Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot

Dario Floreano
Laboratory of Cognitive Technology
AREA Science Park
Trieste, Italy
dario@psicosun.univ.trieste.it

Francesco Mondada
Laboratory of Microcomputing
Swiss Federal Institute of Technology
Lausanne, Switzerland
mondada@di.epfl.ch

Abstract

The paper describes the results of the evolutionary development of a real, neural-network driven mobile robot. The evolutionary approach to the development of neural controllers for autonomous agents has been successfully used by many researchers, but most -if not all- studies have been carried out with computer simulations. Instead, in this research the whole evolutionary process takes places *entirely* on a real robot without human intervention. Although the experiments described here tackle a simple task of navigation and obstacle avoidance, we show a number of emergent phenomena that are characteristic of autonomous agents. The neural controllers of the evolved best individuals display a full exploitation of non-linear and recurrent connections that make them more efficient than analogous man-designed agents. In order to fully understand and describe the robot behavior, we have also employed quantitative ethological tools [13], and showed that the adaptation dynamics conform to predictions made for animals.

1 Introduction

A mechanical device that can operate without being attached to a power supply or an external computer is not necessarily an autonomous robot. Although this may be an additional desirable feature, autonomous robots are rather identified by their ability to adapt to an environment by finding optimal solutions, develop a suitable control system, define their own goals, and, possibly, perform some self-monitoring [19]. All these capabilities cannot be pre-defined, but should rather emerge from the interaction between the robot and its own environment. A possible solution for building autonomous systems consists in using simple components and primitive structures for the control system; in this case, articulated and complex behaviors would be the spontaneous result of the interactions among all these parts through a process of self-organization guided by a continuous exchange of information with the environment. Major steps in this direction have already been taken. Brooks's subsumption architecture [4] is indeed a case of constructive, bottom-up approach toward building autonomous robots that display emergent behaviors. His approach consists of providing the robot with a set of simple behaviors; further behavior-modules can be added on the top of these primitives and connected to them via simple excitatory or inhibitory links. A similar approach has been formulated by Steels [19], who is pursuing the goal of building intelligent agents by focusing on action-centered skills, autonomy, behavior-oriented decomposition, emergent functionality, and layered architectures. In a more general context, Maes [11] has tried to define the theory, methodology, and goals of a new Behavior-Based Artificial Intelligence, as contrasted to the Knowledge-Based Artificial Intelligence. Beside these solutions, some other researchers have fulfilled the requirements of learning and adaptation by employing various sorts of neural networks to control a robotic system [2], [20]; whether pre-wired or plastic, these neural controllers exhibit characteristics of generalization, flexibility, robustness, and, possibly, plastic adaptation. All these features are indeed important prerequisites of autonomous agents. A somehow different step toward design automatization of autonomous robots is taken by those researchers that try to evolve the robot control system. Rather than starting from a designed solution, they describe the primitives of the robot in the form of an artificial chromosome, build many of these chromosome with some random arrangement of the genes, test the control system generated with every chromosome on a robot, select and reproduce only those chromosomes that guarantee the robot a better fitness according to some survival criterion; this process is repeated until the average population performance is good enough or some mutant with exceptional characteristics is born. Although the evolutionary procedure [9], [7] is well known to a vast community of researchers, it is not a straightforward task to apply it to real robots, as we will see later. Our work concerns the evolution of a neural-network-controlled mobile robot. What is really important in our experiments is that the whole evolutionary process takes places *entirely* on a real robot without human intervention. Before going into the description of our results and the following discussion, let us stress two points that we think to be of general relevance, namely the choice of a neural architecture and the role of simulations versus real implementations.

2 Neural Architectures

Artificial neural networks seem to us to be particularly good candidates for the control system of artificial autonomous agents because they possess many desirable features required by the principles of autonomy in real environments (see also [8]). Let us list some of these properties.

- Neural networks are flexible. The ability to learn enables dynamic adaptation of robot behavior to changes in the environment. Even when the synapses are not modifiable, a neural network still exhibits a reasonable degree of flexibility, i.e., it is able to produce appropriate behaviors in response to a range of possible variations of the physical stimulation.
- Artificial neural networks are robust: missing links or malfunctioning of some hardware components do not strongly impair the robot's behavior.
- A neural network deals with the micro-structure of the robot: this means that it can either shape its own structure to exploit at its best the sensory-motor features of the robot [5], or actively select and use only those sensors and motors that are best suited for performing the task [16].
- The well known tolerance to noise (in some cases noise enhances performance [17], or is an essential component for learning, such as in self-organizing neural networks) makes them good candidates for mediating between physical sensors and actuators with intrinsic noise.
- If we do not put limits to the network architecture, and thus have recurrent and lateral connections, and non-linear transfer functions, we have a potentially powerful device that could cope with the temporal structure and complex mappings required by real-world operations.

Finally, artificial neural networks are well-suited structures also for artificial evolution. Small changes in a neural network usually correspond to small changes in its behavior, at least for feed-forward architectures. Genetic algorithms find their way toward a maximum by sampling new solutions obtained by random combinations and mutations, and thus take advantage of the intrinsic "gradualism" of the neural network structure.

3 Simulation versus Implementation

There is currently a hot debate among people trying to understand and reproduce intelligent agents, that could be stated as follows: "Is the simulation a powerful enough tool to draw sound conclusions, or should a theory or an approach be tested on a real agent, i.e., a robot¿' Although both numerical simulations and physical implementations have their own merits in different fields of research, the issue becomes important when we investigate autonomous and intelligent agents. Let us examine in more detail the respective advantages and drawbacks of these two methodologies in our particular case. It is usually argued that computer simulations are fast. High performance serial machines and massively parallel computers nowadays are powerful tools for the virtual reproduction and analysis of complex-system dynamics. In a few days of computation the scientist can reproduce birth and death of whole populations of organisms (see, e.g., [1]). But this holds only to a limited level of sophistication. It is still much faster to have a real camera acquiring images from a real world than simulating the world, the camera, and the image acquisition process (see [8]). This is not a problem of "bottlenecks", but it is due to the fact that enormous calculations are sometimes necessary for simulating a very trivial [1] physical phenomenon, partly because computers are general-purpose machines whereas natural devices are "dedicated hardware". Another common belief is that computer simulations are cheaper. The researcher may think that it is worth exploring a hypothesis or a new algorithm by computer simulations before investing money and time in a robot. Although this may be true in many situations, in some cases it is not. It all depends on the degree of plausibility and "reality" of the simulation. If the standard is intended to be high, then it is very likely that it will involve one or more specialized programmers on the project for many months. Sometimes, this may cost more than building, purchasing, or modifying real robots. It is widely accepted that numerical simulations allow complete control and record of all the variables; it is thus possible to replicate results, analyze phenomena, accelerate or slow down processes. This is certainly true. But why should we have complete control over an autonomous agent? After all, an artificial agent will never be truly autonomous while there is an umbilical chord that limits its field of action Autonomous agents living within a computer are limited by the necessarily-predefined number of experiences and levels of interactions with the environment. If we are to build intelligent autonomous agents, then we will have to

[1] Here "trivial" is meant in a naive sense, as opposed to number of processes or complexity of the dynamics involved

give up -sooner or later- with the obsession of controlling and replicating every possible accident. Computer simulations are still a powerful tool for our research. They provide a viable solution for experimenting non-existing devices or non feasible (with physical tools) hypotheses about the nature and characteristics of artificial agents. "Life as it could be" is indeed one of the major topics of Artificial Life [10], a field of investigation that much inspires research in autonomous agents. Computers leave us free to use our imagination and test the most bizarre hypotheses to recreate new autonomous organisms living and behaving within worlds with different physical laws. However, when we simulate something we must always be aware that we are putting some constraint somewhere at some level. It is not anymore the real world that we are dealing with. And this may be a crucial point when trying to create an autonomous system. By definition, an autonomous agent itself will define the level of interaction with its own environment and alone will choose the relevant information to take into consideration. If we are to restrict at some point the range of available possibilities, we may hamper or greatly reduce the potentiality of our agent. One of the strongest critics made against the simulative approach is that numerical simulations do not consider all the physical laws of the interaction of a real agent with its own environment, such as mass, weight, friction, inertia, etc. Although this may be questionable, it is certainly true that simulations do not take into account Murphy's Laws, such as malfunctioning, component failures, and consumption that govern both artificial and biological organisms. Finally, a real danger with computer simulations is that it cannot be guaranteed that a transfer to the implementation phase will be smooth, if feasible at all. But, let us imagine this to be possible. Who will guarantee to us, then, that the robot is actually doing what it was doing in the simulations? How to compare precise numerical values with behavioral data collected in a noisy world? This is especially important for those researchers who develop the control system of the robot with a computer simulation, and then "inject" the resulting "brain" into the processor of their physical agent and leave it free to move. The analysis and discussion of the reasons why one method should be preferred over the other may take much longer; here, we have only tried to outline a few important topics that we felt relevant for our methodology.

3.1 Evolutionary Development of a Physical Robot

But, for what concerns our specific research, there is a more compelling question. Why are several groups working on simulations, but it is hard -or even impossible- to find cases of generational development of populations of real robots, that is, robots that must survive in a real world on the basis of some fitness criterion, where only the fittest can mate and reproduce through a generational and cyclic process? We believe that the reason is not the cost and waste of material (not fitted robots), or difficulties with the mating procedure, but it is rather based on the construction principles of robots. Most of the available robots are not suited for evolution, in terms of mechanical robustness, design concepts, and automatic evaluation of the robot performance:

- Evolution (Genetic Algorithms) takes a long time; it may require hours, days, weeks, or even months, of continuous functioning of the hardware. Most of available robots tend to break down in these conditions and are not capable of self-repair, as biological organisms often do.
- The common philosophy underlying the construction of robots designed for operating in autonomy dictates that many precise and sophisticated sensory devices should be mounted on the main board. The mechanical solutions for moving around and performing other actions are taken either from well established engineering solutions (three wheel synchronous drive, for instance), or from successful biological organisms (stick insects, ants, etc.). This leads to the construction of complex, highly structured, and fragile mechanisms. For this reason, such robots would easily get trapped in corners and local minima during the first generations of the evolutionary process. Whereas there is no reason in principle why the evolutionary technique should not be applicable to complex robots (and indeed it will have to, at least to some extent), it is definitely true that biological evolution did not start with a structured and sophisticated body coupled with a virtually non-existing brain. Evolutionary studies have shown that there is a gradual coevolution of body and "mind" in biological organisms. Thus, either we start with a robot designed with new principles (simple components and geometry, robust and reliable hardware, only necessary and elementary sensors and actuators), or we provide a complex robot with a set of "basic instincts" (but which?) and let evolution work on higher control structures. We have chosen the first approach because we consider it to be chronologically and logically the first thing to try, and also because the second solution, although viable in principle, may still be problematic at this stage.
- In order to get a sensible behavior out of a *tabula rasa* (whatever type of architecture we use), Genetic Algorithms require a fitness function, i.e., a survival criterion against which each individual of the population is tested. As long as the artificial agent is a virtual entity within a computer it is fairly easy to precisely evaluate its performance. However, when the agent takes form into a physical and mobile body

free to wander in our world, automatic fitness evaluation becomes a non-trivial task. We will take into consideration this issue in a later section.

4 Navigation and Obstacle Avoidance

Because of all the reasons outlined in the section above, we were not certain that the evolutionary approach would have worked with a real robot. Mainly, we did not know how to assess and compare the results that we could have obtained with this approach. Thus, we have decided to start from a classic task, a sort of exercise and test for all people working with mobile robots. The robot had to learn to move in an environment and avoid obstacles. For its simplest formulation, there is already a well-known, optimal, and simple distributed solution for this task: the Braitenberg's vehicle [3], with which we have compared our results. Thus, we have put our robot in an arbitrary environment, set a few parameters concerning the fitness function and the network structure, and let it free to evolve. We have run this experiment many times in order to obtain reliable data and draw sound conclusions. Each time, we have kept track of a some relevant variables during the evolutionary process, analyzed the best organisms, and compared the solutions obtained by evolution with those designed by man.

4.1 Experimental Setup

The robot employed in our experiments is Khepera, a miniature mobile robot [14]. Khepera has many of the characteristics required by the evolutionary approach to autonomous robots. It has a circular shape (Figure 1), with diameter of 55 mm, height of 30 mm, and weight of 70 g, and is supported by two wheels and two small Teflon balls.

The wheels are controlled by two DC motors with incremental encoder (10 pulses per mm of advancement of the robot), and can move in both directions. The simple

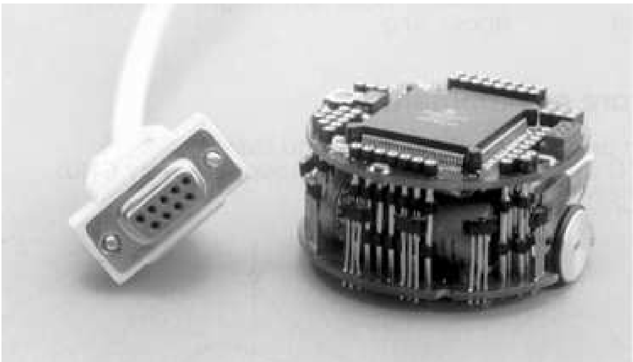

Figure 1: Khepera, the miniature mobile robot.

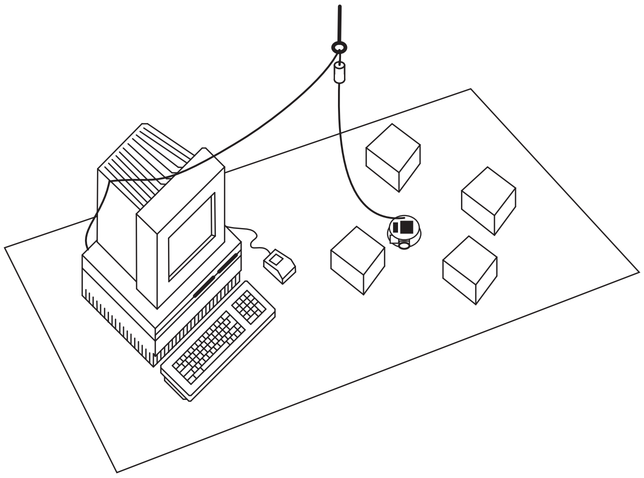

Figure 2: Operating methodology.

geometrical shape and the motor layout allow Khepera to potentially negotiate any type of obstacle and corner. These characteristics, together with many other mechanical solutions, have resulted in a robot that has continuously and reliably operated for weeks and weeks, most of the time crashing into walls and obstacles (due to the functioning principles of Genetic Algorithms). In the basic configuration used here, the robot is provided with eight Infra-Red proximity sensors. Six sensors are positioned on one side of the robot (front), the remaining two on the other side (back). A Motorola 68331 controller with 256 Kbytes of RAM and 512 Kbytes ROM manages all the input-output routines and can communicate via a serial port with a host computer.

Because of its size and design principles, Khepera is well-suited for laboratory experiments. Its communication protocol can exploit all the power and disk size available in a workstation by letting high-level control processes (genetic operators, neural network activation, variables recordings) run on the main station while low-level processes (sensor-reading, motor control, and other real time tasks) run on the on-board processor (Figure 2).

We have adopted this solution for our experiments. Khepera was attached via a serial port to a Sun SparcStation 2 by means of a lightweight aerial cable and specially designed rotating contacts. In this way, while the robot was running, we could keep track of all the populations of organisms that were born, tested, and passed to the genetic operators, together with their "personal life files". At the same time, we could also take advantage of specific software designed for graphic visualization of trajectories and sensory-motor status while the robot was evolving. Skeptics should not consider this methodology as an attempt on the very hearth of autonomy: as stated in the very beginning of this paper, running with

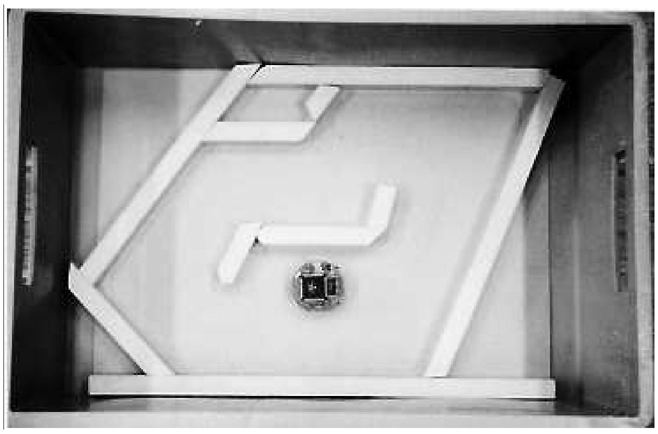

Figure 3: Environment of the experiment.

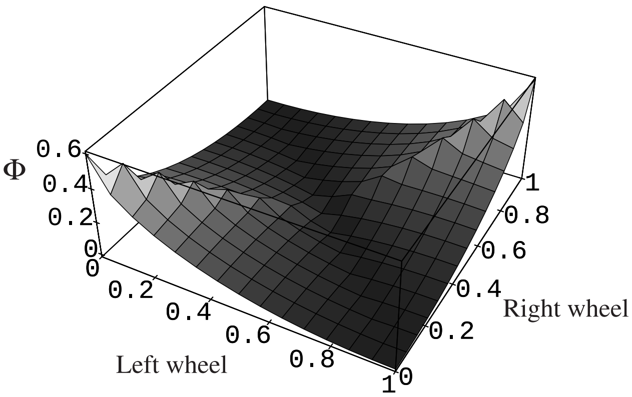

Figure 4: Function surface for i = 0.4. Wheel speed values have already been transformed into positive range where 0.5 is the point of direction inversion. Please note that this is not a full picture of the fitness function maximized by the genetic algorithm, which is instead n-dimensional ($n =$ number of neural network free parameters). Furthermore, it does not take into account the physical characteristics of the environment.

its own batteries is only an optional feature of an autonomous agent. For what concerns Khepera, the robot is not aware of where its own "brain" is located, and this is indeed not important in this experiment of navigation and obstacle avoidance. However, it should be noted that the software that implements the genetic development of neural networks [6] could be slimmed down and downloaded into the robot processor.

The robot was put in an environment consisting in a sort of circular corridor whose external size was approx. 80x50 cm large (Figure 3). The walls were made of light-blue polystyrene and the floor was 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 a room always illuminated from above by a 60-watt bulb light. A serial cable connected the robot to the workstation in our office, a few rooms away from it. Our goal was to develop a robot that could learn to maximize some sort of exploration measure while accurately avoiding all the obstacles on its way. This statement was also the base for the fitness criterion used in the experiments. One of the desirable features of autonomous robots is the independence from an external operator, also during the development process of the control system. This would mean that the performance criterion for an autonomous agent should rely solely on a set of variables that can be measured within the frame of interaction between the robot and the environment. If this constraint is satisfied, we achieve a practical advantage, because the robot could eventually learn to operate in any environment by a continuous self-assessment of its own performance without external controllers. Hence, our fitness criterion Φ was function of three variables, directly measured on the robot, as follows,

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

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

where V is a measure of the average rotation speed of the two wheels, Δv is the algebraic difference between the signed speed values of the wheels (positive is one direction, negative the other) transformed into positive values, and i is the activation value of the proximity sensor with the highest activity. 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 motion direction (Figure 4).

The evolutionary training was a standard genetic algorithm as described by Goldberg [7] with fitness scaling and roulette wheel selection, biased mutations [15], and one-point crossover (experiment parameters are given in the Appendix). The neural network architecture was fixed and consisted of a single layer of synaptic weights from eight input units (clamped to the sensors) to two output units (directly connected to the motors)

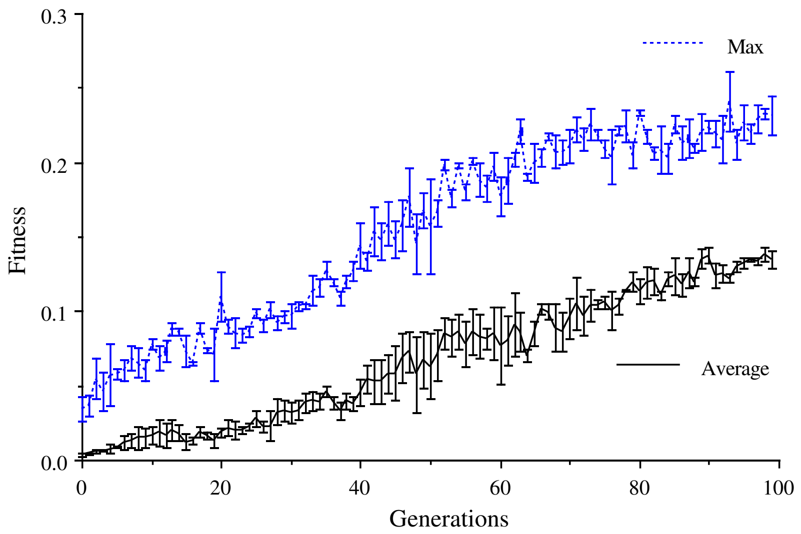

Figure 5: Population average fitness and best individual fitness at each generation. Values are averaged over three runs (S.E. displayed).

with mobile thresholds, logistic activation functions, and discrete-time recurrent connections only within the output layer. Given the small network size, each synaptic connection and each threshold was coded as a floating point number on the chromosome [21]. Each individual of a population was in turn decoded into the corresponding neural networks, the input nodes connected to the robot sensors, the output nodes to the motors, and the robot was left free to move for a given number of steps (motor actions) while its performance Φ was automatically recorded. Each motor action lasted 300 ms. Between one individual and the next, a pair of random velocities was applied to the wheels for 5 seconds. This procedure was aimed at limiting the artifactual inheritance of particular locations between adjacent individuals in the populations.

4.2 Results

Khepera genetically learns to navigate and avoid obstacles in less than 100 generations (Figure 5), each generation taking approximatively 39 minutes. However, around the 50th generation the best individuals already exhibit a near to optimal behavior. Their navigation is extremely smooth, they never bump into walls and corners, and try to keep a straight trajectory. This allows them to perform complete laps of the corridor without turning back. These results are highly reliable and have been replicated in many runs of the experiment.

It is interesting to analyze a single run of the evolutionary development of Khepera by looking at the values of the three fitness components for the best individuals in the population at each generation (Figure 6).

During the initial generations the best individuals are those that move straight at very low velocities (about 2 mm/s). High oscillations of the sensory component indicates that they cannot yet discriminate between walls and empty spaces: it is still much up to individual "luck" (starting location) to avoid crashing into an obstacle. Most of the remaining individuals in the initial generations spend their life by rotating in place. Maximizing the fitness function Φ means to find a balance among the three components because none of them can assume the maximum value without lowering one of the other two. A stable balance is found around the 50th generation. In the remaining 50 generations the robot increases only the global motion speed. However, the global speed never reaches the maximum value (80 mm/s), not even when the evolutionary process is continued until the 200th generation. For all the best individuals, the robot speed peaks at 48 mm/s when positioned in zones free of obstacles. This self-adjustment of the maximum cruising speed has an adaptive meaning. Since sensors and motors are updated only every 300 ms and many passages in the environment are rather narrow, if Khepera had moved faster it would have often crashed into walls without the possibility to detect them. Thus, the system has adapted its own behavior to the physical characteristics of its own sensory system and of the environment where it lives. We have tested some of the best individuals of the last generations in new environments with a variety of objects (differing in shape, color, texture, and light absorbency) and new light conditions (full sunlight, new rooms with different artificial light). We have

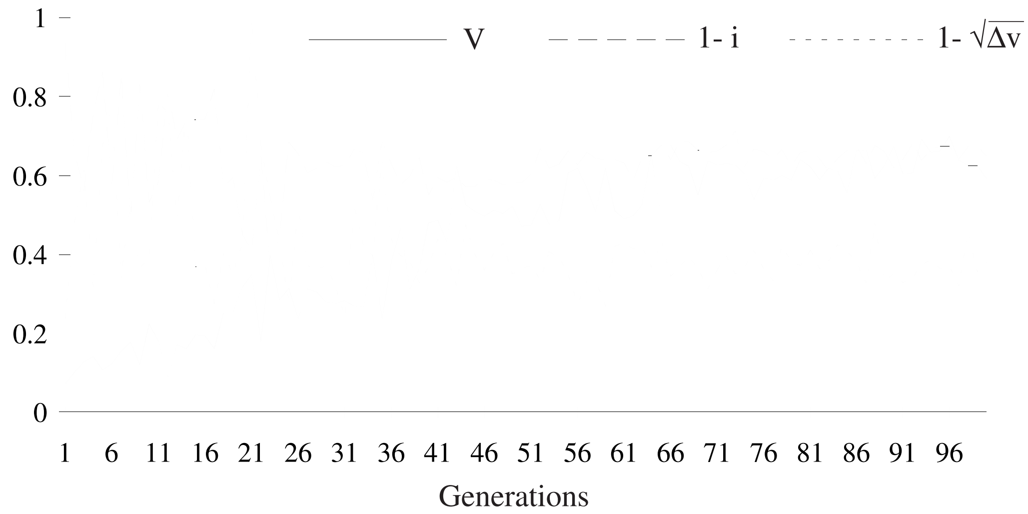

Figure 6: Fitness components values for the best individual of each generation. V is maximized by robot velocity; $1 - i$ is maximized by obstacle avoidance; $1 - \sqrt{\Delta v}$ is maximized by straight trajectory.

also tested the best individuals with other robot bodies (same model, but obviously with slight variations of the sensor responses). In all these cases Khepera navigates successfully without touching any of the objects and trying to keep a straight trajectory. All the individuals tested show a preferential turning direction which solely depends on the initial conditions of the evolutionary run (initial weight values, interaction with the environment), but they can turn in both directions when required by the environment.

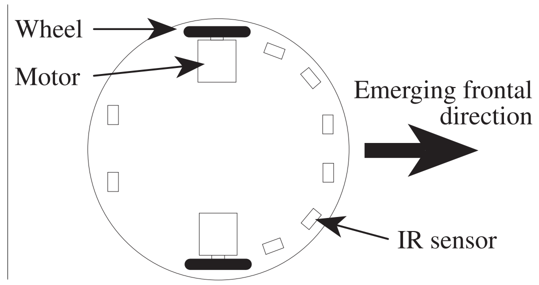

Figure 7: IR sensors and motors layout in Khepera. Diameter size is 55 mm, maximum speed in either direction is 80 mm/s.

4.3 Discussion

A basic characteristic of autonomous systems is the ability to self-regulate their own behavior in order to maximize the probability of survival and reproduction. In this sense adaptation is function of the interaction between two variables, the physical properties of the environment and the characteristics of the organism's body. The success of any plan, strategy, or single action, depends not only on the affordances of the environment, but also on the capacity to detect them and adequately respond. In nature we can observe a continuous evolutionary co-adaptation of body structures and behavioral repertoire. Although we cannot yet expect changes in the hardware structure of an autonomous robot, still we should observe self-selection of the behavioral strategies that exploit at best the physical features of the robot's body and sensory-motor apparatus. We have already seen an example of such a behavioral adaptation in the case of the speed self-regulation of our robot. Another significative example of autonomous adaptation is given by the direction of motion.

Khepera has a perfectly circular and symmetric body shape and the wheels can rotate at equal speeds in both directions. In terms of pure kinematics, thus, it is logical to expect that the robot will equally move in either direction, depending on initial internal and external conditions. However, in all our experiments, early during the evolution the robots develop a frontal direction of motion that corresponds to the side with more sensors (Figure 7). The development of this frontal direction of motion allows the robot to face obstacles with the side that provides a finer resolution and a larger visual angle. Those individuals that move "backward" are very likely to get stuck in convex and sharp corners or fail to detect a lateral collision with a wall; hence, they disappear very soon from the population. (Analogous phenomena of behavioral adaptation to the visual configuration of a simple simulated organism have been shown by [5].) However, rear sensors do not go out of use. The neural networks of the best individuals of the final generations still make use of that information to change trajectory if

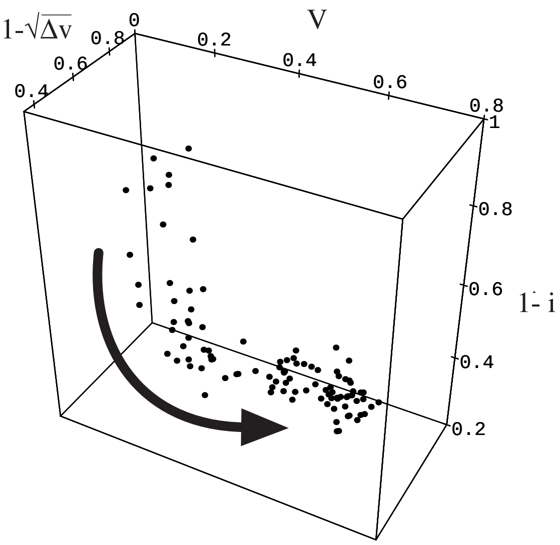

Figure 8: State-space representation of the adaptation process. Each dot is the state of the best individual of a generation. The arrow shows the direction of motion during evolution. The dots concentrate in a sub-space indicated by the arrow tip in the last 20 generations. Axes range spans from 0 to 1 (only covered space is shown in the picture).

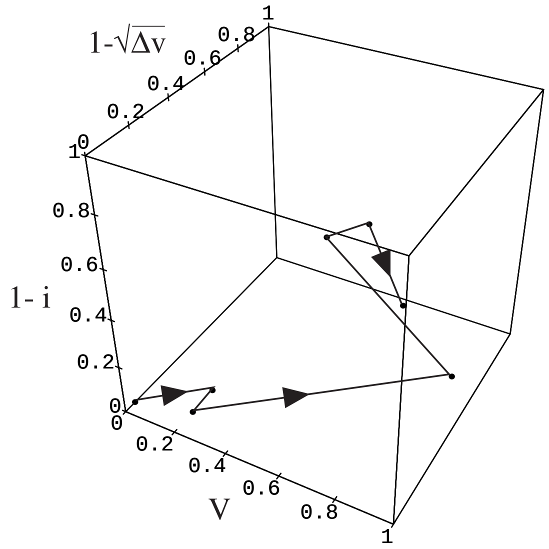

Figure 9: State-space representation of the behavior of the best individual of the last generation, when pulled apart from its equilibrium state.

something is approaching the robot from the back. As for any dynamic system, also in the case of evolved robots it is important to understand and try to describe the state-transition phase. But an autonomous system is not completely controllable and observable [12]. This holds also for our robot, both because the dynamics and results of the evolutionary process cannot be controlled, and because the inner functioning of the neural network, as we will see later, is not linear and each state depends upon a previous history of states. However, as in the case of animals, the activity of an autonomous agent depends on the state of the agent itself, such as its level of energy, the perception of the environment, and the memory of previous states. This analysis yields to the construction of an n-dimensional state space, where the axes are provided by n state variables considered. This "state-space approach" has been used in ethology [13] to describe animal behavior in quantitative terms, and can be applied also to our agent. We can describe our agent as a point in a three dimensional space given by the values of the three fitness components and monitor its change in time.

Figure 8 is a state-space plot of the best individuals of each generation during evolution. The adaptation process is described by a reduction of oscillations and by a gradual displacement toward a sub-region of this space. This region of the adaptation space is compact and bounded, and represents the stability conditions of the system [12] that satisfy the survival criterion.

Our system is asymptotically stable because, when disturbed (by the recombination and mutation operators during the last 20 generations), it tends to stay within the same adaptation zone. This holds also when we analyze the behavior of a single individual. If we disturb the system by pulling it away from its equilibrium state, it will tend to return and stay in its original state (Figure 9). This analysis may be carried further on along the lines of the "Adaptation Theorem" of Sibly and McFarland [18], but there is not space enough here (we will consider this issue in further detail with more complex examples of evolved behaviors in another paper). A final consideration is deserved by the comparison between our agent and a distributed Braitenberg's vehicle designed to go straight and avoid obstacles. Braitenberg's vehicle (which has been implemented on Khepera too) is a linear reactive system that, basically, when some sensors are activated, gives more energy to the ipsilateral motor and inhibits the controlateral one. The pattern of synaptic connections is symmetrical about the front axis. This very simple system is very efficient, but is gets stuck as soon as two symmetric and controlateral sensors become equally activated. In this case the total amount of energy given to each motor is equal and tends to 0. Instead, our agent has developed a pattern of synaptic connections similar to Braitenberg's vehicle, but it has also accurately exploited the recurrent connections at the output layer and the non-linearities embedded in the activation functions. The best individuals of the last generations never get stuck in such cases because the state of the motors is not uniquely defined by the current state of the sensors, but also by the previous history of actions.

5 Conclusion

We have described and analyzed a working example of an artificial autonomous agent. Our robot satisfies most of the basic criteria that underlie the definition of autonomous agents. Through the evolutionary process Khepera has automatically and autonomously developed the optimal distributed control system to survive in the environment where it has been placed. The role of the human experimenter has been indeed rather small, specifically limited to formulate only the survival criterion and the global structure of the net. We have neither pre-designed the behaviors of the robot, nor have intervened during evolution. The robot itself and alone has developed -starting from a sort of *tabula rasa* - a set of strategies and behaviors as a result of the adaptation to the environment and its own body. Despite its simple components and the simple survival criterion, it is difficult to control and predict the robot behavior, due to the non-linearities and feedback connections exploited for optimal navigation and obstacle avoidance. We have tried to describe our agent's behavior with quantitative ethological tools, and we have also showed two emergent phenomena such as speed self-regulation and frontal direction. Our current work is aimed at using the same approach in more complex environments where the fitness criterion is not anymore fixed by the experimenter, but is the natural and logical result of the interaction between the physical characteristics of the robot and the type of environment. We have already obtained new significant results where homing for battery recharge is purely an emergent behavior. These data make us confident in thinking that our approach is a valid methodology for automatically creating complex autonomous agents. Future work will enable evolvability and more flexibility (through a major adherence to biological plausibility) in the neural network structure and will employ learning during life as well.

Appendix

Genetic algorithm parameters:

Population size	80
Generation number	100
Crossover probability	0.1
Mutation probability	0.2
Mutation range	± 0.5
Initial weight range	± 0.5
Final weight range	Not bounded
Life length	80 actions
Action duration	300 ms

Acknowledgments

We would like to thank Edo Franzi and André Guignard for the important work in the design of Khepera. Dario Floreano has been partially supported by the Italian National Research Council, grant n. 92.01065.PF67 (co-ordinator W. Gerbino). Francesco Mondada has been partially supported by the Swiss National Research Foundation (project PNR23).

References

[1] D. H. Ackley and M. L. Littman. Interactions between learning and evolution. In C.G. Langton, J.D. Farmer, S. Rasmussen, and C. Taylor, editors, *Artificial Life II: Proceedings Volume of Santa Fe Conference*, volume XI. Addison Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, 1991.

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

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

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

[5] D. T. Cliff and S. G. Bullock. Adding "foveal vision" to Wilson's animat. Technical report CSRP 263, University of Sussex, School of Cognitive and Computing Sciences, December 1992.

[6] D. Floreano. Robogen: A software package for evolutionary control systems. Release 1.1. Technical report LabTeCo No. 93-01, Cognitive Technology Laboratory, AREA Science Park, Trieste, Italy, 1993.

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

[8] I. Harvey, P. Husbands, and D. T. Cliff. Issues in evolutionary robotics. In J. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1992. MIT Press-Bradford Books.

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

[10] C. G. Langton. Preface. In C.G. Langton, J.D. Farmer, S. Rasmussen, and C. Taylor, editors, *Artificial Life II: Proceedings Volume of Santa Fe Conference*, volume XI. Addison Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, 1991.

[11] P. Maes. Behavior-based artificial intelligence. In J. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1992. MIT Press-Bradford Books.

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

[13] D. J. McFarland and A. Houston. *Quantitative Ethology: the state-space approach.* Pitman Books, London, 1981.

[14] 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.

[15] 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.

[16] S. Nolfi. Genotypes for neural networks. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks.* MIT Press, Cambridge, MA, 1993.

[17] S. Nolfi, D. Parisi, and R. Pedone. How noise helps generalization in feed-forward networks. In E. R. Caianiello, editor, *Parallel Architectures and Neural Networks: Proceedings of the 6th Italian Workshop on Neural Networks*, Singapore, 1992. World Scientific P.

[18] R. M. Sibly and D. J. McFarland. A state-space approach to motivation. In D. J. McFarland, editor, *Motivational Control Systems Analysis.* Academic Press, London, 1974.

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

[20] P. F. M. J. Verschure, B. J. A. Kröse, and R. Pfeifer. Distributed adaptive control: The self-organization of structured behavior. *Robotics and Autonomous Agents*, 9:181–96, 1992.

[21] X. Yao. A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, 4:203–222, 1993.