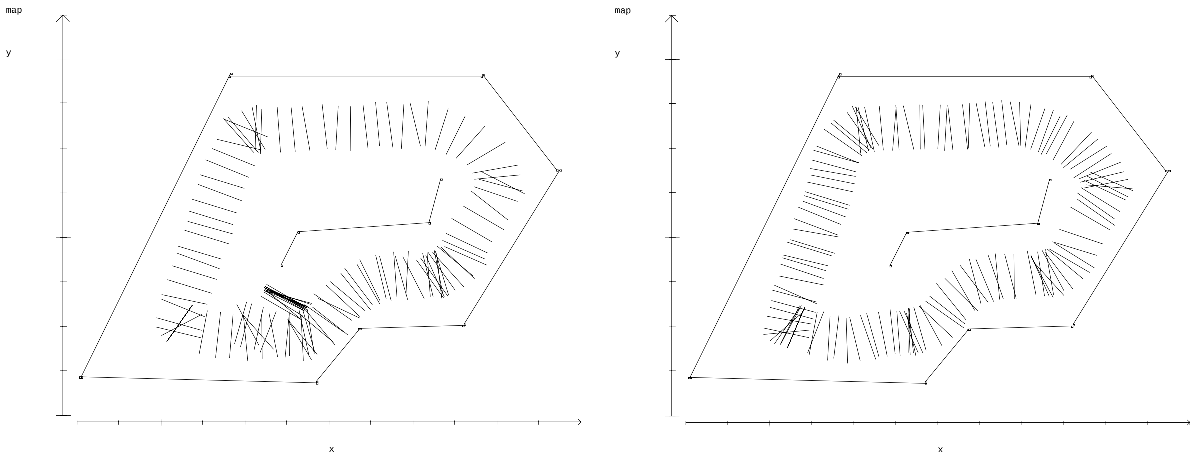

Figure 20: Trajectory of the robot that learns to navigate during life. Position data, visualized as bars representing the axis connecting the two wheels, were acquired with the laser positioning system every 300 ms. Data refer to the best individual of the last generation of one evolutionary run. *Left*: trajectory during the first lap (the robot starts in the lower portion of the environment and advances counterclockwise). *Right*: trajectory at the second lap.

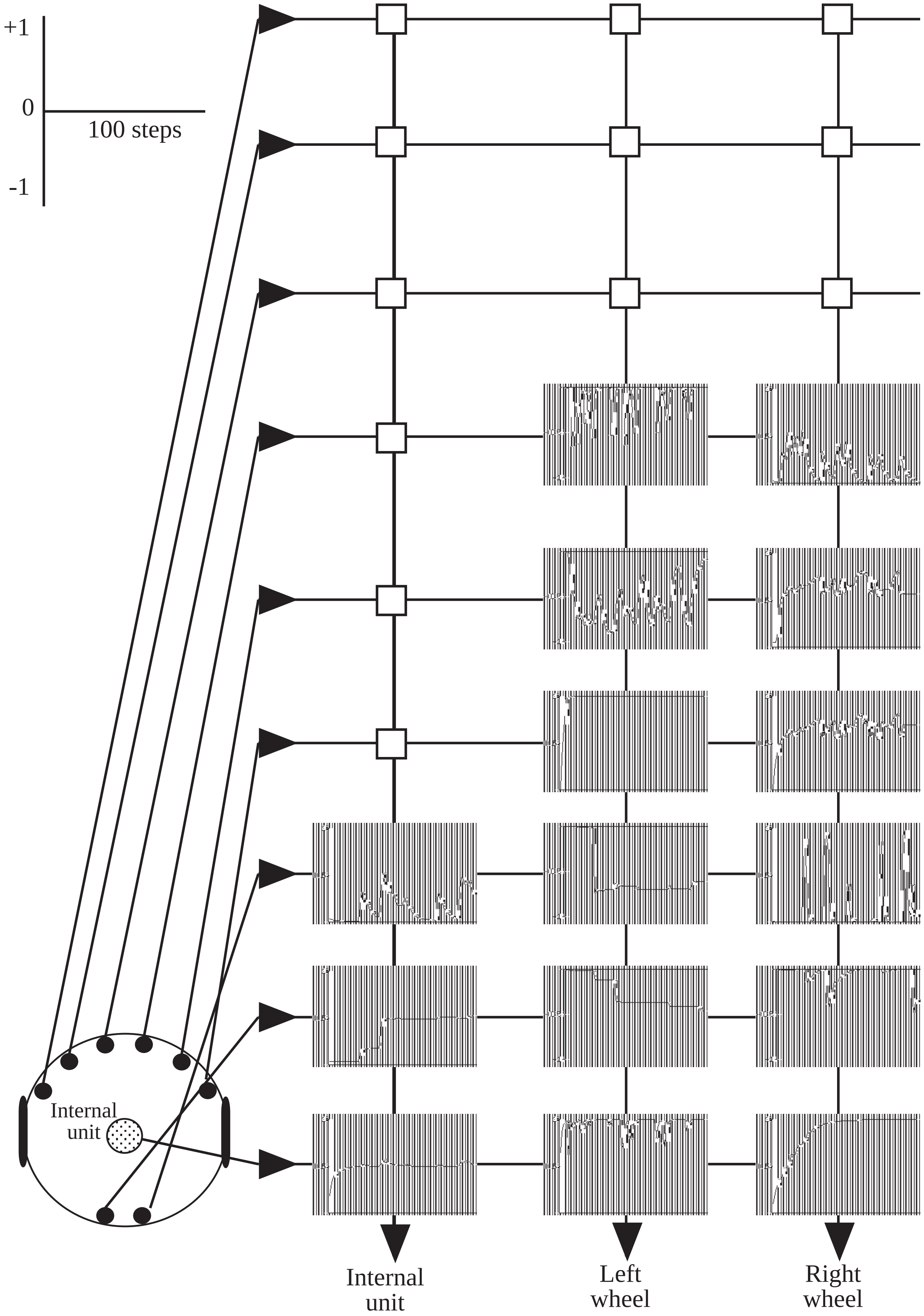

Figure 21: Synaptic strengths recorded every 300 ms during the first 100 actions of the robot. Small white boxes represent synapses that remain close to zero.

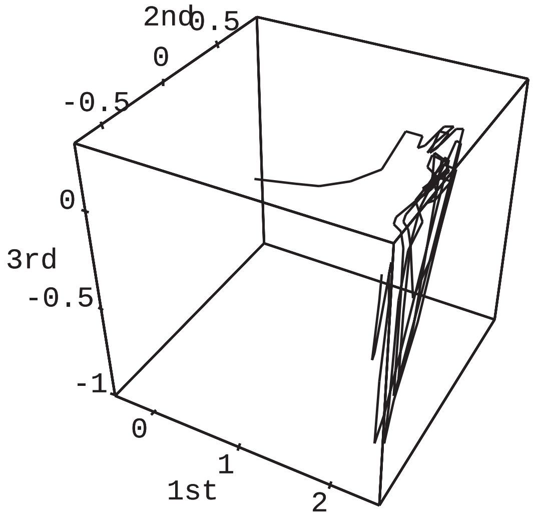

Figure 22: State-space representation of synapse dynamics during the first 100 actions plotted as trajectory within the space of the first three principal components. Arrows indicate the starting position and the range of oscillation between action sequences 20-80 and 80-100. Oscillations within the subspace of the third (smallest) component correspond to fine trajectory adjustments.