

Evolving Mobile Robots in Simulated and Real Environments

Orazio Miglino*, Henrik Hautop Lund**, Stefano Nolfi***

*Department of Psychology, University of Palermo, Italy
e-mail: orazio@caio.irmkant.rm.cnr.it

**Department of Computer Science, Aarhus University, Denmark
e-mail: henrik@caio.irmkant.rm.cnr.it

***Institute of Psychology, National Research Council, Rome, Italy
e-mail: stefano@kant.irmkant.rm.cnr.it

Abstract

The problem of the validity of simulation is particularly relevant for methodologies that use machine learning techniques to develop control systems for autonomous robots, like, for instance, the Artificial Life approach named Evolutionary Robotics. In fact, despite that it has been demonstrated that training or evolving robots in the real environment is possible, the number of trials needed to test the system discourage the use of physical robots during the training period. By evolving neural controllers for a Khepera robot in computer simulations and then transferring the obtained agents in the real environment we will show that: (a) an accurate model of a particular robot-environment dynamics can be built by sampling the real world through the sensors and the actuators of the robot; (b) the performance gap between the obtained behaviors in simulated and real environment may be significantly reduced by introducing a "conservative" form of noise; (c) if a decrease in performance is observed when the system is transferred in the real environment, successful and robust results can be obtained by continuing the evolutionary process in the real environment for few generations.

1. Introduction

Artificial Life proposes that we will gain knowledge about *life* by the study of human-made systems. Therefore, most Artificial Life experiments are performed by computer models of some kind of natural system. As a study of *artificial* life, this leads to interesting new life forms, while as a study of *natural* life, it leads to the proposal of new, and the verifications of old, biological theories. Though, from other scientific communities it has been questioned whether these life forms and theories are valid when transferred from the simulated

environments to the real environment. This discussion also enters in Artificial Life approach to robotics, namely Evolutionary Robotics.

Despite most roboticists regularly use simulations to test their models, the validity of computer simulations to build autonomous robots is very debated. Computer simulations may be very helpful to train and test robotics models. However, as Brooks [3] pointed out, ".. it is very hard to simulate the actual dynamics of the real world". This may imply that ".. effort will go into solving problems that simply do not come up in the real world with a physical robot .." and that " programs which work well on simulated robots will completely fail on real robots".

There are several reasons why those who want to use computer models to develop control systems for real robots may encounter problems:

(a) Numerical simulations do not usually consider all the physical laws of the interaction of a real agent with its own environment, such as mass, weight, friction, inertia, etc.

(b) Physical sensors deliver uncertain values, and commands to actuators have very uncertain effects, whereas simulative models often use grid-worlds and sensors which return perfect information.

(c) Different physical sensors and actuators, even if apparently identical, may perform differently because of slight differences in the electronics and mechanics or because of their different positions on the robot. This fact is usually ignored in computer models.

The problem of the validity of simulation is particularly relevant for studies in Evolutionary Robotics. In fact, despite that it has been demonstrated that training or evolving robots in the real environment is possible [7, 9], the number of trials needed to train the system discourage the use of physical robots during the training period. We will examine the problem of building a simulator in the context of Evolutionary Robotics [3, 4], that is, we will try to develop control systems for autonomous robots through the use of artificial evolution. In particular, we will show how the problems described above can be overcome by carefully designing the simulator. By evolving neural controllers for a Khepera robot in simulation and then transferring the obtained control system to the physical environment we will show that:

(a) an accurate model of a particular robot-environment dynamics can be obtained by sampling the real world through the sensors and the actuators of the robot; (b) the performance gap between obtained behaviors in the simulated and real environment may be significantly reduced by introducing a "conservative" form of noise; (c) if a decrease in performance is observed when the system is transferred to the real environment, successful and robust results can be obtained by continuing the evolutionary process in the real environment for few generations.

2. Related Work

Recently, some researchers have investigated the problem of transferring control systems of autonomous robot agents from simulation to reality in the context of the Evolutionary Robotics approach. Miglino, Nafasi, and Taylor [14] describe a mobile Lego robot controlled by an artificial neural network that should explore an open arena. The network was trained in simulation by using an evolutionary technique and then transferred in the real robot. In this work the authors showed, that the addition of noise to the sensors in the simulated environment may reduce the discrepancy between the simulated and the embodied conditions. The authors also showed that there is an optimal amount of noise to use and that such optimal value can be computed by comparing performances in simulated and embodied conditions.

Nolfi, Miglino, and Parisi (described in [16]) presented some experiments with the miniature mobile robot called Khepera (see below) that performed an obstacle avoidance task. The authors developed a simulator based on empirical sampled sensor readings and claimed that this sampling procedure can reduce the performance gap between the embodied and un-embodied conditions. By using an evolutionary technique, the authors also showed that the degradation in performance after the transfer of the system from the simulated to the embodied condition can be easily recovered by continuing the evolutionary process in the real environment for few generations. Recently, Nolfi and Parisi [20] showed how this approach can be generalized to a more complex task in which a Khepera with an arm and a

gripper evolves an ability to recognize and pick-up small round objects while avoiding walls. The control systems evolved in simulation resulted able to accomplish the task also once transferred to the real environment.

Jakobi, Husbands and Harvey [10] also described some experiments with Khepera performing an obstacle avoidance and a light-approaching tasks. As was previously demonstrated by Miglino, Nafasi and Taylor [14], the authors found that noise may reduce the discrepancy between the simulated and the embodied condition. The authors also claim that by adding the optimal amount of noise, almost identical behaviors are observed in the two conditions.

Several other authors have described experiences of successful transfer of systems evolved in simulation in the real environment. For example, Yamauchi and Beer [21] describe an experiment in which dynamical neural networks were evolved in simulation to solve a landmark recognition task using a sonar. Such networks were then tested on a Nomad 200 robot with a built in wall-following behavior.

3. The Experimental Setup

3.1 The robot and the environment

In our experiments we used Khepera (see figure 1), a miniature mobile robot developed at E.P.F.L. in Lausanne [16]. Khepera has a circular shape with a diameter of 55 mm., a height of 30 mm., and a weight of 70g.; it is supported by two wheels and two small teflon balls. The wheels are driven by extremely accurate stepper motors under P.I.D. control and can move both forward or backward. The robot is provided with eight infra-red proximity sensors. Six sensors are positioned on the front of the robot, the remaining two on the back. A Motorola 68331 controller with 256 Kbytes RAM and 512 Kbytes ROM manages all the input-output routines and can communicate via a serial port with a host computer. Khepera was attached to a host computer by means of a lightweight aerial cable and specially designed rotating contacts. This configuration allowed a full track and record of all important variables by exploiting the storage capabilities of the host computer; at the same time it provided

electrical power without using time-consuming homing algorithms or large heavy-duty batteries.

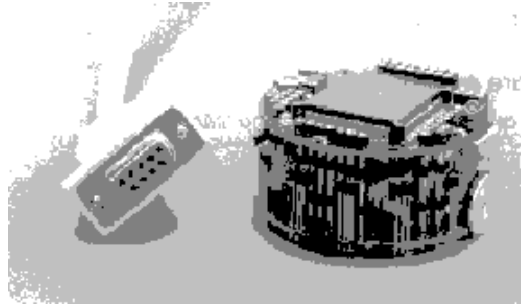


Figure 1. The Khepera robot.

The environment of the robot were a rectangular arena 60 x 35 cm surrounded by walls with 3 round obstacles of 5.5 cm placed in the center (see Figure 2). Walls and obstacles were covered with white paper.

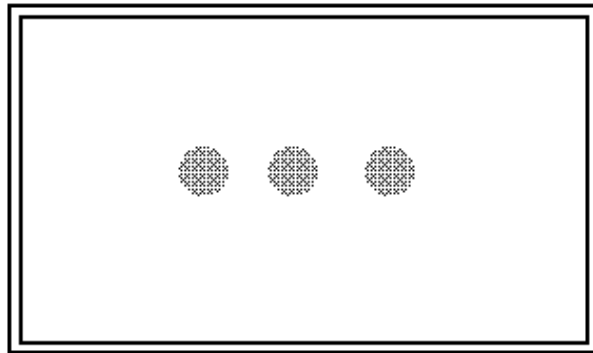


Figure 2. The environment used in the experiments.

Khepera should perform an obstacle avoidance behavior moving forward as fast as possible, moving in as straight a line as possible and keeping as far away from objects as possible. In order to evaluate individual performance we used the following formula:

$$F = \sum_{i=1}^{500} V_i (1 - DV_i^2) (1 - I_i) \quad (1)$$

where V_i is the average rotation speed of the two wheels, DV_i is the algebraic difference between signed speed values of the wheels, and I_i is the activation values of the proximity sensor with the highest activity at time i . For experiments with Khepera which used a similar environment and the same formula to score individuals see [7, 10,17] (however, in [10] the authors used a simplified version of the formula taking into account only the first 2 terms). Each individual was evaluated for 500 cycles, of 100 ms. each, starting from a different

random generated position in the environment. Thus, individual performance (F) was calculated by summing the value resulted from the formula above each cycle for 500 times.

3.2 Simulating the robot and the environment

As was argued in the introduction, there are several facts that must be taken into account when designing a simulative model of a robot and of its environment. The robot and the environment should be accurately reproduced in the software model and any kind of information that is not available to the real robot can not be provided to the simulated agent. From the sensory point of view, one should consider (a) that sensors deliver uncertain responses and (b) that different physical sensors, even if apparently identical, may perform differently because of slight differences in the electronics and mechanics or because of their different positions in the robot. Finally, from the actuators point of view one should consider (a) the physical aspects of the interaction between the robot and the environment (such as mass, friction, inertia, etc.) (b) the fact that values and commands to actuators often have very uncertain effects (c) the fact that different physical actuators, even if apparently identical, may perform differently.

3.2.1 Simulating Khepera's Infra Red Sensors

By inspecting the infra-red sensors of two different Khepera we found that, as expected, the responses of the sensors vary because of slight differences in the environment (e.g. ambient light settings, colour, and shape of the objects) and other unpredictable causes (noise). However, one of the most important factors affecting the variation of the response of sensors appeared to be the intrinsic properties of the particular sensor. In other words, each sensor responds in a significantly different way from other sensors when exposed to the same external stimulus (see Figure 3). Compare for instance the 5th and 6th sensors in Figure 3. These two sensors, which are placed close together in the right side of Khepera, act in totally different ways: the 6th sensor is sensitive within angle and distance ranges that are both nearly twice as large as those of the 5th sensor. This also implies, as we actually found, that

two different Khepera may perform very differently from each other in identical conditions because of the differences in the sensory characteristics.

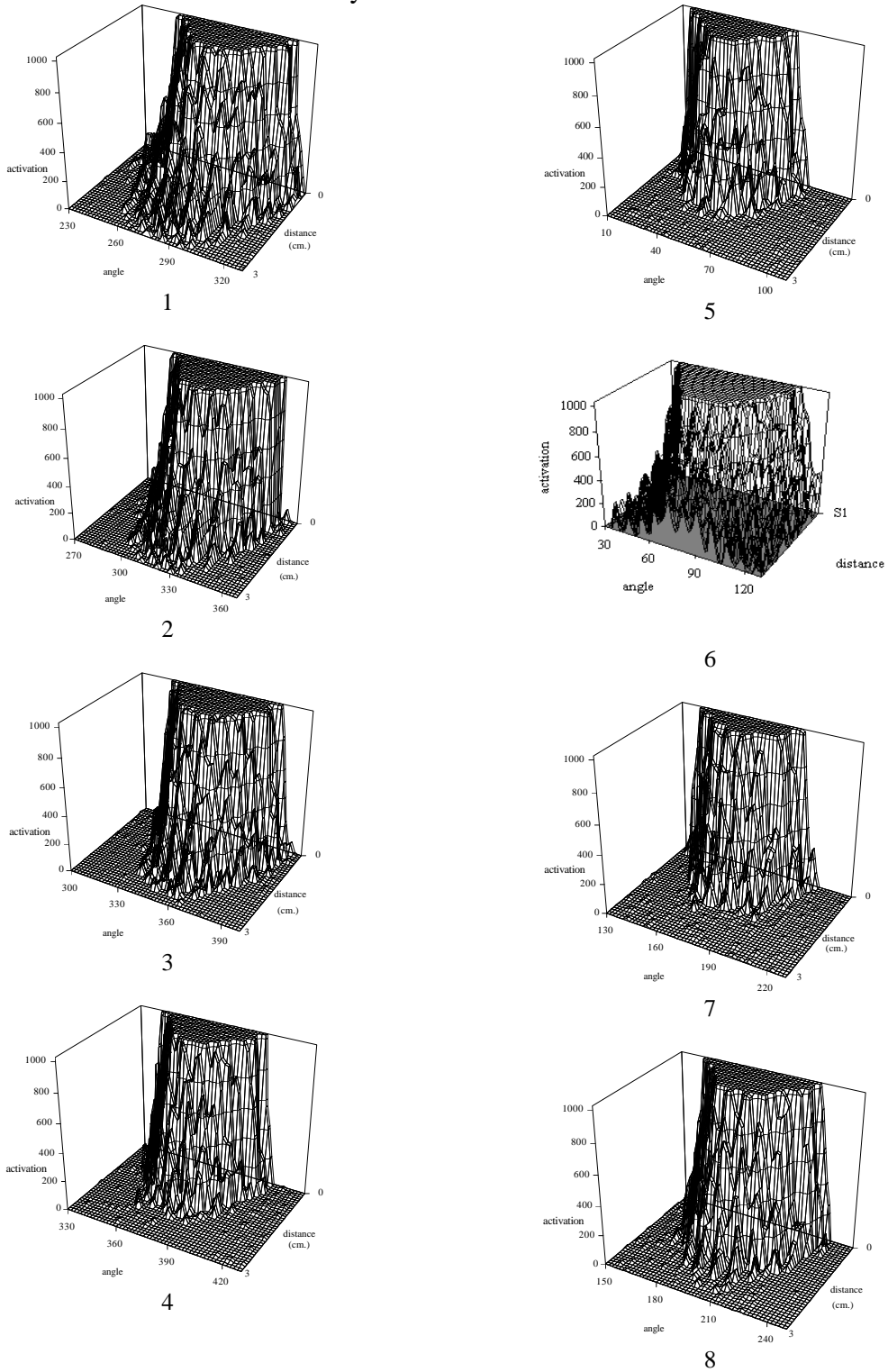


Figure 3. Sensors activation of an individual Khepera at different orientations and distances with respect to a wall.

In order to take into account the idiosyncrasies of each sensor we decided to empirically sample the different classes of objects in the environment (walls and obstacles) through Khepera real sensors. We let Khepera turn 360 degrees at different distances with respect to a wall and to an obstacle, while, in the meantime, recording the activation level of the sensors. More precisely, the activation level of each of the eight infra-red sensors was recorded for 180 orientations and for 20 different distances (the result of such sampling procedures in the case of a wall is shown in Figure 3). The resulting matrices were then used by the simulator to set the activation levels of the simulated agent depending on its current position in the simulated environment.

This procedure has several advantages: it is simple and accounts for the idiosyncrasies of each individual sensor. It allows to build a model of an individual robot taking into account the specificities of that robot that make it different from others apparently identical exemplars. It also accounts for the idiosyncrasies of the environment by empirically modelling the environment itself, instead of building a mathematical model of it. However, it must be noted that modelling more complex environment with this methodology may result significantly more expensive because each different class of objects should be sampled and because non-symmetrical objects may require different sampling for different points of view. To illustrate this last point see Figure 4. Uniform round objects are perceived in the same way independently from the point of view and thus require to be sampled once, while rectangular objects are perceived differently from different points of view and thus require multiple samplings.

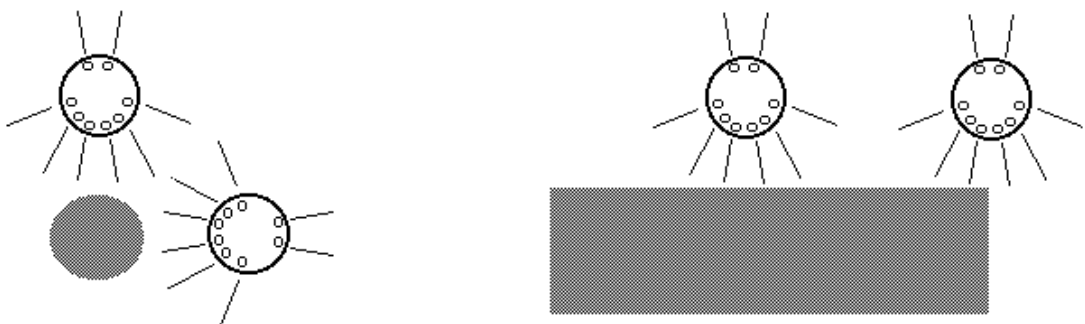


Figure 4. The perception of the symmetrical objects varies as a function of the agent orientation and distance but not of the point of view while perception of non-symmetrical objects varies also as a function of the point of view.

Finally, it should be noted that perception of multiple objects at the same time should be modelled in some way, too. In these cases, we calculated the activation of sensors by summing the measured activation produced by different objects but, of course, it is possible that in the physical environments combinations of different objects may produce slightly different effects.

To account for the fact that objects of the same types may be perceived differently because of variation in the ambient light or because of slight differences in the objects themselves we introduced, in a set of experiments, a special form of noise that we call "conservative position noise". This noise makes the simulated agent perceive objects as if they were farther or closer (with respect to a random selected axis) than what they really are, thus producing "illusions" similar to those arising in the physical environment because of differences in the illumination of objects, shadows, or because of slight physical differences between objects of the same type. This noise was implemented with two values that represent the current distortion on the x and y axes which are initially randomly assigned between a given range. A small random value was algebraically summed to current distortion values each time step in order to progressively change the type and the amount of the distortion of the perceived objects during a robot's life. It is important to notice that the robot is not moved from its position, but rather it senses the world as if it would have moved.

In other experiments, we used an alternative technique. To take into account the fact that sensors deliver uncertain responses we applied noise to the simulated sensors by adding a random selected value between a given range to the activation level of each simulated sensor. The effect of these two forms of noise will be described in the following section.

3.2.2 Simulating Khepera's Motors

Khepera has a very accurate motor apparatus because the PID control algorithm built-in in the robot, which assures that the actual speed never varies significantly from the set speed and because friction and inertia have very limited effects given the lightness of the robot. This characteristics makes it easy to build a simulated model of Khepera's motors accurate

enough. However, it should be considered, that in the case of other robots the situation may be quantitatively very different.

In order to model Khepera's motors, once again, we proceeded experimentally by sampling the effect of the different motors setting in the real environment. More specifically, we sampled how the individual Khepera that we were modelling moved and turned for each of the 20*20 possible states of the two motors. The obtained measures were used by the simulator to set the activation level of the neural network input units, and to compute the displacement of the robot in the simulated environment. This methodology allows to account for most of the problems described in section 3.2. To account for the fact that actuators may produce, in any case, unpredictable uncertain effects and that the ground may present irregularities, noise can be applied in the simulated environment to the movements of the simulated agent.

A similar approach is described in [10]. By using the speed sensors of Khepera and letting it move in the real environment, the authors empirically calculated the value of three constants of the formula used to determine the movements of the robot in the simulated environment. Further, they used a similar procedure, that however did not take into account the differences between each sensor, to model Khepera infra-red sensors.

3.3 The control system

To implement the control system we decided to use a neural network. Although within the Evolutionary Robotics community there are examples of controllers for autonomous robots implemented with explicit programs written in a high-level language [2, 12] or with a form of classifier systems [6], many researchers decided to use neural networks [4, 7, 14, 17, 21]. The capability of a neural network to deal with noise and the possibility to conjugate traditional neural learning algorithms (at individuals level) with genetic algorithms (at population level) are some of the reasons that makes neural networks attractive [4, 7, 17]. We used a feed-forward neural network with eight input units (clamped to the infra-red sensors), and two output units directly connected to the motors (see Figure 5).

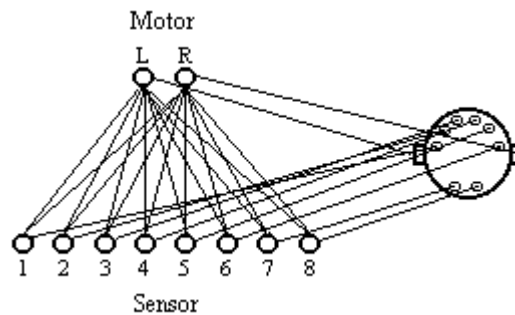


Figure 5. The neural network used to control the robot.

3.4 The evolutionary process

To evolve neural controllers able to perform the task described above we used a form of genetic algorithm. We begin with 100 randomly generated genotypes each representing a network with a different set of connection weights assigned randomly. This is Generation 0 (G0). G0 networks are allowed to perform 500 actions (about 2 seconds in the simulated environment using an IBM RISK/6000 computer or about 50 seconds in the real environment). The robot's initial position was randomly assigned. At the end of their "life" robots have the possibility to reproduce. However, only the 20 best performing individuals reproduce (agamically) by generating 5 copies of their neural networks. These 20x5=100 new neural control systems constitute the next generation (G1). Random mutations (25%) are introduced in the copying process resulting in possible changes of the connection weights. The process is repeated for 200 generations in the simulated environment and then is continued for other 20 generation in the real environment.

The genetic encoding scheme was a direct one-to-one mapping. The encoding scheme is the way in which the phenotype (in this case the connection weights of the neural network) is encoded in the genotype (the representation on which the genetic algorithm operates). The one-to-one mapping is the simplest encoding scheme in which to each phenotypical character corresponds one and only one 'gene'. In our case to each connection weights corresponds a floating point number. The genotype of each individual thus consists of 18 floating point value representing the 16 weights and the 2 biases of the neural network. (For more complex encoding schemes that also allow the evolution of the neural topology, see [9, 19]).

4. Results

In order to test the correspondence of performances and behaviors in simulated and in real environment and the role of the different forms of noise discussed above, we ran three sets of experiments. In a first set of experiments (*no-noise* condition) we did not apply any form of noise in the simulated environment. In a second set of experiments (*noise* condition) we applied a standard form of noise to the sensory activation of the robot by adding a random number in the range 0.0 to 0.4 (sensory activation varies between 0.0 and 1.0). Through empirical testings among 5 different noise ranges, the 0.0 to 0.4 range was found to give the best performance and less dependency on initial random seeds, so this range was chosen. In a third set of experiments we applied to the sensors the conservative form of noise described in section 3.2.1. (*conservative noise* condition). In this last case, we set the sensory activation of the robot as if it was translated on the Cartesian axes of x and y coordinates in the range between -30 and 30 mm. The choice of this range was, like the choice of noise range, due to empirical testings of different conservative noise ranges where the -30 to 30 mm. range produced the best performances and less dependency on initial random seeds.

For each of the three conditions (*no-noise*, *noise*, and *conservative noise*) we ran 5 experiments with different random assignments of the initial weights. The average and peak performances (i.e. the average performance of the population and the performances of the best 20 individuals of the population) through out generations are shown in Figures 6, 7, and 8 respectively.

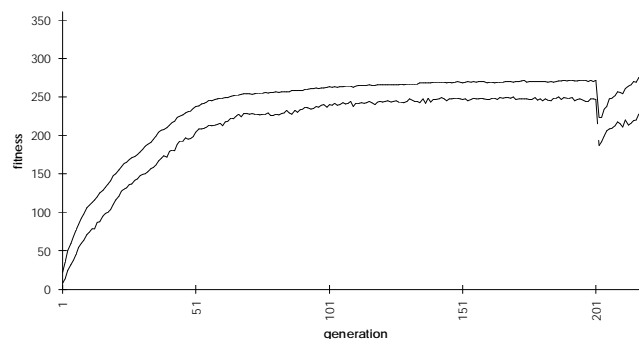


Figure 6. Peak and Average performances through out generations in the *no-noise* condition. Average result of 5 experiments. The first 200 generations are evolved in the simulated environment, the last 20 generations in the real environment.

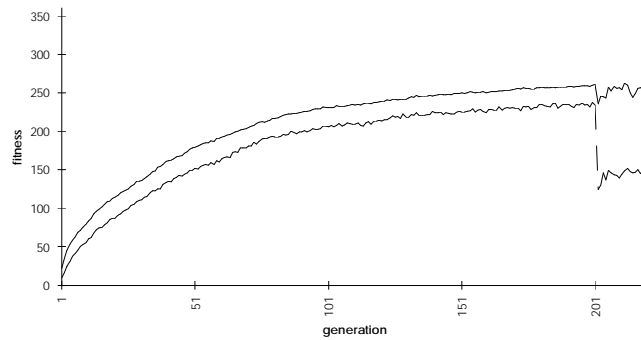


Figure 7. Peak and Average performances through out generations in the *noise* condition. Average results of 5 experiments. The first 200 generations are evolved in the simulated environment, the last 20 generations in the real environment.

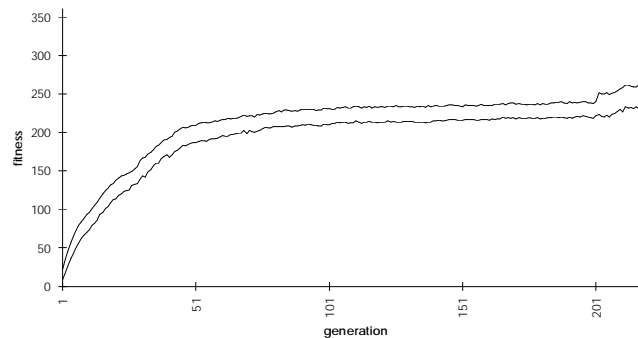


Figure 8. Peak and Average performances through out generations in the *conservative noise* condition. Average results of 5 experiments. The first 200 generations are evolved in the simulated environment, the last 20 generations in the real environment.

As can be seen, in all conditions performances increase quickly and the decrease in performances after the transfer in the real environment is limited if we look at the peak performances (i.e. the upper curves in each of the three conditions). This means that the gap between the simulated and the real condition is very limited in all cases. It is also interesting to note that such a gap can be reduced by adding noise in the simulated environment. In the *noise* condition the decrease in the peak performance is reduced (even if, on the contrary, average performance decreases more with respect to the *no-noise* condition). Individuals in the *conservative noise* condition reach a bit lower fitness score in the simulated environment with respect to the two other cases with either no noise or sensor noise. Yet, when transferred from the simulated to the real environment at generation 200, individuals treated with *conservative noise* perform quantitatively better than individuals evolved in the *no-noise* or *noise* condition.

Individuals of the *no-noise* condition rapidly recover the performance loss when transferred to the real environment, while individuals of the *noise* condition are able to recover only from the peak performance point of view. Individuals of the *conservative noise* condition do not need to recover at all. Despite of that, they are able to adapt better to the new environment as it is shown by the increase in performances.

To study the statistical significance of the obtained results we made a two factors ANOVA to compare, for each of the three conditions, the performances of the 500 individuals of generation 200 in the simulated environment and in the real environment. The analysis revealed that statistically significant different performances are obtained in the first two conditions but not in the last one (see Table 1). This means that performances of evolved individuals in the *conservative noise* condition do not significantly differ in the simulated and in the real environment.

	degrees of freedom	ratio of Fisher	interval of confidence	Remarks
no noise	1, 998	164.14	.01	significant effect
sensor noise	1, 998	549.79	.01	significant effect
conservative noise	1, 998	0.74	.01	no significant effect

Table 1. Results of two factors analysis of variance, for each experimental condition, between performances of individuals at generation 200 in the simulated and in the real environment.

We also compared, for each of the three conditions, the performance in the real environment of the 500 individuals of generation 200 with performance of individuals of generation 219 (i.e. individual that was trained also in the real environment). The analysis revealed a statistically significant effect in all of the three cases (see Table 2). This means that in all cases, performances increase significantly by continuing the evolutionary process in the real environment.

	degrees of freedom	ratio of Fisher	interval of confidence	Remarks
no noise	1, 998	74.09	.01	significant effect
sensor noise	1, 998	30.74	.01	significant effect
conservative noise	1, 998	26.01	.01	significant effect

Table 2. Results of two factors analysis of variance, for each experimental condition, between performances obtained in the real environment at generation 200 (at beginning of evolution in the real environment) and at generation 219.

All the analyses described above concern performances (fitness scores). In order to verify what happens at the level of behaviors, we compared the trajectories of the best individuals of generation 200 (i.e. individuals evolved in the simulated environment) in the simulated and in the real environment. For each experiment, we tested the best individual by placing it in the same position in the simulated and in the real environment and then let it move for 500 steps. Figures 9, 10, and 11 show, for each of the three conditions and for each of the five experiments, the trajectories observed in the two environments.

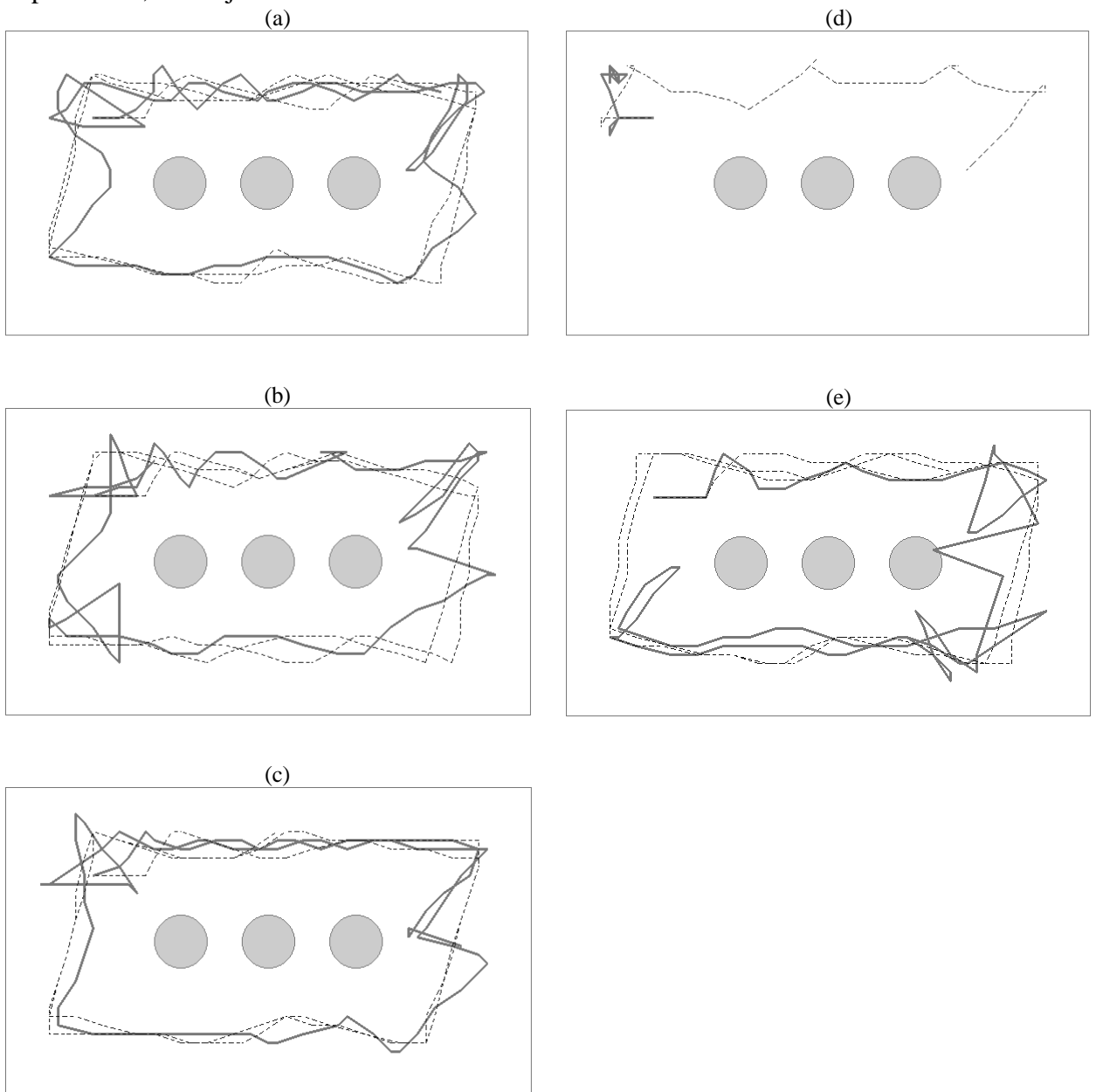


Figure 9. Trajectories of the best individuals of generation 200 of the five *no-noise* experiments in the simulated (dashed line) and real (full line) environment.

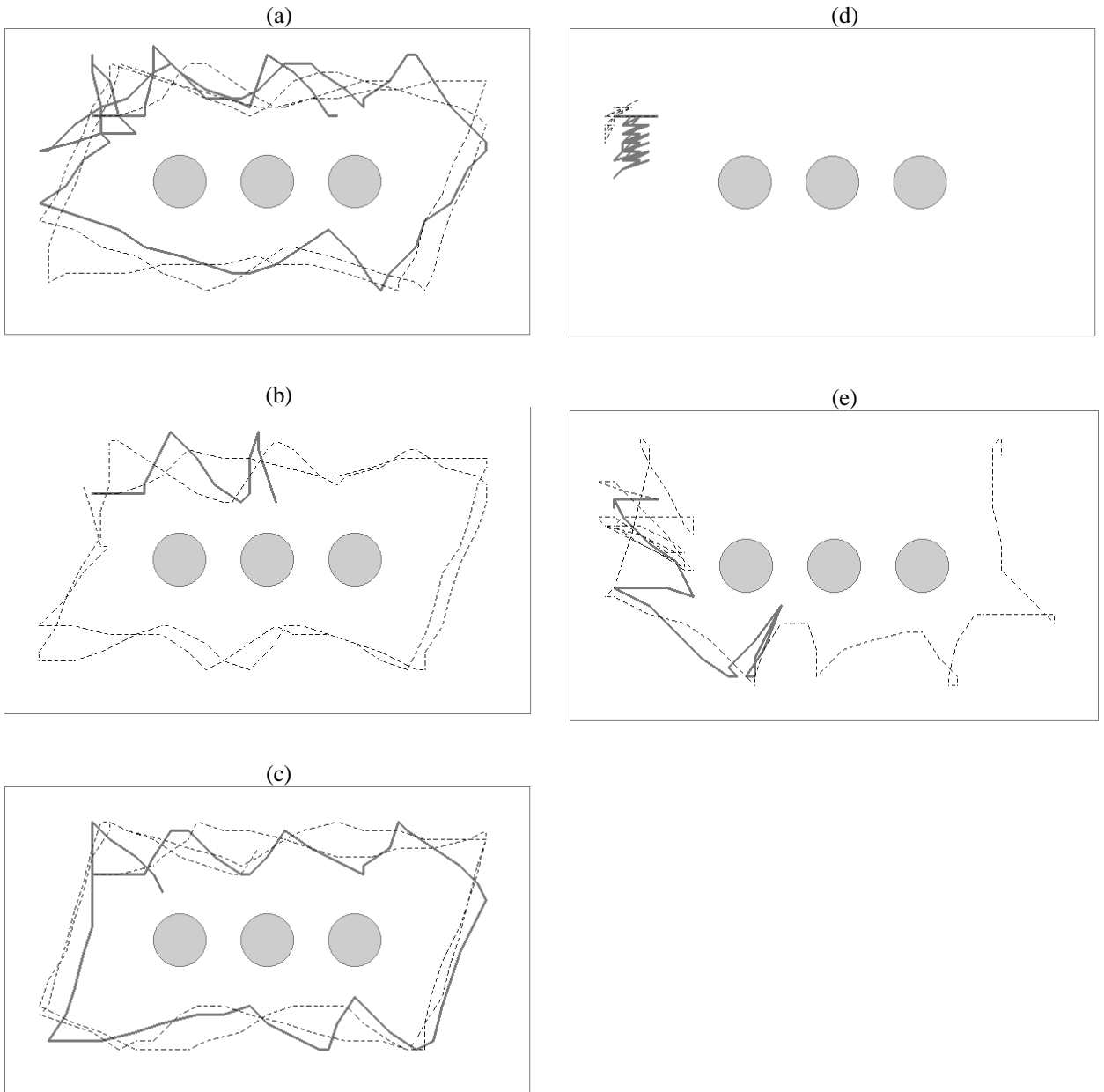


Figure 10. Trajectories of the best individuals of generation 200 of the five *noise* experiments in the simulated (dashed line) and real (full line) environment.

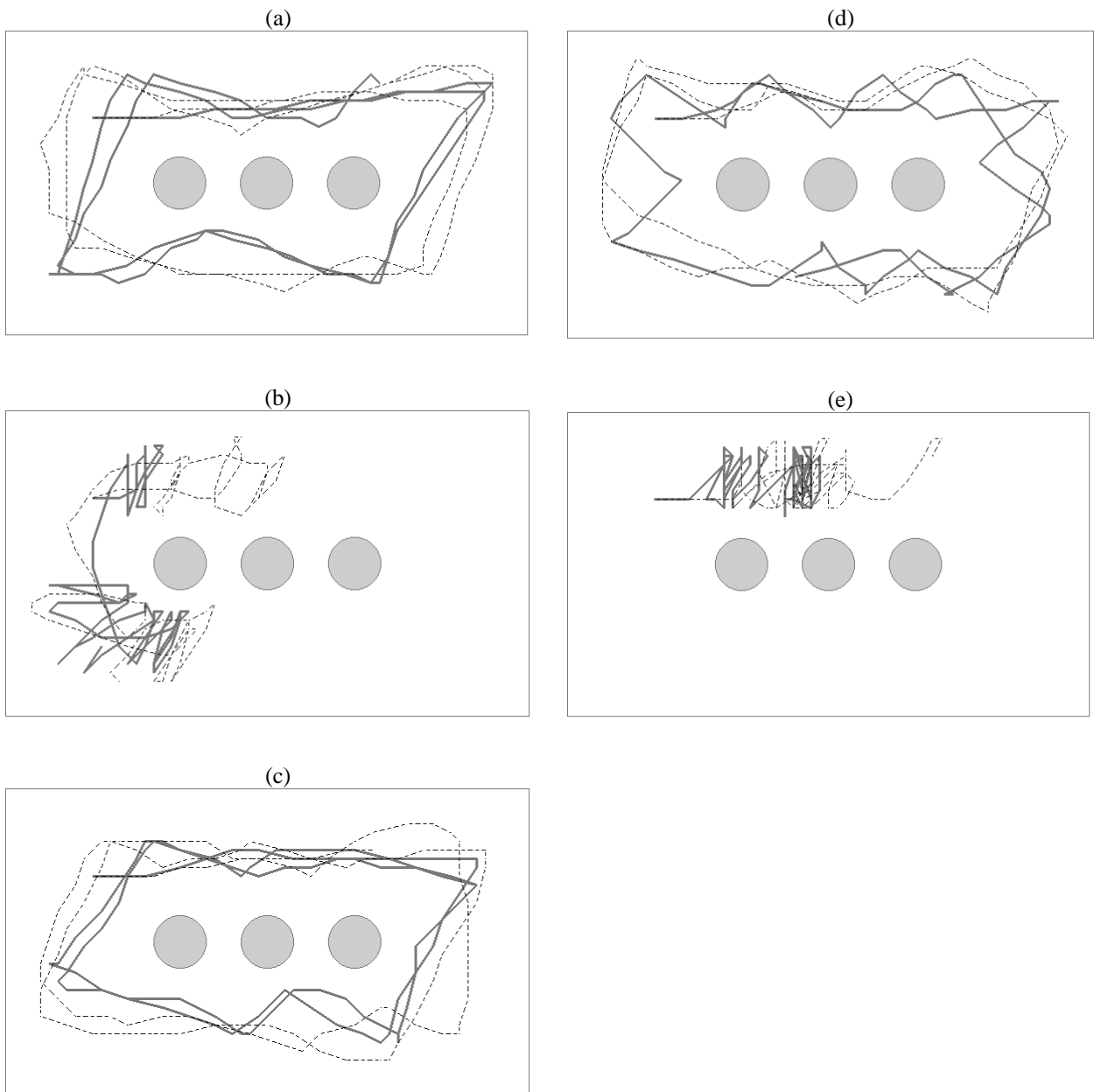


Figure 11. Trajectories of the best individuals of generation 200 of the five *conservative noise* experiments in the simulated (dashed line) and real (full line) environment.

As could be expected from the analysis of the performances, individuals evolved in the *conservative noise* condition are the ones which perform more similarly in the simulated and in the real environments. Trajectories match almost perfectly for individuals evolved in the *conservative noise* condition (see Figure 11) while in the other two conditions in some cases the trajectories are very different (see Figures 9 and 10). What happens is that individuals of the *no-noise* condition, when transferred to the real environment, often turn too much (more

than 90°) in the corners and get stuck there (cf., Figures 10, *a, b, c, e*). In the *noise* condition instead robots move slower in the real environment with respect to the simulated one, and sometimes even hit into obstacles. Robots trained in the *conservative-noise* condition do not show any significant discrepancies in the two environments except in case *d* in which the robot moves a bit slower in the real environment.

To obtain a quantitative measure of the discrepancy of behaviors we calculated the Euclidean distance between the coordinates of the robot in the two environments each 5 steps so that by summing the discrepancies obtained each time step we can have a measure of divergence between two trajectories, $D_{\#steps}$:

$$D_{\#steps} = \sum_{i=1}^{\#steps} \sqrt{(y_j^r - y_j^s)^2 + (x_j^r - x_j^s)^2} \quad (2)$$

where (x_j^r, y_j^r) and (x_j^s, y_j^s) are the positions of the real and simulated robot, respectively, at step j . Figure 12 shows the result of this analysis. As can be seen *conservative noise* appears to reduce very significantly the discrepancy between behaviors in simulated and real environments.

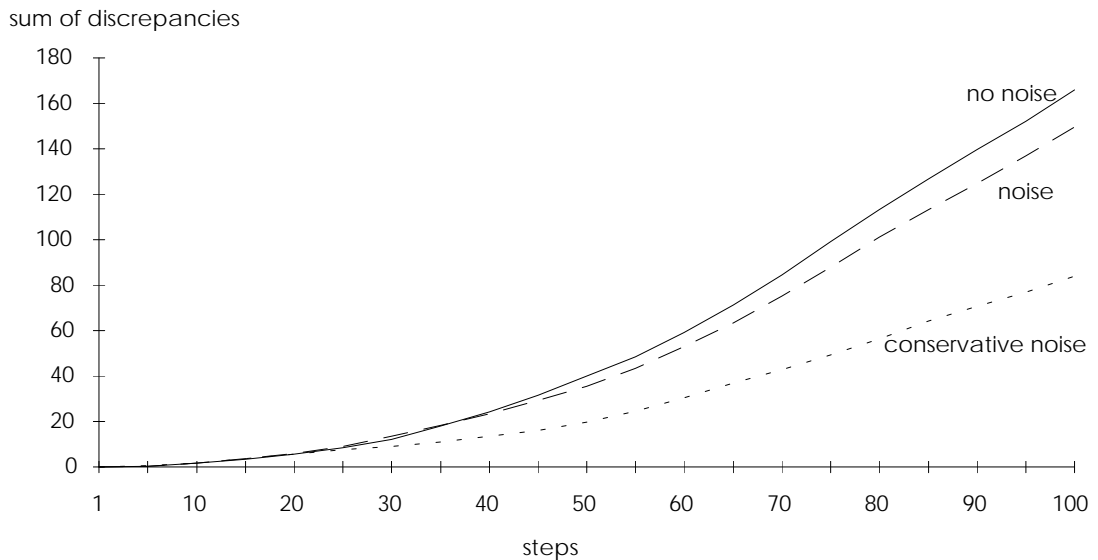


Figure 12. Discrepancy of behaviors in the simulated and real environment for each of the three conditions.

5. Conclusions

We have investigated the validity of computer simulations to develop control systems for autonomous mobile robots. By evolving neural controllers for a Khepera robot in computer simulation and then transferring the obtained control systems to the physical environment, we showed that it is possible to design an accurate enough model of a mobile robot and of its environment.

We claimed that, to accurately simulate a robot, it is very important: (a) to take into account the fact that identical physical sensors and actuators may perform very differently, (b) to account in some way for noise and for other characteristics of the robot and of the environment not included in the simulator (ambient light, slight differences in colour and shape of the objects, etc.). As we showed, problem (a) can be solved by sampling the real world through the sensors and the actuators of the robot. This method, in fact, allows to build a model of a physical individual robot that take into account the differences between robots of the same type and between different identical components of the same robot. The second problem (b) can be faced instead by introducing in the simulator different forms of noise. A conservative form of noise can be used to emulate the characteristics of the robot and of the environment that have not been included in the simulated model and that may produce unpredictable but regular form of distortion between the real and the simulated (and certainly simplified) environment. A standard form of noise can be used to account for the intrinsic noise present in the real environment due to unpredictable causes and to the fact that sensors and actuators return uncertain values.

Our simulative results supported this hypothesis. By comparing performances and behaviors in simulated and real environment we showed that similar results are obtained even without the addition of noise (see Figures 6 and 9). Moreover, we showed that adding noise to the simulated sensors can reduce the gap between simulated and real environment (see Figures 7, 10, and 12). Further, we showed that a regular form of noise (what we called "conservative noise") resulted even more powerful in reducing the gap between simulation and reality (see Figures 8, 11, and 12). In fact, performances of individuals evolved in simulation in

conservative noise condition do not significantly differ in the simulated and in the real environment (see Table 1).

Finally we showed that if a decrease in performance is observed when the system is transferred to the real environment, successful and robust results can be obtained by continuing the evolutionary process in the real environment for few generations. (see Figures 6, 7 and 8). As was showed in Table 2, in all described experiments performances increased significantly by continuing the evolutionary process in the real environment. This result is important because it implies that simulation can be considered a useful tool even if the gap between simulation and reality cannot be completely eliminated. It is not necessary that behaviors in the two environment maps exactly, because by continuing the training process in the real environment perfectly adapted individuals can be obtained.

A general problem of Evolutionary Robotics, that we have avoided to discuss so far, is how to specify an appropriate fitness criterion for a given task. This problem also enters in our experiments with the simulated/physical approach. The fitness criterion for the obstacle-avoidance task (formula 1) seems very obvious with one component for maximizing speed, a second for movement in a straight direction, and a third for the avoidance of obstacles. Yet, this fitness criterion may lead to some deceptive solutions. Look for instance at Figure 11 d. The shown behaviors in both simulated and the real environment seem to be very bad, and they do certainly not fulfill the task. Yet, such behaviors will score relatively high on the fitness measure, since the strategy is to move very fast forward and backward without turning much. The use of the simulated/physical approach to Evolutionary Robotics might make it more easy to overcome such deceptive problems, since it is possible to add special fitness components to the fitness criterion in the simulator, which would not be possible to do in the real environment, as for instance counting number of cells passed by the robot (cf. [14]).

Evolutionary Robotics is an appealing idea arised from the Artificial Life community, but it needs much experimental investigations. This work is a contribute to produce quantitative data and empirical techniques that can be used or criticized by other researchers. We used an obstacle-avoidance navigation task to show the proposed standard techniques usefulness in

the simulated/physical approach regime. When standards have been reached more complex tasks can be investigated. We have here tried to outline some first passes toward such. Comparing the Khepera robot sensors different responses to a physical input, it was clear that it is extremely hard to construct a detailed mathematical model of robot/environment interactions. Therefore, we developed the methodology for building efficient simulators of robots/environment interaction dynamics by letting the robot itself capture the significant physical features of itself and the "world" in which it acts. We think that this simulated/physical approach is the correct methodology for Evolutionary Robotics. According this point of view, the main problem consists to reduce the gap between performances observed in simulated environment and performances obtained in the real environment. From an engineering perspective, we have to observe the same behaviors in both environments. From a more biologically plausible point of view, as that of Evolutionary Robotics, the problem is how to produce artificial genotypes to adapt to different environments (e.g. simulated and real environments). In this perspective, Evolutionary Robotics intersects other interesting research fields in Artificial Life such as phenotypic plasticity [5, 11, 19], pre-adaptations [13, 15] ontogenetic and phylogenetic evolution [1, 19]. In order to understand these phenomena more thoroughly, Evolutionary Robotics might come to play a crucial role in future when standards for the field have been obtained. Such standards will allow more complex experiments to be performed and more complex processes to be understood.

Acknowledgement

The authors would like to thank Domenico Parisi, C.N.R. for useful suggestions and discussions.

References

1. Belew, R. & Mitchel, M. (Eds.) (1995). *Plastic Individual in Evolving Populations*. SFI series, Addison-Wesley.

2. Brooks, R. A (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2 (1).
3. Brooks, R. A. (1992). Artificial life and real robots. In F. J. Varela, P. Bourguine (Eds.), *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. Cambridge, MA: MIT Press/Bradford Books.
4. Cliff, D. T., Harvey, I., & Husbands, P. (1993). Explorations in Evolutionary Robotics. *Adaptive Behavior*, 2, 73-110.
5. Dellaert, F & Beer, R. D. (1994). Toward an evolvable model of development for autonomous agent synthesis. In R. Brooks and P. Maes (Eds.) *Proceedings of fourth International Conference on Artificial Life*. Cambridge, MA: MIT Press
6. Dorigo, M., & Schnepf, U. (1993). Genetic-based machine learning and behavior based robotics: a new synthesis. *IEEE Transaction on Systems, Man and Cybernetics*, 23,141,153.
7. Floreano D., Mondada, F. (1994). Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot. In D. Cliff, P. Husbands, J. Meyer, S. W. Wilson (Eds.) *From Animals to Animats 3: Proceedings of Third Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press/Bradford Books.
8. Harvey, I. (1992). Species adaptations algorithms: A basis for a continuing SAGA. In F. Varela, P. Bourguine (Eds.) *Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life (ECAL91)*. Cambridge, Ma: MIT Press/Bradford Books.

9. Harvey, I., Husbands, P., & Cliff., D. (1994). Seeing the light: Artificial evolution, real vision. In D. Cliff, P. Husband, J-A Meyer, and S. Wilson (Eds.), *From Animals to Animats 3, Proceedings of 3rd International Conference on Simulation of Adaptive Behavior*. Cambridge, Ma: MIT Press/Bradford Books.
10. Jakobi, N., Husbands, P., & Harvey, I. (in press). Noise and The Reality Gap: The Use of Simulation in Evolutionary Robotics. To appear in *Proceedings of Third European Conference on Artificial Life (ECAL'95)*. Springer-Verlag.
11. Kodjabachian, J. & Meyer, J. A. (1994). Development, Learning and Evolution in Animats. In: D. P. Gaussier and J-D. Nicoud (Eds.) *Proceedings of the Intl. Conf. From Perception to Action*, Los Alamitos, CA: IEEE Press.
12. Koza J. R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge MA: MIT Press/Bradford Books.
13. Lund H. H., D. Parisi. (in press). Pre-adaptations in Populations of Neural Networks Living in a Changing Environment. To appear in *Artificial Life*.
14. Miglino O., Nafasi, K., & Taylor, C. (1994). Selection for Wandering Behavior in a Small Robot. *Technical Report*. UCLA-CRSP-94-01. Department of Cognitive Science, University of California, Los Angeles. To appear in *Artificial Life*.
15. Miglino, O., Nolfi S. & Parisi, D. (1995). Discontinuity in evolution: how different levels of organization imply pre-adaptation. In Belew, R. & Mitchel, M. (Eds.) *Plastic Individual in Evolving Populations*. SFI series, Addison-Wesley.

16. Mondada F., Franzi, E., & Ienne, P. (1993). Mobile Robot miniaturisation: A tool for investigation in control algorithms. In: *Proceedings of the third International Symposium on Experimental Robotics*, Kyoto, Japan.
17. Nolfi, S., Floreano, D., Miglino, O., & Mondada, F. (1994). How to evolve autonomuos robots: different approaches in evolutionary robotics. In R. Brooks and P. Maes (Eds.) *Proceedings of fourth International Conference on Artificial Life*. Cambridge, MA. MIT Press
18. Nolfi, S., Miglino, O., & Parisi, D. (1994). Phenotypic Plasticity in Evolving Neural Networks. In: D. P. Gaussier and J-D. Nicoud (Eds.) *Proceedings of the Intl. Conf. From Perception to Action*. Los Alamitos, CA: IEEE Press
19. Nolfi, S., & Parisi, D., (1993). Phylogenetic Recapitulation in the Ontogeny of Artificial Neural Networks. *Technical Report 93-01*, Institute of Psychology, C.N.R., Rome.
20. Nolfi, S., & Parisi, D. (1995). Evolving non-trivial behaviors on real robots: An autonomous robot that pick up objects. *Technical Report 95-03*, Institute of Psychology, C.N.R., Rome.
21. Yamauchi B., & Beer, R. (1994). Integrating reactive, sequential, and learning behavior using dinamical neural networks. In D. Cliff, P. Husband, J-A Meyer, and S. Wilson (Eds.), *From Animals to Animats 3, Proceedings of third International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press/Bradford Books.

The corrispondnce should be addressed to: Orazio Miglino
Institute of Psychology, National Research Council
Viale Marx 15
00137 Rome, Italy.
e-mail: orazio@caio.irmkant.rm.cnr.it
voice: 0039-6-86090227
fax: 0039-6-824737