Evolution of Homing Navigation in a Real Mobile Robot

Dario Floreano, Francesco Mondada

Abstract— **In this paper we describe the evolution of a discrete-time recurrent neural network to control a real mobile robot. In all our experiments the evolutionary procedure is carried out entirely on the physical robot without human intervention. We show that the autonomous development of a set of behaviors for locating a battery charger and periodically returning to it can be achieved by lifting constraints in the design of the robot/environment interactions that were employed in a preliminary experiment. The emergent homing behavior is based on the autonomous development of an internal neural topographic map (which is not pre-designed) that allows the robot to choose the appropriate trajectory as function of location and remaining energy.**

Keywords— **Autonomous Robots, Genetic Algorithms, Neural Networks.**

I. Introduction

AUTONOMOUS biological agents are characterized by robust and reliable self-adaptation to the characteristics of the environment without external supervision or control [1]. This adaptation process takes place while the agent operates in its own environment [2]. In several real world situations it would be desirable to employ robots that have some of the features of autonomous systems, i.e. that are capable of developing new behaviors or adapting existing strategies according to the –often unpredictable– characteristics of real environments.

As a reaction to the partial failure of the classical AI approach to develop robust control systems for robots that need to operate autonomously in real world situations [3], a novel approach, termed *behavior-based robotics* [4], [5], [6], has recently emerged. Whereas classic AI is more concerned with a high level definition of the environment and of the knowledge required by the system, behavior-based robotics stresses the importance of continuous interaction between the robot and its own environment for the dynamic development of the control system and for the assessment of its performance [5]. It also puts emphasis on the autonomy of the system which should be completely self-contained and which should find the most appropriate solutions to satisfy simultaneously several –sometimes conflicting– goals.

Within this latter approach, a number of researchers have successfully employed an evolutionary procedure [7], [8] to develop the control system of *simulated robots* [9], [10], [11], [12], [6], [13]. The rich variety of structures that have been put under evolution (feed-forward neural networks [9], dynamic recurrent neurons [11], [14], classifier systems [6], and Lisp code [10], [13]) and the large number of evolved behaviors (locating food sources, wall-following, obstacle avoidance, chemotaxis and tropotaxis, corridor following, light orientation, box pushing, gait control, etc.) have empirically demonstrated the power and generality of the evolutionary methodology. However, we think that computer simulations of robots can hardly 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 malfunctioning play a major role (see also [2], [15]). Thus, although one may obtain comparable results in simplified environments and well defined tasks [16], it has not yet been shown that the same holds for more complex situations.

In this paper we describe the evolution of a discrete-time recurrent neural network to control a real mobile robot. In all our experiments the evolutionary procedure is carried out entirely on the physical robot without human intervention. Traditional experiments employing genetic algorithms and neural networks have been concerned with finding the network parameters yielding optimal behaviors for carefully pre-defined tasks. The main goal of our research is to show that more complex behaviors can emerge by reducing the constraints imposed by the fitness function and by increasing the affordances of the environment (characteristics of the world that can be exploited by the agent for its own survival) [17]. Here, the choice of more *ecological* settings favours the autonomous development of a homing behavior for battery recharge that is not directly specified in the fitness function.

We will describe two experiments. In the first experiment (serving as a test of the methodology and as a benchmark) we explicitly evolve the ability to navigate in a corridor with several sharp convex and concave corners. Although the fitness function is precisely engineered to perform straight motion and avoid obstacles, the evolved robots display a number of interesting solutions that have not been pre-designed. In the second experiment we provide the robot with a simulated battery (the battery is simulated for the purpose of saving time, as it will be shown later), we introduce in the environment a battery charger and a light source, and we greatly simplify the fitness function employed in the previous experiment. Although the fitness function does not specify the location of the battery station or the fact that the robot should reach it, the robot learns to find and to periodically return to it while keeps

This research has been carried out at the Microcomputing Laboratory (LAMI), Swiss Federal Institute of Technology at Lausanne (EPFL), Switzerland. E-mail: floreano, mondada@di.epfl.ch. WWW: http://lamiwww.epfl.ch/w3lami/team/mondada

Financial support has been provided by the Swiss National Research Foundation (project PNR23).

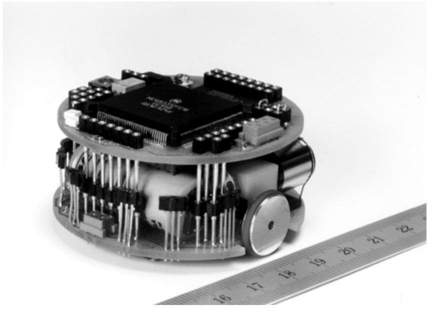

Fig. 1. Khepera, the miniature mobile robot. The ruler is in centimeters.

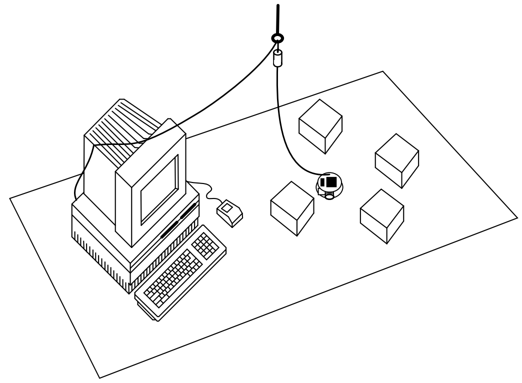

Fig. 2. Working setup.

moving and avoiding the walls. The resulting behaviors and the evolved neural mechanisms are studied in detail by analyzing the neural activity while the robot is tested in a number of situations.

II. General Method

A. The robot

The Khepera robot employed in our experiments is circular, compact, and robust (Figure 1). Khepera is a miniature robot: it has a diameter of 55 mm, it is 30 mm high, and its weight is 70 g. 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. The geometrical shape and the motor layout of Khepera provide for easier negotiation of corners and obstacles when its control system is still immature. Its small size and good mechanical design provide intrinsic robustness. In its basic version 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). These measures do not have linear characteristics, are not filtered by correction mechanisms, and depend on a number of external factors, such as the surface properties of objects and the illumination conditions. Several new single sensors and complete modules (such as a stereo-vision module and a gripper module) can be easily added, thanks to the hardware and software modularity of the system [18].

B. Experimental setup

The analysis of the emergent behavior plays an important role in autonomous robotics. It is thus necessary to develop new tools and methodologies to study the robot behavior [19], [20]. The setup employed here reflects our concern to study and understand the solutions provided by the evolutionary procedure. In our experiments the robot was attached to a Sun SPARCstation via a serial line by means of an aerial cable and specially designed rotating contacts (Figure 2). All low-level processes –such as sensor reading, motor control, and other monitoring processes– were performed by the on-board micro-controller, while other processes (neural network activation and genetic operators) were managed by the Sun CPU. This procedure should not be considered as a limitation of the autonomy of the system. Here the cable is a useful device that allows us to keep full track of the robot behavior and development by exploiting the data-storage capabilities of the workstation and all the special software for on-line behavior monitoring and analysis[1]. The cable was also used to supply electrical power, a useful option for experiments in evolutionary robotics where the robot may spend a long time displaying non-efficient behaviors. This configuration allowed a complete and precise analysis of the robot trajectories and of the functioning of its neural control system. An external positioning laser device was employed for post-training analysis of the evolved control systems: the robot was provided with an additional "helmet" for capturing the light signal emitted by the laser device and computing its own absolute position (Figure 3). This computation was carried out by a separate processor placed on the helmet. This information was then passed to the workstation where special software was used for automatic on-line analysis and display of the trajectories along with the sensors and motors states, and the neural network internal variables. The helmet, which could be easily added and removed, did not affect the robot motion and sensor activation. Such tools allowed us to perform neuroethological observations of the robot during normal operating conditions by relating the behavior displayed with the internal activity of the neural network.

C. The evolutionary procedure and the neuron model

The evolutionary procedure employed in the experiments consisted in applying a simple genetic algorithm [8] to the synaptic weight values (including the neuron thresholds) of the neural network that controlled the robot. Given

[1] The whole algorithm, including the genetic algorithm and the neural network representations, could be easily downloaded into the on-board controller, but this is not necessary for situations other than public demonstration.

Fig. 3. A close view of the robot with the "helmet" for capturing laser signals and the laser device on the background.

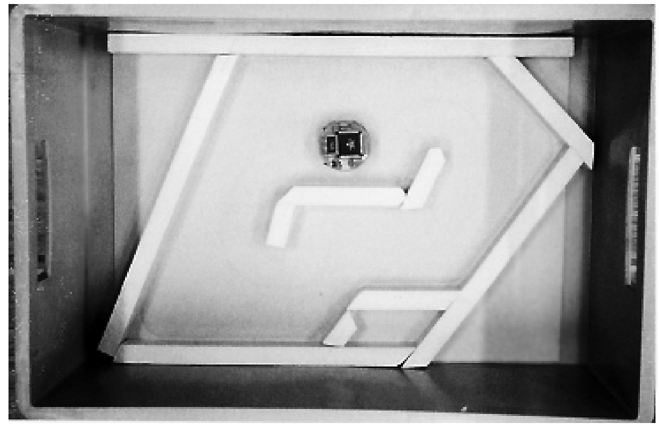

Fig. 4. Environment of the experiment on navigation and obstacle avoidance.

the small size of the networks used and the fixed architecture, the synaptic weight values were individually coded as floating point numbers on the chromosome. Each chromosome in the population had the same constant length corresponding to the number of synaptic connections and neuron thresholds in the network. An initial population of individuals was created by assigning to each gene in the chromosomes a new value drawn from a uniform random distribution of continuous numbers within a small positive and negative range (see Appendix for all the numerical details of the experiments).

Each individual, in turn, was decoded into the corresponding neural network, the input units were attached to the sensors of the robot and the output unit activations were directly used to set the velocity of the wheels. The robot was left free to move as a result of the activity generated by the neural network while its performance was recorded and accumulated according to a pre-designed fitness function. Each robot could move for a limited number of actions, each lasting a few hundred milliseconds. There was no synaptic change during the life of each individual. Between each individual and the next, a random velocity was applied to each wheel of the robot for 5 seconds in order to avoid artifactual influences between successive individuals in the same population and attempt to provide –on average– similar starting conditions for all individuals.

When all the individuals in the population had been tested, three genetic operators –selective reproduction, crossover, and mutation– were applied to create a completely new population of the same size. Selective reproduction consisted of a linear scaling of the fitness values [21] followed by a probabilistic allocation of a number of offspring proportional to the fitness value of each individual. All offspring, simple copies of their parents, were then randomly paired and a random single-point crossover was performed with a given probability. Each value of the newly obtained strings was then mutated with a given probability by adding a small random value within a negative and positive mutation range ("biased mutation" [22]).

The neuron model is simply described by a linear sum of the incoming weighted inputs (the threshold is taken as the contribution of an additional weighted input coming from a neuron which is always active) filtered through a sigmoid squashing function. The synaptic connections can take positive or negative unbounded values. In both experiments the neuron activation is updated approximately every 350 ms. The input units receive the activation values of the sensors and compress them in the continuous range 0-1. The number of input units varies between the two experiments, according to the number of sensors employed. Two output units are employed in both experiments. The activation of each output unit is used to set the velocity of the corresponding wheel within a continuous range where 0.0 is maximum speed in one direction, 0.5 corresponds to absence of motion, and 1.0 is maximum speed in the other direction.

III. Navigation and obstacle avoidance

This first experiment was aimed at explicitly evolving the ability to perform straight navigation while avoiding the obstacles encountered in the environment. The fitness function employed was very precisely engineered to achieve this type of behavior. Nevertheless, the evolved control systems displayed a number of interesting solutions that were indirectly instrumental to providing better performance (see [23] for a detailed discussion of the results).

A. The experiment

The robot was put in an environment consisting of a sort of circular corridor whose external size was approx. 80x50 cm large (Figure 4). The walls were made of light-blue polystyrene and the floor was made of a gray thick paper. The robot could sense the walls with the IR proximity sensors. Since the corridors were rather narrow (8-12 cm), some sensors were slightly active most of the time. The environment was within a portable box positioned in

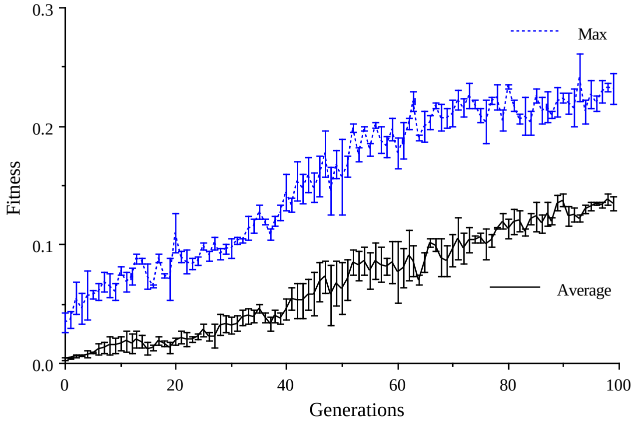

Fig. 5. Population average fitness and best individual fitness at each generation. Values are averaged over three runs (bars represent standard errors).

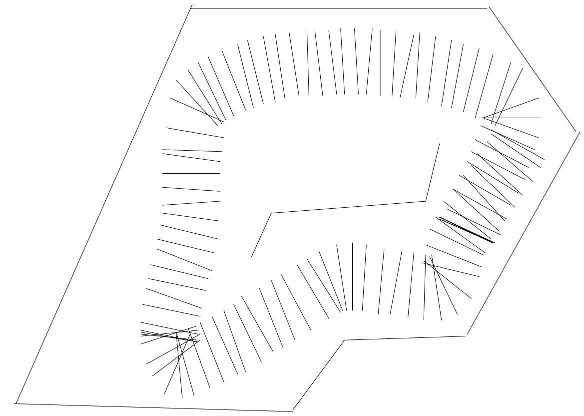

Fig. 6. The trajectory performed by one of the evolved robots. Segments represent successive displacements of the axis connecting the two wheels. The direction of motion is anti-clockwise.

a room always illuminated from above by a 60-watt light bulb. The fitness criterion Φ was described as a function of three variables, directly measured on the robot at each time step, as follows,

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

$$\begin{array}{ccccc} 0 & \leq & V & \leq & 1 \\ 0 & \leq & \Delta v & \leq & 1 \\ 0 & \leq & i & \leq & 1 \end{array}$$

where V is a measure of 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 activation value of the proximity sensor with the highest activity. The fitness values are accumulated during the life of the agent and then divided by the number of steps. The function Φ has three components: the first one is maximized by speed, the second by straight direction, and the third by obstacle avoidance. Since the robot has a circular shape and the wheels can rotate in both directions, this function has a symmetric surface with two equal maxima, each corresponding to one direction of motion.

The neural network architecture was fixed and consisted of a single layer of synaptic weights from eight input units (each connected to one of the IR proximity sensors placed around the body of the robot) to two output units (directly connected to the motors) with discrete-time recurrent connections only within the output layer. Numerical details of the genetic runs are given in the Appendix.

B. Results

Khepera learned to navigate and avoid obstacles in less than 100 generations (Figure 5), each generation taking approximately 40 minutes. However, around the 50th generation the best individuals already exhibited a near to optimal behavior. Their navigation was extremely smooth, they never bumped into walls and corners, and succeeded in maintaining a straight trajectory when possible. They could perform complete laps of the corridor without turning on themselves (Figure 6). These results are highly reliable and have been replicated in many runs of the experiment. Early during evolution the individuals evolved a frontal direction of motion, corresponding to the side where more sensors are available. Those individuals that moved in the other direction were very likely to get stuck in a convex corner without being able to detect it (because of the poor information provided by the two sensors) and, hence, soon disappeared from the population (see [14] for a similar example of evolutionary adaptation of the control system to the visual configuration of a simulated agent). When compared to the performance of a simple Braitenberg vehicle [24] (type 3c modified and implemented on Khepera to perform obstacle avoidance), our evolved robot displayed a better global performance, especially when facing concave corners. In fact, unlike the feedforward and internal symmetric structure of the Braitenberg vehicle which cannot drive the robot away from symmetric frontal obstacles, the evolved settings of the recurrent connections are such that our robot never became trapped [23]. The best robots also displayed a self-regulation of the cruising speed (approximately three quarters of the maximum available speed) that depended upon the characteristics of the environment, the response properties of the sensors, and the refreshing rate of the neurons.

IV. Battery recharge

The goal of this new experiment was to test the hypothesis that, when employing an evolutionary procedure, more complex behaviors do not necessarily have to be specified in the objective fitness function, but rather emerge from a mere change of the physical characteristics of the robot and of the environment described in the previous experiment.

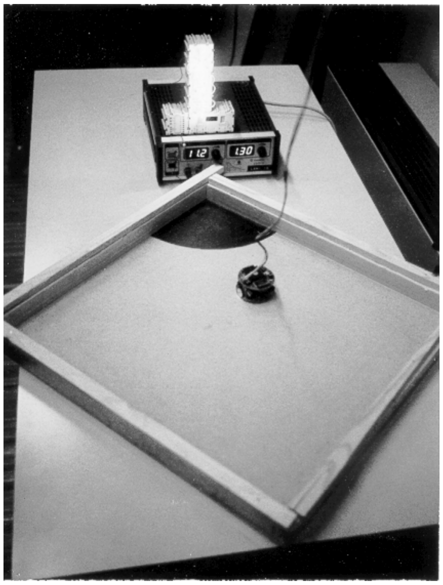

Fig. 7. The environment of the experiment on battery recharge. The light tower is positioned in the far corner over the recharging area which is painted black. There are no other light sources in the room.

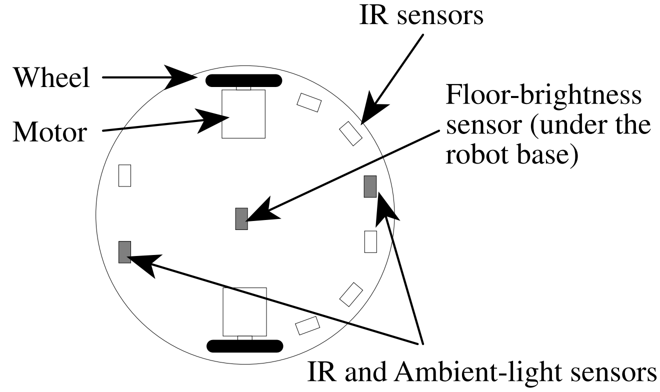

Fig. 8. Sensory-motor layout.

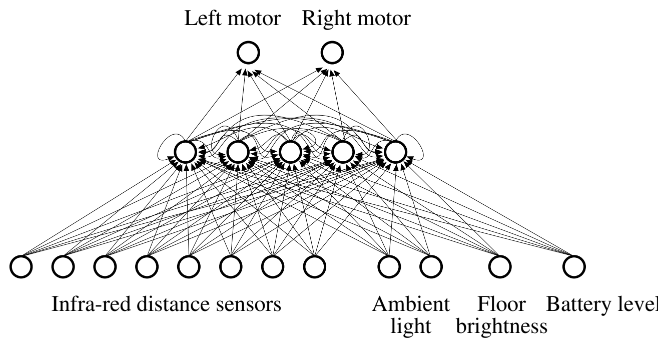

Fig. 9. The neural network. The input layer consists of twelve receptors, each clamped to one sensor (8 for infrared emitted light, 2 for ambient light, 1 for floor brightness, and 1 for battery charge) and fully connected to five hidden units. A set of recurrent connections [25] are added to the hidden units. The hidden units are fully connected to two motor neurons, each controlling the speed of rotation of the corresponding wheel.

More precisely, we were interested in observing whether the robot discovered the presence of a place where it could recharge its (simulated) batteries and modify its global behavior by using an even simpler version of the fitness function employed in the previous experiment.

A. The experiment

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 7) as in the previous experiment. 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 currently under construction. When the robot happened to be over the black area, its simulated battery became instantaneously recharged[2].

Khepera was equipped with its basic set of eight infrared sensors (proximity sensors) whose activation is inversely proportional to the distance from an object. Two sensors, each on one side of the body, were also enabled for measuring ambient light. Additionally, another ambient light sensor was placed under the robot platform, pointing downward, and its signal was thresholded so that it was always active, except when over the black painted area in the corner (Figure 8). The robot was 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. The reason why we simulated the battery and the battery charger, rather than using the hardware available, is time. Considering that the built-in battery lasts about 40 minutes and it requires further 30 minutes for a full recharge, a complete evolutionary run with the same parameters used here would have taken something like 6 years, whereas our experiment lasted only 10 days.

The neural network controlling the robot was a multilayer perceptron of continuous sigmoid units (Figure 9). The hidden layer consisted of 5 units with recurrent connections [25]; we did not attempt to optimize the number of units required and the connectivity pattern. Each robot 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

[2] The real battery charger grabs Khepera when it perceives it by means of a simple sensor placed on the black platform; the neural network is automatically disconnected, and the robot built-in battery is charged; then, the neural network is attached again to the sensors and motors, and the robot is left free to move.

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 individual was evaluated during its life according to the following fitness function Φ,

$$\Phi = V(1-i), \qquad 0 \le V \le 1,\ 0 \le i \le 1 \tag{2}$$

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. It must be noticed that Φ is a simpler version of the fitness employed in the previous experiment because the component responsible for straight motion has been removed: thus a robot could achieve a reasonable performance even by simply spinning in a place far from the walls. The fitness value was computed and accumulated at each step, except when the robot was on the black area (although later observations showed that the fitness function itself yielded values extremely close to 0 when the robot was on the black area[3]). 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 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. A different strategy could be –as mentioned above– moving in circles at maximum speed in the centre of the arena.

B. Results

We left Khepera in a dark room lit only by the small light-tower, and monitored its evolution on our workstation for the next 10 days (each generation, initially lasting approx. 45 minutes, took increasingly longer time as the individuals started to locate the recharging area). From time to time we went into the room to replace some of the small light bulbs (when we realized that they were blown), but we never stopped evolution. Both the population average-fitness and the fitness of the best individual steadily increased along the corrsponding 240 generations (Figure 10).

The increasing number of actions performed by the best individuals at each generation (Figure 11) suggests that the robot gradually learned to pass over the recharging zone. The combined data of the best fitness values and of the corresponding life durations showed that, mainly in the last 90 generations, the individuals increased their own life duration and spent a shorter period of time over the recharging area (recall that no fitness value is given while the robot is over it). However, since from these data we could not draw more precise conclusions, we analyzed in detail the behavior and the neural-network functioning of the best individual of the last generation.

[3]Indeed most of the time 0.0: due to the small size of the area and to the vicinity of the walls, both components are very close to zero.

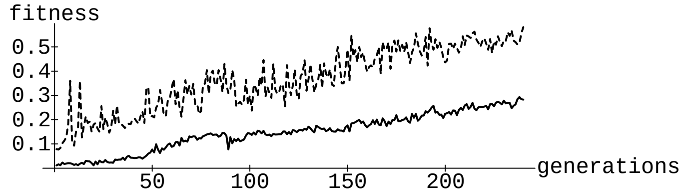

Fig. 10. Average population fitness (continuous line) and fitness of the best individual (dotted line) at each generation. Theoretical values of 1 could not be practically reached; empirical calculations based upon the maximum feasible speed and the characteristics of the environment give a maximum achievable fitness of 0.7.

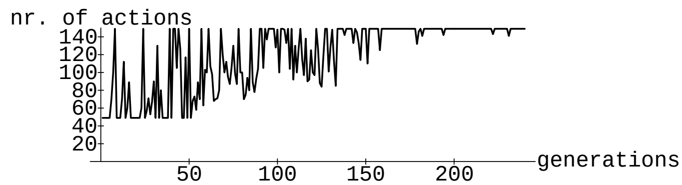

Fig. 11. Number of actions during life for the best individual at each generation. 50 actions (approximately 20 seconds) represent the minimum life length because each individual starts with a full battery. If an individual performs more than 150 actions, its life is automatically truncated and the next agent is evaluated.

C. Neuro-ethological analysis

We resorted to a method of analysis employed by ethologists and neurophysiologists by testing the robot behavior in a number of situations while recording all its internal (battery charge and neuron activations) and external (positions, sensor activations, motor activations) variables. For this purpose, Khepera was fitted with the "helmet" for absolute position measurement (Figure 3). We synchronized the measuring device on the helmet with the neural network activation dynamics (one measure every 380 ms), loaded the best neural network of the last generation, and performed a number of tests.

In the first test the robot was placed in the recharging

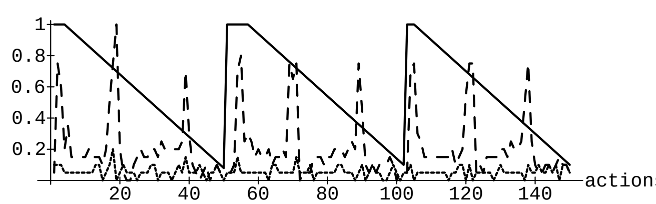

Fig. 12. Battery level (continuous line) and motor activities (dotted lines) during the life of the best individual of the last generation. The robot starts on the recharging area facing the light. Motor activity 0.5 means stasis, activity 0 corresponds to backward max. speed, activity 1 to forward max. speed. Spikes in only one motor activity indicate fast turning in place. Most of the time the robot moves backward at nearly max. speed.

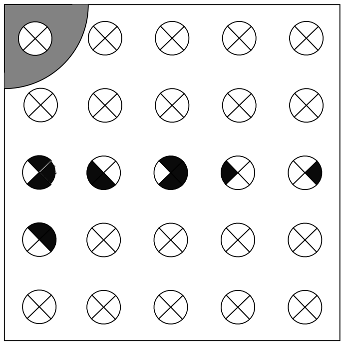

Fig. 13. Map of success/failures to reach the recharging area for each starting location and direction. Each sector in the circles indicates a different initial orientation; the circles correspond to the robot circumference. Black sectors correspond to starting orientations/locations from where the robot missed the recharging area after a full path (most of the misses were within a few millimeters, the largest was 3 cm).

area and left free to move while we recorded its battery level and motor activities (Figure 12). The robot rapidly moves out of the recharging area where it returns only when its battery level is about 0.1 (that means about less than two seconds before a complete discharge, or approx. 5 steps). The robot is always extremely precise in timing its return to the recharging area, as it can be seen by the regularity of the peaks in the line corresponding to the battery level. Also, the period of time spent over the recharging area is reduced to a minimum necessary to turn on itself and move out, as documented by the sharp increment of activity difference between the two motors in correspondence of the full charge level. The robot displays a preferential direction of motion (see Figure 12), although the sharpest turns in correspondence of the walls are performed by full acceleration of one wheel and full inversion of the other (thus turning in place). Most of the time the robot goes at nearly full speed along a slightly bended trajectory, and it always turns to the right when a wall is encountered.

Once the robot has found the recharging zone, without regard to the starting position in the arena, it always manages to return to it a very large number of times, without necessarily performing always the same trajectories. However, the robot did not find the recharging zone in the first place from a few starting positions and orientations in the arena (Figure 13). Nevertheless, it should be noted that the robot could indeed reach the recharging area from these locations if it was already moving: in fact, due to the recurrent connections on the hidden nodes, the same sensory information may yield different actions depending upon the previous history of the agent.

In another series of tests we positioned the robot at various locations and left it free to move while we recorded its position and the corresponding activations of the hidden nodes (Figure 14) every 380 ms. Hidden nodes were labelled as v_h0, v_h1, v_h2, v_h3, and v_h4. These measures revealed a non-stereotypical behavior and very complex internal processing. Most of the time the robot performed

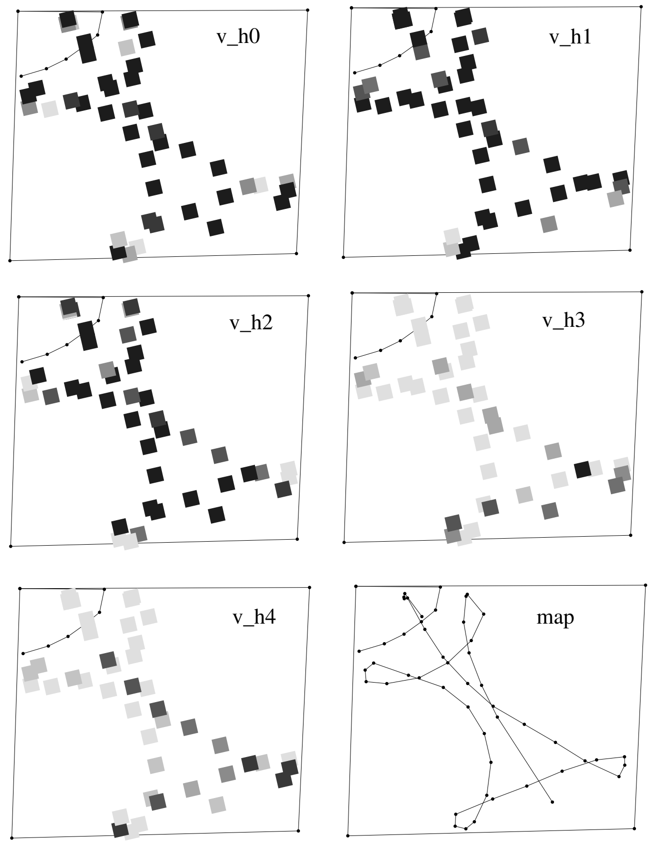

Fig. 14. Visualization of the hidden node activations while the robot moves in normal conditions. Darker squares mean higher node activation. The robot starts in the lower portion of the arena. The bottom-right window plots only the trajectory. The recharging area is visible in the top left corner.

long nearly-straight trajectories across the arena until it arrived very close to a wall, where it performed a sharp turn. Although the trajectories were very different even when the robot started from the same location because of noise in the sensors, friction against the floor and sometimes against the walls, and internal recurrent states, it generally performed 3 or 4 turns before moving toward the recharging area. In order to understand the strategies employed for homing, we performed an analogous test with the light tower switched off (Figure 15).

By comparing the behaviors in the two conditions (light on and light off), it becomes apparent that in both cases the robot relies on a set of semi-automatic procedures to perform the turns at the walls and the semi-linear trajectories (although they are more curved when the light is off). However, when the battery reaches a critical level, a somehow different behavioral strategy takes control which tries to correct the trajectory in order to reach the recharging area by using the information in the light gradient. When the light is off, this behavioral switch is documented by the beginning of the circular path in the middle of the arena in the likely attempt to find a light source during the last 18 steps. These tests indicate that the robot starts planning

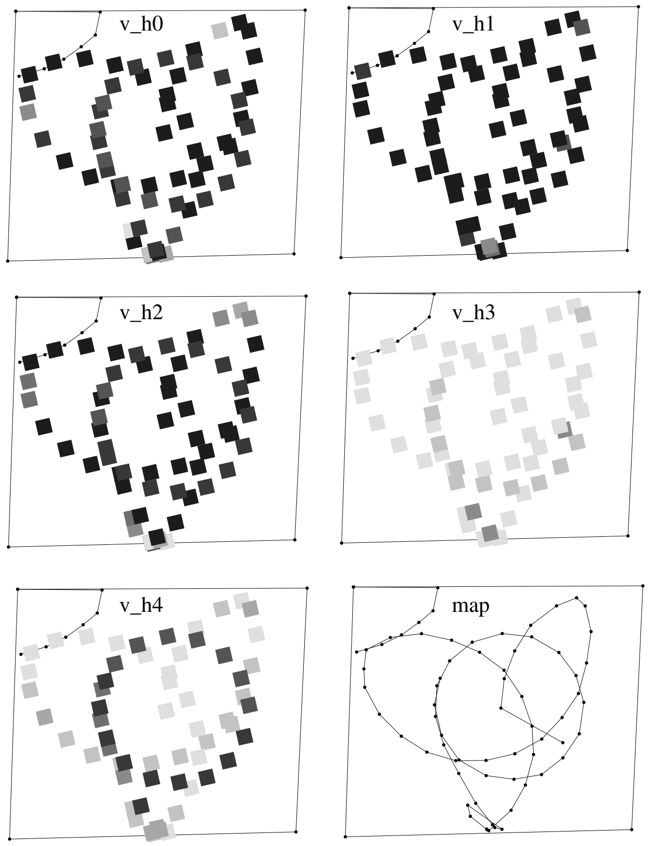

Fig. 15. Visualization of the hidden node activations while the robot moves when the light tower is switched off. The robot starts approximately in the center of the arena and ends up in circular trajectory during the last 18 steps.

the trajectory that will lead it to the recharging zone when the battery level is about one third of full charge, that is before the last turn against one of the walls in normal operating conditions.

A confirmation of this hypothesis comes from the analysis of the states of the hidden nodes. Although their functioning is distributed and highly dependent on their previous states (in tests where all the recurrent connections were cut off the robot could not even navigate properly), it is possible to detect a certain degree of specialization. In particular, the node labelled v_h4 seems to be responsible for battery check and path planning in the last steps. In all our "live-tests" it always kept a constant low level of activation (except for the two steps after the turns), but when the battery reached the critical level it progressively raised its activation state until the robot reached the recharging area. Hidden nodes v_h0 and v_h2 were nearly always highly active, except when approaching a wall: they are very likely to account for the automatic behavior of straight navigation and obstacle avoidance. Finally, hidden nodes v_h1 and v_h3 may contribute to path planning: the temporary change in activity of node v_h1 after a turn suggests that it partially controls the trajectory, while the slight rise in activity of node v_h3 when the battery is low is synergic

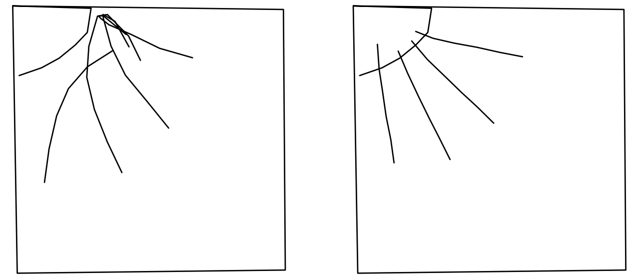

Fig. 16. Trajectories with full battery (left) and low battery (right). Four different paths are shown for each condition. Both the trajectories and the environment contours have been plotted using the laser positioning device.

with the activity of node v_h4.

To definitely ascertain the choice of different behavior strategies depending upon the battery charge level, we performed another test where we compared the trajectories of the robot with a fully charged battery (level 1.0) and a nearly-exhausted battery (level 0.12, i.e. max. 6 steps left). In both conditions the robot was positioned at four different locations equi-distant from the recharging area and regularly spaced. As clearly shown in Figure 16, the robot accurately avoided the recharging area when the battery was charged and it moved straight toward it when the battery was low.

In order to better understand the functioning of the system, we clamped the signal coming from the recurrent connections to the average node activity and we measured a single activation of each hidden unit while the robot was placed at several regularly-spaced positions in the arena. The single "shots" for every location were taken in four different conditions: low battery and facing light, low battery and facing direction opposite to light, full battery and facing light, and full battery and facing direction opposite to light[4]. These measures can be displayed as four maps – each for one of the measuring conditions– of the environment with the corresponding activity level of a single hidden unit. The resulting activity maps of node v_h4 (Figure 17) display remarkable topographical representations of the environment that are also head-direction specific because the major change in shape corresponds to the change in the facing direction. Similar activations map neighbouring locations in the environment. The organization of the resulting map is regularly oriented toward the recharging area when the robot faces the corner opposite to the light tower, but it displays a completely different pattern when the robot faces the light tower. Although the geometrical organization in the former situation is more regular, in the latter situation (when the robot faces the light) one can still recognize the recharging area and a sort of gate before it (the entrance of the virtual gate actually corresponds to

[4] The facing direction corresponded to the direction of motion of the evolved robot.

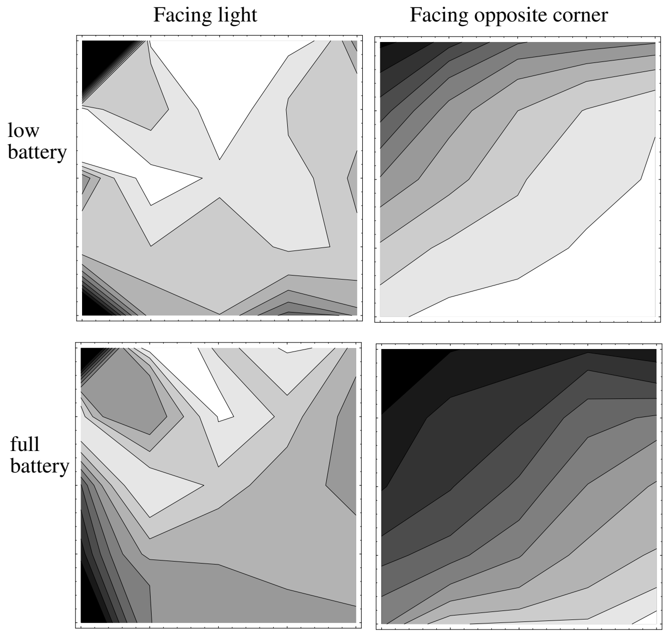

Fig. 17. Contour map of the activation levels of node v_h4 for different conditions while the robot was positioned at various locations in the environment. The recharging area is located at the top left corner of each map. See text for explanation.

the robot preferred approaching direction). Also the other neurons display a topological organization (but not so precise), except for node v_h0. Most of them are not head-direction specific (although node v_h3 displays an activity pattern close to that of node v_h4).

V. Discussion

Genetic algorithms can be successfully used to develop the control system of a real mobile robot. We have contrasted two methods of developing behaviors in two separate experiments. The first method –used in the first experiment– consists in a detailed specification of the fitness function that is tailored for a precise task. Although this choice may provide interesting and successful results, the behavior of the evolved agent depends upon the choice of the experimenter. In this sense, there is not much difference between the tuning of the objective function in a supervised neural algorithm and the engineering of the fitness function in the genetic algorithm. In this case, the design of the fitness function even for a simple task requires some effort and empirical trials because it is not possible to identify and specify in advance the desired actions of an autonomous agent [11], [15]. Additionally, the evolved agent can hardly be said to be autonomous because its behavior is dictated by the experimenter [2]. An alternative method is to consider the fitness measure not as a detailed and complex function to be optimized in order to achieve a desired behavior, but rather as a general survival criterion that is automatically translated into a set of specific constraints by the characteristics of the interactions between the organism and the environment. If one follows this approach, the adaptation process yields ecologically-grounded behavior (i.e., necessary for survival), rather than a mere *task*. The second and main experiment described here has been conceived within this latter framework. We have lifted one main constraint from the fitness function and we have allowed a richer interaction between the robot and the environment. Straight navigation, location of the battery charger, and timely homing are sub-goals created by the agent itself in order to maximize a more general and indirectly related fitness function. The spirit of our approach consists in making artificial evolution closer to natural evolution: Darwinian evolution is not an *optimization algorithm*, it has no sense of predetermined goal-directedness, but is rather a dynamic process governed by the principle of the survival of the individual [26]. To this extent, our methodology goes along the lines of those who think that the fitness function is not a global and precisely defined criterion, but is rather a characteristic of the individual and of the environment where it lives [27], [28], [29] and, hence, it may also change with time [30]. Although we have still maintained a vague global performance criterion and we have not taken into consideration the important influence of sociogenetic and ontogenetic learning [26], our results are the first showing that these principles can be applied to robot learning in order to obtain more complex and meaningful behaviors. Following on this principle, it would be interesting to try the same experiment by eliminating the fitness function and simply reproducing those individuals that live longer.

The behavior of the evolved agent relies on a topological neural representation of the world that was gradually built through a process of interaction with the environment. The few failures in reaching the recharging area from some starting locations (Figure 13) thus might be due to a sub-optimal or not fully formed map (when we stopped the evolutionary process, the fitness measure was still increasing, as shown in Figure 10). The functioning of node v_h4 (Figure 17) vaguely resembles the classic findings about the organization of the rat hippocampus, where most of the cells are "place cells", i.e. they fire only when the rat is in a particular portion of its environment [31]. Given the constraints of our neural model (few nodes, continuous activation, discrete dynamics, and homogeneous properties), the similarity between the rat hyppocampus and the control system of the robot is only *functional*: whereas the rat "knows" its own location by the firing of some specific place-cells, the evolved agent represents this information in terms of specific activation levels. With regard to the head-direction activity of node v_h4, agreement (in functional terms) is found with recent findings about the existence of few "head-direction" cells (whose firing modality depends upon the direction of the rat's head) in regions neighbouring the rat hippocampus [32]. All this amounts to saying that the behavior of the evolved robot cannot be purely explained by a stimulus-reaction paradigm because it is mediated by the autonomous development of an internal representation of the environment which reflects the goals defined by the robot itself (a similar conclusion was reached nearly 50 years ago in psychology [33] and formed the basis for a large conceptual revolution).

A rather interesting result comes from the dual role played by node v_h4 as an orienteering device and as a controller of battery charge. The latter role is masked during the "single-shot" measures used for the map plot of Figure 17 where, to make things even more complicated, the activation levels are very similar in the two battery conditions (low battery and full battery). Node v_h4 is nevertheless responsible for monitoring the battery level, but this feature is revealed only during the free running of the robot (Figure 14 and Figure 15)[5]. On the other hand, the orienteering function is not apparently revealed when the measures are taken during a free run: in this case the underlying representation of the environment is masked by the pattern of temporal activity sustained by the continuous (approximated by discrete dynamics) flow of information from the recurrent connections that is used to monitor the battery charge. Such a dual and concurrent processing modality has been hypothesized for biological neurons [34] too, but it can be hardly analyzed in living organisms because of technical difficulties. However, it can be displayed and thus analyzed in artificial neural network models which have recurrent (discrete or continuous) dynamics. It remains to be seen whether this feature emerges only when the neural network is embedded in a sensorimotor agent and the learning is not controlled by gradient descent techniques, or whether it is a more general computational strategy of these types of networks.

We have succeeded at replicating the results of the second experiment with different initial synaptic weight values (the adaptation process was much faster reaching comparable performances in 150 generations), but we succeeded only partially when we introduced obstacles in the environment. When we introduced obstacles (small circular shapes of the same material as the walls) from the beginning of the evolutionary run, the robot did not learn to reach regularly the recharging area. In order to make the fitness surface smoother, we adopted a more gradual approach by introducing obstacles only after that the robot learned to locate the recharging area (see [35] and [36] for a similar procedure). The resulting behavior was not completely satisfactory, although the robot displayed some degree of adaptation to the new environment (this experiment lasted almost 1,000 hours): the best individuals could reach the recharging area only from a very few starting positions. This limitation poses a serious question on how well our simple method would scale to harsher environments. Our opinion is that allowing the agent to learn during its life (for example with a local reinforcement learning algorithm) would help to circumvent these difficulties. If an agent displays adaptation capabilities during its life, the evolutionary process becomes more powerful and robust because the selection procedure can evaluate a large set of values for each string (due to the oscillations of performance on the fitness surface caused by synaptic change), rather than a single value [37] as in our case. Some experimental results on the evolution of simulated learning agents have shown both a speed-up in the convergence time [38] and the ability to deal with a very complex environment [39]. Our current work is focusing on this approach.

An alternative solution to the scalability problems outlined above and to the reduction of the evolution time could be provided by a more efficient genetic encoding that would use more compact or suitable representations which capture the essential features of a neural network model. Several methods [40], [41], [42], [43], [44], [45] have been proposed and shown to yield better and faster solutions than traditional encoding methods on simple tasks, but only a few [46], [36] have been applied to autonomous agents. Although these methods can evolve modular architectures that are suited to the requirements of the task, it is still premature to assess the superiority of one approach over the other for the evolution of real robots.

VI. Related work

There is a large literature on the application of evolutionary techniques to the design and training of neural networks (see [47] for a specific bibliography, [48], [49], [50] for a description of the various approaches employed, and the 1993 special issue on Evolutionary Computation of the IEEE Transactions on Neural Networks for an outline of more recent results). Only a small subset of this *corpus* of research focuses on the development of autonomous agents and the results are mostly based on computer simulations, rather than on real robots. Nevertheless, the research reported here is based and variously related to some of these contributions, which we briefly mention below.

Cliff [51] provides a theoretical background for the study of simulated organisms situated in closed environment and defines *Computational Neuroethology* as the attempt to relate behavior with the activity of neural mechanisms using the methodology of computational neuroscience (a similar approach is described also in [52]). On similar lines is the work by Parisi, Cecconi and Nolfi [9] who also stress the importance for an evolving organism to learn to predict the sensory consequences of its own actions in order to develop an internal world model. Beer and Gallagher [11] use a genetic algorithm to develop a set of chemotactic behaviors for a simulated agent with a circular and symmetric structure (geometrically similar to the robot employed in our experiments) and to control locomotion of a six-legged agent. In their work they evolve the synaptic weights of continuous-time recurrent neural networks. Several explorations in this direction have been described by Cliff, Husbands, and Harvey in a number of papers. Their major claim is that artificial evolution represents an alternative and more fruitful approach (contrasted to design by hand) to developing the control systems of autonomous mobile robots [12]. In their view the evolutionary method should be incremental [53] and operate on recurrent real-time neu-

[5] We tried to combine the two measuring procedures by disconnecting the motor neurons from the wheels and recording the node activations for a few seconds at all location in the environment. In this situation all the neurons started to display an asynchronous and cyclic pattern of activity that was completely uncorrelated with any external and internal parameter (position, orientation, battery status).

ral networks [54]. They suggest that some sort of visual processing is necessary for evolving non-trivial behaviors, but also say that careful simulations of the robot and of the robot/environment interactions can be necessary because of time constraints. Developing on these lines, they present results of several evolved behaviors for a simulated robot with a very simple visual system [55], [56], [57], [14]. Floreano [35] has studied the evolution of a simulated agent who developed the ability to reach a nest where it could eat the food found in the external environment. Since the fitness function was simply the number of food objects eaten, the location of the nest and the ability to periodically visit it was an indirect achievement, in analogy with what found in the experiments on battery recharge reported here. Similarly, reaching for a hidden location by means of visual landmarks has been reported by Treves, Miglino and Parisi [58] in evolved simulated agents. Their analysis of the resulting neural activity (in simple feed-forward architecture) reveals a functional analogy with the neural mechanisms employed by rats to navigate and resembles the topological patterns of activity described here (Figure 17).

Neural networks are not the only structure that have been used to evolve simulated autonomous agents. The evolution of programs (whose composition is similar to the symbolic expressions found in Lisp), also termed *Genetic Programming* by its inventor Koza [10], has been successfully employed –among other examples– to recreate the patterns of locomotion of a lizard [59], to evolve coordinate group motion of visually guided agents [60] and to develop corridor-following behaviors [13]. Dorigo and Schnepf [6] have developed a parallel robot controller (ALECSYS) based on a classifier system evolved by means of a genetic algorithm that can coordinate several different behaviors of a simulated agent. The design and the evaluation of their system is strongly based on ethological considerations.

The results obtained by evolving simulated agents may have little in common with the evolution of real robots [61]. Although for simple environments and simple tasks (obstacle avoidance and light following) the control system evolved in a computer simulation may be directly transferred into a real robot [62], [63], [16], [46], [20], this method is not guaranteed to work in more complex domains. The difficulty of making faithful simulations of complex visually-guided robots, has led a group of researchers to apply the evolutionary procedure directly on a real robot [64]. Harvey, Husbands, and Cliff have evolved target-approaching and object-following behaviors on a real robot with a circular body and a rotating camera suspended from a gantry-frame which allows 2-dimensional motion on the surface of the environment [36]. The evolution of the morphology of the visual system along with the structure of the neural network has resulted in smart and economical, but efficient, solutions. However, the development of increasingly more complex behaviors was achieved by using specifically-engineered fitness functions and by resorting to an incremental approach.

Our approach is also related to two researches reported in this special issue which employ evolutionary training of neural controllers, although both resort to a simulation for the training phase. Baluja shows that genetic algorithms can provide strategies to control an autonomous land vehicle (ALVINN) that are comparable to those found by a supervised learning algorithm [65]. The main difference from our work is that Baluja knows exactly what are the appropriate actions that the vehicle should take in the situations employed for training and can exploit this knowledge to assess the performance of the neural networks. Meeden compares the behavioral strategies developed by a reinforcement learning algorithm and a genetic algorithm for a robot that must either seek or avoid a light source while avoiding obstacles [66]. Her research is close to ours because she employs the same type of neural network (with the same number of hidden units) and genetic encoding, a similar evolutionary procedure, and a similar environment, but it differs in that the goals of the robot are explicitly included and weighted in the fitness function (as in our benchmark experiment). Given the complementary solutions found by the reinforcement learning and the evolutionary procedure, she supports our conclusions suggesting that a method that combines ontogenetic and phylogenetic learning seems to be more promising than either technique in isolation.

VII. Summary

We have described the application of an evolutionary procedure to a real mobile robot in two different settings. In the first experiment the environment and the robot shell [20] were very simple, and the fitness function was very detailed and aimed at developing a specific behavior, i.e. straight navigation and obstacle avoidance. The results displayed the evolution of efficient strategies which partially relied on the autonomous development of a frontal direction of motion and of the most appropriate cruising speed. In the second experiment we lifted several constraints by increasing the complexity of the environment (adding an oriented light source and a simulated recharge station) and of the robot shell (adding more sensors and a simulated battery), and decreasing the complexity of the fitness function. These modifications allowed a larger number of degrees of freedom in the interactions between the robot and its environment. Since the fitness values were summed during the life of the individual (as in the previous experiment), the introduction of variable life length gave an evolutionary advantage to those individuals who autonomously discovered the location of the recharge station and learned how to use it. The evolved behaviors displayed characteristics of self-sufficiency because the robot could keep itself "alive" by periodically charging its own simulated battery. As in the case of animals, this behavior relied on two important stages: the monitoring of the "physiological" variable (the level of the battery charge), and the calibration of the monitored message (i.e. the decision about when to initiate the sequence of actions needed for reaching the recharging area) [1].

The experiments showed that it is possible to perform behavior engineering [20] of intelligent agents without resorting to the abstract design of their cognitive abilities and artificially restricting its range of actions. The results stressed the importance of situation assessment in adaptive agents [19] for the autonomous development of intelligent behaviors and interesting computational strategies.

Appendix

	Navigation	Homing
Population size	80	100
Number of generations	100	240
Chromosome length	20	102
Crossover probability	0.1	0.1
Mutation probability	0.2	0.2
Mutation range	±0.5	±0.5
Initial weight range	±0.5	±0.5
Final weight range	Not bound	Not bound
Life length (actions)	80 fixed	150 max
Action duration	330 ms	380 ms

Acknowledgments

We would like to thank Edo Franzi and André Guignard for their important contribution in the design of Khepera, Yves Cheneval for the analysis software, the anonymous referees, Leslie Smith, and the Guest Editor Marco Dorigo for valuable comments that helped to improve the quality of the original manuscript.

References

[1] D. J. McFarland, "Autonomy and self-sufficiency in robots", AI-Memo 92-03, Artificial Intelligence Laboratory, Vrije Universiteit Brussel, Belgium, 1992.

[2] L. Steels, "Building agents out of autonomous behavior systems", in *The "artificial life" route to "artificial intelligence". Building situated embodied agents*, L. Steels and R. Brooks, Eds. Lawrence Erlbaum, New Haven, 1993.

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

[4] R. A. Brooks, "Intelligence without representation", *Artificial Intelligence*, vol. 47, pp. 139–59, 1991.

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

[6] M. Dorigo and U. Schnepf, "Genetic-based machine learning and behavior based robotics: a new synthesis", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, pp. 141–154, 1993.

[7] I. Rechenberg, *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Friedrich Fromann Verlag, Stuttgart, 1973.

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

[9] D. Parisi, F. Cecconi, and S. Nolfi, "Econets: Neural networks that learn in an environment", *Network*, vol. 1, pp. 149–168, 1990.

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

[11] R. D. Beer and J. C. Gallagher, "Evolving dynamical neural networks for adaptive behavior", *Adaptive Behavior*, vol. 1, pp. 91–122, 1992.

[12] D. Cliff, I. Harvey, and P. Husbands, "Explorations in evolutionary robotics", *Adaptive Behavior*, vol. 2, pp. 73–110, 1993.

[13] C. W. Reynolds, "Evolution of corridor following behavior in a noisy world", in *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, Eds. MIT Press-Bradford Books, Cambridge, MA, 1994.

[14] D. T. Cliff and S. G. Bullock, "Adding "foveal vision" to Wilson's animat", *Adaptive Behavior*, vol. 2, pp. 49–72, 1993.

[15] L. Steels, "The Artificial Life Roots of Artificial Intelligence", *Artificial Life*, vol. 1, pp. 75–110, 1994.

[16] S. Nolfi, D. Floreano, O. Miglino, and F. Mondada, "How to evolve autonomous robots: Different approaches in evolutionary robotics", in *Proceedings of the Fourth Workshop on Artificial Life*, R. Brooks and P. Maes, Eds., Boston, MA, 1994, MIT Press.

[17] J. J. Gibson, *The Ecological Approach to Visual Perception*, Houghton Mifflin, Boston, 1979.

[18] 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.

[19] M. J. Patel, "Situation Assessment and Adaptive Learning: Theoretical and Experimental Issues", in *Proceedings of the Second International Round-Table on Abstract Intelligent Agents*, Rome, 1994, ERCIM Research Report 03/94-R030 CNR.

[20] M. Colombetti, M. Dorigo, and G. Borghi, "Behavior Analysis and Training: A Methodology for Behavior Engineering", *IEEE Transactions on Systems, Man and Cybernetics*, vol. This issue, 1996.

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

[22] D. Montana and L. Davis, "Training feed forward neural networks using genetic algorithms", in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, San Mateo, CA, 1989, Morgan Kaufmann.

[23] D. Floreano and F. Mondada, "Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot", in *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, Eds. MIT Press-Bradford Books, Cambridge, MA, 1994.

[24] V. Braitenberg, *Vehicles. Experiments in Synthetic Psychology*, MIT Press, Cambridge, MA, 1984.

[25] J. L. Elman, "Finding Structure in Time", *Cognitive Science*, vol. 14, pp. 179–211, 1990.

[26] W. Atmar, "Notes on the Simulation of Evolution", *IEEE Transactions on Neural Networks*, vol. 5, pp. 130–148, 1993.

[27] P. Cariani, "Emergence and Artificial Life", in *Artificial Life II: Proceedings Volume of Santa Fe Conference*, C.G. Langton, J.D. Farmer, S. Rasmussen, and C. Taylor, Eds., vol. XI. Addison Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, Reading, MA, 1991.

[28] T. S. Ray, "An approach to the synthesis of life", in *Artificial Life II: Proceedings Volume of Santa Fe Conference*, C.G. Langton, J.D. Farmer, S. Rasmussen, and C. Taylor, Eds., vol. XI. Addison Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, 1991.

[29] M. M. Rizki and M. Conrad, "Computing the Theory of Evolution", *Physica*, vol. 22D, pp. 83–99, 1986.

[30] H. H. Lund and D. Parisi, "Simulations with an Evolvable Fitness Formula", Tech. Rep. 94-01, Institute of Psychology, CNR, February 1994.

[31] J. O'Keefe and L. Nadel, *The Hippocampus as a Cognitive Map*, Clarendon Press, Oxford, 1978.

[32] J. S. Taube, R. U. Muller, and J. B. Jr. Ranck, "Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis", *Journal of Neuroscience*, vol. 10, pp. 420–435, 1990.

[33] E. C. Tolman, "Cognitive maps in rats and men", *Psychological Review*, vol. 55, pp. 189–208, 1948.

[34] M. Abeles, *Corticonics*, Cambridge University Press, Cambridge, 1991.

[35] D. Floreano, "Emergence of Home-Based Foraging Strategies in Ecosystems of Neural Networks", in *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, J. Meyer, H. L. Roitblat, and S. W. Wilson, Eds. MIT Press-Bradford Books, Cambridge, MA, 1993.

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

[37] G. E. Hinton and S. J. Nowlan, "How learning can guide evolution", *Complex Systems*, vol. 1, pp. 495–502, 1987.
[38] S. Nolfi, J. L. Elman, and D. Parisi, "Learning and evolution in neural networks", *Adaptive Behavior*, vol. 3, pp. 5–28, 1994.
[39] D. H. Ackley and M. L. Littman, "Interactions between learning and evolution", in *Artificial Life II: Proceedings Volume of Santa Fe Conference*, C.G. Langton, J.D. Farmer, S. Rasmussen, and C. Taylor, Eds., vol. XI. Addison Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, Reading, MA, 1991.
[40] H. Kitano, "Designing neural networks using genetic algorithms with graph generation system", *Complex Systems*, vol. 4, pp. 461–476, 1990.
[41] S. A. Harp, T. Samad, and A. Guha, "Toward the genetic synthesis of neural networks", in *Proceedings of the Third International Conference on Genetic Algorithms*, J. D. Schaffer, Ed. Morgan Kaufmann, San Mateo, CA, 1989.
[42] E. J. W. Boers and H. Kuiper, "Biological metaphors and the design of modular artificial neural networks", Tech. Rep. Master Thesis, Computer Science, University of Leiden, The Netherlands, June 1992.
[43] F. Gruau, "Genetic systems of boolean neural networks with a cell rewriting developmental process", in *Combination of Genetic Algorithms and Neural Networks*, D. Whitley and J. D. Schaffer, Eds. IEEE Computer Society Press, Los Alamitos, CA, 1992.
[44] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An Evolutionary Algorithm That Constructs Recurrent Neural Networks", *IEEE Transactions on Neural Networks*, vol. 5, pp. 54–65, 1993.
[45] V. Maniezzo, "Genetic Evolution of the Topology and Weight Distribution of Neural Networks", *IEEE Transactions on Neural Networks*, vol. 5, pp. 39–53, 1993.
[46] S. Nolfi, O. Miglino, and D. Parisi, "Phenotipic Plasticity in Evolving Neural Networks", in *Proceedings of the conference From Perception to Action*, J-D. Nicoud and P. Gaussier, Eds. IEEE Computer Press, Los Alamitos, CA, 1994.
[47] M. Rudnick, "A Bibliography of the Intersection of Genetic Search and Artificial Neural Networks", Tech. Rep. CS/E 90-001, Department of Computer Science and Engineering, Oregon Graduate Center, January 1990.
[48] G. Weiss, "Combining neural and evolutionary learning: Aspects and approaches", Tech. Rep. FKI-132-90, Institut für Informatik, Technische Universität München, May 1990.
[49] J. D. Schaffer, D. Whitley, and L. J. Eshelman, "Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art", in *Proceedings of an International Workshop on the Combinations of Genetic Algorithms and Neural Networks (COGANN-92)*. 1992, IEEE Press.
[50] X. Yao, "A review of evolutionary artificial neural networks", *International Journal of Intelligent Systems*, vol. 4, pp. 203–222, 1993.
[51] D. T. Cliff, "Computational neuroethology: a provisional manifesto", in *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, J. A. Meyer and S. W. Wilson, Eds. MIT Press-Bradford Books, Cambridge, MA, 1991.
[52] R. D. Beer, *Intelligence as adaptive behavior: an experiment in computational neuroethology*, Academic Press, San Diego, CA, 1990.
[53] D. Cliff, I. Harvey, and P. Husbands, "Incremental Evolution of Neural Architectures for Adaptive Behaviour", Tech. Rep. CSRP 256, School of Cognitive and Computing Science, University of Sussex, December 1992.
[54] I. Harvey, P. Husbands, and D. Cliff, "Issues in Evolutionary Robotics", in *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, J. Meyer, H. L. Roitblat, and S. W. Wilson, Eds. MIT Press-Bradford Books, Cambridge, MA, 1993.
[55] D. T. Cliff, P. Husbands, and I. Harvey, "Evolving visually guided robots", in *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, J. Meyer, H. L. Roitblat, and S. W. Wilson, Eds. MIT Press-Bradford Books, Cambridge, MA, 1993.
[56] D. Cliff, P. Husbands, and I. Harvey, "Analysis of Evolved Sensory-Motor Controllers", Tech. Rep. CSRP 264, School of Cognitive and Computing Science, University of Sussex, December 1992.
[57] P. Husbands, I. Harvey, and D. Cliff, "Analysing Recurrent Dynamical Networks Evolved for Robot Control", Tech. Rep. CSRP 265, School of Cognitive and Computing Science, University of Sussex, January 1993.
[58] A. Treves, O. Miglino, and D. Parisi, "Rats, nets, maps, and the emergence of place cells", *Psychobiology*, vol. 20(1), pp. 1–8, 1992.
[59] J. R. Koza, J. P. Rice, and J. Roughgarden, "Evolution of Food-Foraging Strategies for the Caribbean *Anolis* Lizard Using Genetic Programming", *Adaptive Behavior*, vol. 1, pp. 171–199, 1992.
[60] C. W. Reynolds, "An Evolved, Vision-Based Behavioral Model of Coordinated Group Motion", in *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, J. Meyer, H. L. Roitblat, and S. W. Wilson, Eds. MIT Press-Bradford Books, Cambridge, MA, 1993.
[61] R. Brooks, "Artificial Life and real robots", in *Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life*, F. J. Varela and P. Bourgine, Eds. The MIT Press/Bradford Books, Cambridge, MA, 1992.
[62] M. Colombetti and M. Dorigo, "Learning to Control an Autonomous Robot by Distributed Genetic Algorithms", in *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, J. Meyer, H. L. Roitblat, and S. W. Wilson, Eds. MIT Press-Bradford Books, Cambridge, MA, 1993.
[63] O. Miglino, C. Nafasi, and C. Taylor, "Selection for Wandering Behavior in a Small Robot", Tech. Rep. UCLA-CRSP-94-01, Department of Cognitive Science, UCLA, January 1994.
[64] P. Husbands, I. Harvey, D. Cliff, and G. Miller, "The Use of Genetic Algorithms for the Development of Sensorimotor Control Systems", in *Proceedings of the conference From Perception to Action*, J-D. Nicoud and P. Gaussier, Eds. IEEE Computer Press, Los Alamitos, CA, 1994.
[65] S. Baluja, "Evolution of an Artificial Neural Network Based Autonomous Land Vehicle Controller", *IEEE Transactions on Systems, Man and Cybernetics*, vol. This issue, 1996.
[66] L. Meeden, "Developing Neural Network Controllers for Robots", *IEEE Transactions on Systems, Man and Cybernetics*, vol. This issue, 1996.

Dario Floreano has received an MSc in Neural Computation from the University of Stirling (U.K.) in 1992 and a PhD in Psychology from the University of Trieste (Italy) in 1995. This research was done while he was research assistant at the Swiss Federal Institute of Technology in Lausanne. Currently, he is a research fellow at the Centre for Cognitive and Computational Neuroscience of the University of Stirling (U.K.). His main interest is understanding and synthetizing the basic principles of adaptation in autonomous intelligent agents.

Francesco Mondada is a PhD student at the Microcomputing Laboratory (LAMI) of the Swiss Federal Institute of Technology of Lausanne, Switzerland (EPFL). An engineer in microtechnics of the EPFL, he received in 1994 an MSc in Computer Science from this same school. Since 1991, he has been working on mobile robotics, especially on the miniature mobile robot Khepera. His main interest is the development of methodologies for the design of control structures for autonomous mobile robots.