Evolution of neural control structures: some experiments on mobile robots

Francesco Mondada Dario Floreano

Laboratory of Microcomputing (LAMI)
Swiss Federal Institute of Technology
Lausanne, Switzerland
mondada,floreano@di.epfl.ch

Abstract

From perception to action and from action to perception, all elements of an autonomous agent are interdependent and need to be strongly coherent. The final behavior of the agent is the result of the global activity of this loop and every weakeness or incoherence of a single element has strong consequences on the performances of the agent. We think that, for the purpose of building autonomous robots, all these elements need to be developed together in continuous interaction with the environment. We describe the implementation of a possible solution (artificial neural networks and genetic algorithms) on a real mobile robot through a set of three different experiments. We focus our attention on three different aspects of the control structure: perception, internal representation and action. In all the experiments these aspects are not considered as single processing elements, but as part of an agent. For every experiment, the advantages and disadvantages of this approach are presented and discussed. The results show that the combination of genetic algorithms and neural networks is a very interesting technique for the development of control structures in autonomous agents. The time necessary for evolution, on the other hand, is a very important limitation of the evolutionary approach.

Keywords: Genetic Algorithms – Neural Networks – Autonomous Agents – Mobile Robots

1 Introduction

The design of physical autonomous agents and their control structure continues to be a hot research field, where techniques are applied to several kinds of robots [Brooks90] [Franceschini91] [Yuta87] [Mondada93] in a large number of possible application fields [Schofield94] [Everett93] [Wendell93] [Yamamoto93] [Burhanpurkar94]. The central problem is how to give the necessary autonomy to the agent while still controlling its final useful behaviour. Even the basic knowlegde and fundamental mechanisms of functioning required by the agent give rise to big discussions. Both from the biological and from the robotics point of view, people like Braitenberg [Braitenberg84] and Brooks [Brooks91] have shown that it is not necessary to use complex structures or sophisticated internal representations to obtain "intelligent" behaviors. The most important point expressed by these researchers is that the control system must fit very well the robot body and these two elements of the agent must fit the task and the environment in which the agent acts. In ants, as observed by Beckers

[BECKERS94], the mechanisms for path optimisation are very simple but very well adapted to the chemical facilities of these insects, to their enormous sensing capabilities, to the environment and to the tasks that the ant colony must perform to survive.

These observations from the biological world are well known by many robotics researchers who look at them with interest. The main problem is how to design such agents starting from a given technology and a given task to be achieved. By simply copying biological mechanisms we can better understand the strategies used by animals to solve the survival problem, but we need a more basic and simple methodology to design autonomous agents that use our actual technology and fit our needs.

We have addressed this problem using a combination of genetic algorithms and neural networks. We believe that the evolutionary methodology associated with a neural-network based control structure can solve an important part of the actual problems encountered in the design of autonomous mobile robots. We have tested this assumption through a set of three different experiments. In these experiments we focus our attention on three aspects of the control structure: perception, internal representation, and action. Here we present a global overview of our experience together with practical considerations of the advantages and limitations of the evolutionary approach applied to the development of autonomous control structures in real mobile robots.

2 Experimental setup

In all the experiments the evolutionary procedure has been carried out entirely on a real miniature mobile robot – Khepera – designed and built at the LAMI. Khepera is a miniature mobile robot of circular shape, 55 mm in diameter, 30 mm height, and weighing 70 g [MONDADA93] (figure 1). It is supported by two wheels, each controlled by a DC motor with an incremental encoder (12 pulses per mm of robot displacement). In its basic configuration it has 8 infrared proximity sensors placed around its body (6 on one side, 2 on the other) that can also be enabled to measure ambient light. A Motorola 68331 microcontroller manages all these functionalities. Khepera has a modular structure: various modules including sensors and actuators can be added using a piggy - back extension bus. At the software level, every additional module is automatically identified and the extended functionalities become available without any software

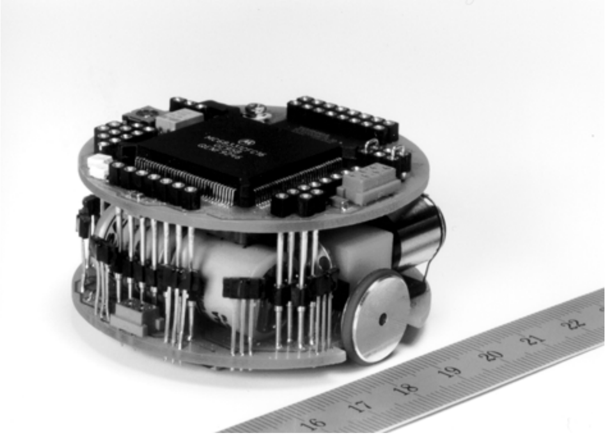


Figure 1: Khepera, the miniature mobile robot (55 mm of diameter).

extension. This software and hardware modularity allows the experimenter to employ the most suitable configuration for the type of environment and behavior under study. Khepera can be attached to a workstation by means of a light-weight aerial cable and specially designed rotating contacts. This configuration allows the experimenter to exploit the computational power (e.g., for neural network computations and on-line analyses), memory storage capabilities (for recording all the relevant data of the experiment) and interaction facilities (e.g., graphical display, keyboard, mouse) of the workstation, while all low-level and real-time processes (such as sensor reading, motor control, and other internal operations) run on the microcontroller that is placed on the robot. The cable can also provide electric power, an important feature to have when genetic algorithms (that require long periods of time) are employed.

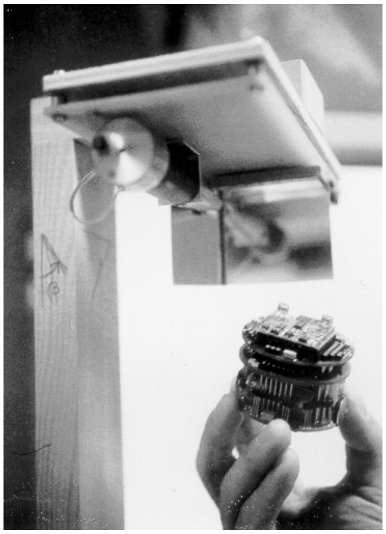


Figure 2: A close view of the robot with the "helmet" for capturing laser signals and the laser device on the background.

In research on autonomous agents the analysis of the resulting solutions and behaviors is very important. It is often the case that the emergent solutions cannot be easily understood with traditional analytic tools [McFarland92]. For this reasons, it can be useful to resort to the methodologies employed by ethologists and neurophysiologists. These techniques require a set of appropriate tools (hardware and software) for recording, storing, and analysing behavioral and internal variables of the agent while it is freely acting in its own environment. For this purpose, an external positioning laser device has been built at the LAMI (figure 2). During the adaptation process, or at a later stage, the robot can be equipped with a special "helmet" for capturing the laser signal and computing its exact position. This information is not passed to the agent's control structure, but is sent through the cable to the computer where a software combines and displays behavioral data with neural network variables. This setup allowed us to perform standard neuro-ethological analyses on the agent in its natural settings.

In all our experiments we employed a standard genetic algorithm [Goldberg89] with fitness scaling and "biased mutations" [Montana89]. The population size was fixed and set to 100 individuals. The neural network synaptic weights and thresholds were coded on the chromosome as floating point numbers. In all the experiments described in this article the network architecture could not be modified by the genetic algorithm. The neural networks employed were multilayer perceptrons of sigmoid units with a set of recurrent connections at the hidden layer

[ELMAN90]. Only for the first experiment on navigation we did not employ hidden units; hence, the recurrent connections were at the output layer. Each chromosome of a population was in turn decoded into a neural network, the input units were clamped to the sensors activations and the output unit values were directly used to control the speed of each individual wheel (we did not employ action commands such as "turn x degrees to the left"); the robot was let free to move for a certain amount of time while its fitness was recorded and stored away for selective reproduction. Between one individual and the next, the robot was programmed to perform a random motion lasting 5 seconds (sufficient – in principle – to go from one side of the "world" to the opposite one) in order to avoid artificial position inheritances between neighbouring chromosomes in the population file. After all the individuals in the population had been evaluated, we applied the three genetic operators (selective reproduction, chromosome crossover, and gene mutation) to create a new population of the same size. Selective reproduction consisted of a linear scaling of the fitness values [GOLDBERG89] 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" [MONTANA89]).

3 Navigation and obstacle avoidance

In the first experiment we attempted to develop a very simple, well defined, and widely studied behavior: navigation while avoiding obstacles. This behavior was chosen so simple in order to test our methodology and assess some basic ideas:

- The evolutionary methodology is known to be slow and most of the actual research work in this domain is carried out in simulations. In our case the evolution is carried out on a real robot because we want to exploit the interaction with the real environment during the evolutionary process. This is the best way to develop a real agent that behaves in a real environment. This first experiment was planned to test the efficiency of "real world evolution" on a real mobile robot, if feasible at all.

- The control system must be suited to the body of the agent and use in an optimal way the sensors and actuators available. When the control structure is designed by an engineer, all elements of the processing chain – from sensors to actuators – are designed as noise-free and linear as possible. However, this is not the way in which biological control systems work. Moreover, in some artificial control systems, these characteristics are introduced at some later stage within the algorithm[1] to improve the processing capabilities of the system. This paradox shows that our current methodologies are limited when dealing with the real world. Therefore, a second goal of this experiments was to observe the ability of the evolutionary process to exploit raw data coming directly from sensors which are not corrected for non-linearities and slightly-differing response properties.

- There are several techniques to change the synaptic weights of a neural network. In case of autonomous agents, it is rather difficult to employ strictly supervised learning algorithms because the correct system output is not always available or computable. Unsupervised

[1] In the form of non-linear activation functions and noise

learning algorithms can be employed to the limited extent in which the behavior is encoded in the choice of the learning rules and architectures used to adjust the synaptic weights. Moreover, these learning rules can hardly deal with complex network structures, like recurrent connections, for instance. Genetic algorithms do not require a direct specification of the desired values for the output units of the network at a given moment, but only a specification of the desired global behavior to be performed during the life of the agent. This is a very interesting feature for the developement of autonomous agents which we felt worth trying on a real mobile robot.

3.1 Experiment and results

The robot was put in a 50x80 cm maze with irregularly shaped walls and the neural network was connected to the 8 infrared proximity sensors and to the two motors. The fitness function Φ was designed to select the individuals that could keep a low sensor activity and high motor activation as follows,

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

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 (all the three variables assume values between 0 and 1). We let the system free to evolve for 100 generations (although the best individuals around the 50th generation already displayed an obstacle avoidance behavior with straight trajectories), until the average fitness reached a stable level. The resulting behavior is a smooth navigation and efficient obstacle avoidance (see [FLOREANO94] for more details and other considerations).

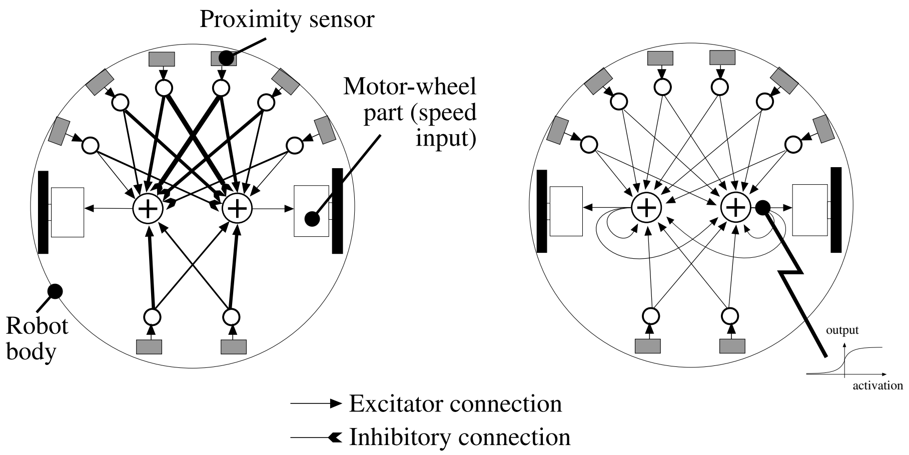
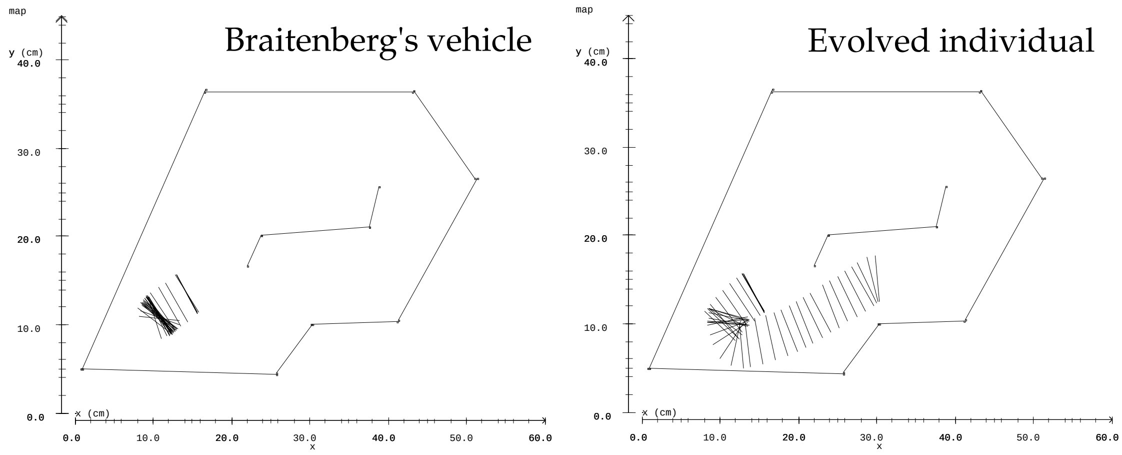


Figure 3: *Left*: The neural network structure implemented on Khepera to perform obstacle avoidance. This structure is inspired upon Braitenberg's vehicle 3c, and adapted to the Khepera sensors and actuators. The performances (robustness of the detection, smoothness of the trajectory) of this control structure are very good, except for small corners where this robot gets trapped. *Right*: The structure of the neural network under evolution.


Braitenberg's vehicle

Evolved individual

Figure 4: *Left*: Braitenberg's vehicle, implemented as a neural network (figure 3 left) on Khepera, gets trapped in a corner. Each sensor sends excitatory signals to the ipsilateral motor and inhibitory signals to the controlateral motor. Given its symmetric and feed-forward structure, this control system gets stuck when it receives equal stimulation from two opposite sensors. The small bars in the figure represent the axis between the two wheels. The position was computed and plotted every 100 ms by using the information provided by the laser device. *Right*: The evolved agent can easily negotiate the same symmetric sensory stimulation by exploiting the recurrent connections at the output layer. This agent did never get stuck in any of the difficult situations used in our tests.

3.2 Discussion

From the analysis of the resulting behaviors, we noticed that all the individuals in the population had developed a preferential direction of motion, corresponding to the side where the higher number of sensors are located (this happened for three out of three genetic runs, each starting from initial different synaptic weights). This autonomous choice was the result of a self-selection of the most relevant sensory devices for performing adequate actions. Individuals that moved in the other direction (where there are only two sensors) disappeared soon from the population because they probably failed to accurately detect (and thus avoid) spiky corners or plain walls from certain angles.

Not only the control structure was coherent with the body of the robot, but also the actions of the robot were coherent with the sensor characteristics. In fact the maximal speed used by the evolved control structure was not the maximal speed available on the robot during this experiment, but was much lower. On one hand this can be considered as very strange, due to the fact that the evolutionary process favours a fast robot (the term V in the fitness function). On the other hand, the short-range detection power of the sensors (about 5 cm) and the slow sampling rate (set to 300ms) are such that it is sometimes impossible, at maximum speed, to detect a wall before crashing into it in the narrow environment employed for the experiments. The term $(1 - i)$ of the fitness function is responsible for the development of a slower and more careful robot. The result of this is a reduction of the maximum cruising speed to approximately 70% of the maximum available speed.

When we compared the resulting-network behavior with the behavior of a Braitenberg's vehicle [Braitenberg84] designed for obstacle avoidance and implemented on Khepera (figure 3 left), our agent performed significantly better by finding its way out of symmetric situations (such as corners or dead-ends) even though it had not been specifically designed for those situations (figure 4). This was due to a proper development of the synaptic weights for the recurrent connections.

In this first experiment we also found a first problem. As in many other experiments using genetic algorithms, the design of the fitness function was quite delicate. In preliminary experiments we set the fitness function as follows,

$$\Phi = V(1 - i) \tag{2}$$

which seemed the best indicator of the behavior described above. However, when this fitness function was employed, the evolutionary process developed agents which turned in circles with a diameter a little bit smaller that the larger space available in the environment. This was a very smart behavior that maximised rotation speed and minimised collisions with obstacles, but was not the behavior that we desired. For this reason we added the third term, which penalized bending. We tried several variations before reaching the final and good one (where the square root of the rotation difference was finally introduced).

4 Homing

The results described in the previous experiment were encouraging because the paradigm employed provided a number of interesting and efficient solutions without much human intervention. Nevertheless, several aspects of the resulting behavior were explicitly described in the fitness function – even though the level of description was very different from that employed by the classic programming approach. This was in part due to the fact that, whereas biological organisms have internal drives (which must be satisfied in order to survive), our simple robot did not have particular needs to satisfy: the fitness function was an artificial mean of providing the robot with such drives. In the second experiment we decided to use a more ecological setup by providing the robot with a short-life simulated battery[2] and the environment with a restricted area where the robot could charge its battery; along with these modifications, we simplified the fitness function by canceling the "bending-penalty" term.

4.1 Experiment and results

The environment was similar to that employed in the previous experiment (although the geometrical shape was more regular and without obstacles in it). The floor of the recharging area was painted black and a tower with several small DC lamps was placed above it illuminating the whole arena (figure 5). The neural network received input from the eight proximity sensors, from three ambient light sensors (one placed beneath the robot surface to detect the floor brightness, and the other two on the two opposite sides of the robot), and from a sensor of battery charge. The battery had a simulated linear discharge that allowed an autonomy of approximately 20 seconds (50 actions ca.). If the robot happened to pass by the recharging area the battery was automatically and instantaneously recharged. In order to avoid unlimited life durations, we established a top limit of 60 seconds per individual, after which the life was automatically terminated. The fitness function employed was a simplified version of that used

[2] We could have used the battery already placed on Khepera, but the genetic run would have lasted more than 270 times longer.

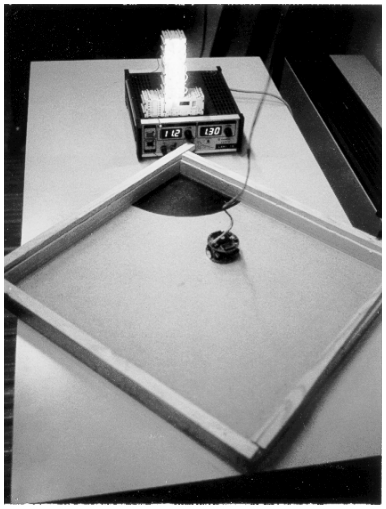


Figure 5: The environment with the light tower, the recharging area (where the floor is painted black), and the robot.

in the previous experiment, without the term $(1 - \sqrt{\Delta v})$: it did not include any term for driving the robot toward the recharging area or for processing the information coming from the ambient light receptors and from the battery sensor.

Both the average fitness of the population and the fitness of the best individuals steadily increased during 240 generations, showing that the robot learnt to move around the environment without hitting the walls (see [Floreano95a] for further details, full analysis, and other considerations). In the meanwhile, also the duration of the lifespan of the individuals increased and reached the top limit after 150 generations (figure 6). By periodically recharging the battery, the robot could perform a higher number of actions and thus obtain higher fitness scores.

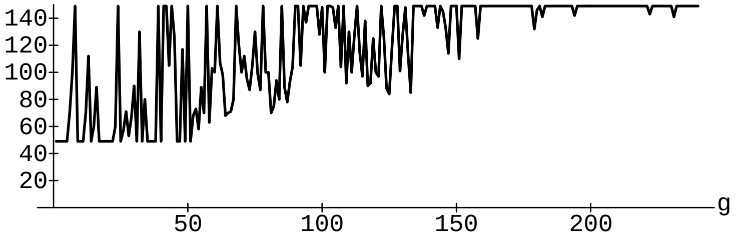


Figure 6: The number of actions performed by the best individuals at each generation. Each action lasts about 380 ms. Since each individual starts its own life with a fully charged battery, 50 actions are a lower bound (see text).

In order to analyze the agent's behavior we equipped Khepera with the helmet for capturing the laser signal and computing its position. We downloaded the neural network corresponding to the best individual of the last generation into the robot, and synchronized the helmet device with the network activations. The information about the robot exact position, the sensor and motor states, and the activation of the internal network nodes while the robot was freely moving were passed to the workstation where a special software performed neuroethological analyses which related the neural activity with the displayed behaviors.

These analyses showed that the robot spent most of its time wandering around the environment without hitting the walls, but when the battery reached a minimum level, it started orienting toward the recharging area. The critical battery level that corresponded to a change in behavior was autonomously estabilished by the robot and varied depending on the robot location. The robot was extremely precise in timing its return to the recharging area, spending about 90% of its time moving in the outside environment.

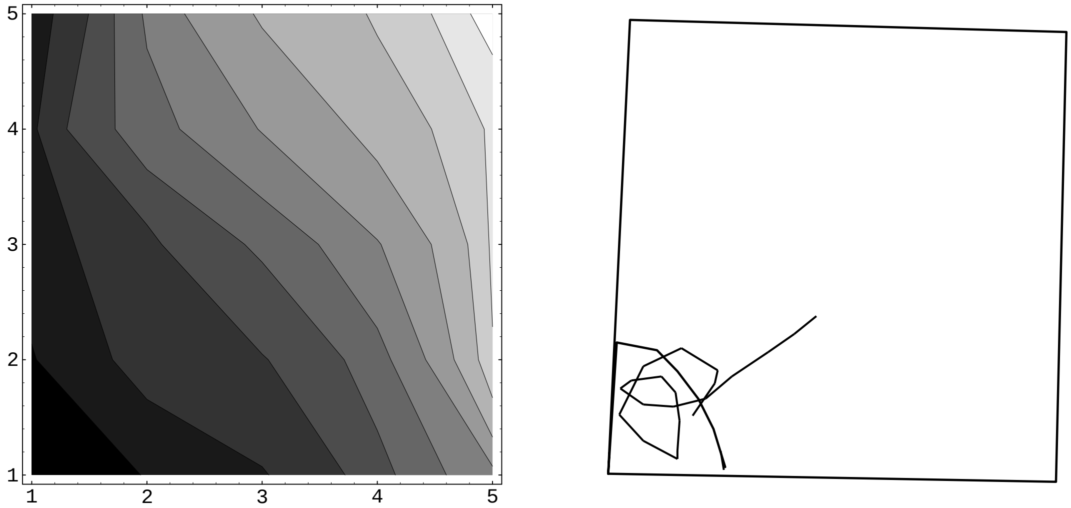


Figure 7: *Left*: Activation levels for an internal node measured at several locations evenly distributed in the environment. The recharging area is located in the bottom left corner (corresponding to the highest activation level). *Right*: The trajectory performed by the robot when the blackpainting is removed from the recharging area and the battery is not recharged (The recharging area is located in the bottom left corner). All data points have taken using the laser device. The recharging area boundaries are drawn here only for sake of clarity.

4.2 Discussion

In this experience the robot autonomously evolved to ability to use the raw sensor data and built an internal representation of the world in order to find the recharging area and return to this place at a given time. This behavior is based on an accurate evaluation of the battery residual time and on an internal representation of the environment. In fact some of the hidden nodes displayed activation levels that clearely mapped the environment geometry (figure 7 left), providing a basic information about the relative position of the battery recharger. Nevertheless this information was not sufficient to achieve a complete and correct navigation. The final behavior was achieved by a strong interaction between this information, the data coming from the collision sensors and the actions taken by the agent. This interaction can be observed both at the behavioral and node activation level. Figure 7 describes the behavior of the evolved robot when the battery recharger was removed: the robot displayed a clear active search of the recharger, which made combined use of perception (light, obstacle and recharger detection), internal representations (self-location) and action (movements in the former recharging area). Once it reached the area after a normal path, the robot started to explore its surroundings without leaving that zone until its battery was completely exhausted.

The evolutionary process was driven by a very simple fitness function, showing that the problem of the fitness design is often ill posed. The need of a very complex and detailed fitness function is often due to a wrong conception of the problem. It is not necessary and perhaps also wrong, as already mentioned, to control all choices of the autonomous agent. The behavior can emerge from the autonomous interaction of the agent with its environment and satisfy the final goal of the agent. For this reason the fitness function could be very general, which is an interesting advantage for the designer, giving the autonomous agent sufficient freedom in its structural and functional choices. This experience clearly showed that the evolutionary process, if well posed, takes into account not only the physical configuration of the agent, but also every implicit need for the self-sufficiency of the robot. From this point of view the fitness function should only describe the real final useful function that the agent should perform for the user, without any detailed instruction about how to perform this function and how to perform all basic operations necessary to the self-sufficiency of the robot.

If on one side the evolutionary process showed very interesting advantages for the development of complex autonomous agents, on the other side we observed an important disadvantage of this approach: the time needed for the evolution. Whereas this disadvantage is minimised in computer simulations, in real experiments this is a major limitation. One run of the experiment described above took about 10 days. If we compare the difference in time and complexity between this experiment and the previous one, it is easy to understand that the practical development of useful autonomous behaviors is quite impossible with this approach: it will simply take too much time. The solutions found on pure simulations are not useful, as already mentioned, but the combination of simulation and real experimentation could help in finding basic structures and behaviors [NOLFI94]. Nevertheless, new strategies need to be found for the efficient development of a fully autonomous control structure embedded in a real mobile robot which can provide practically useful behaviors.

4.3 Grasping

In a further attempt to make the settings even more ecological, we decided to make the fitness function simpler by eliminating the two components employed in the previous experiment. Those components partially specified what the robot should have done to achieve higher fitness, i.e. avoid obstacles and keep motors active. We thought that this behavior specification could be lifted if a number of objects were scattered in the environment and the robot provided with a gripper module. In that case, the fitness would only be the number of objects gripped. Trajectories, homing, wall avoidance, and other behaviors (such as searching for and, possibly, grouping the objects) should be purely emergent. For this reason the fitness function was defined as follows,

$$\Phi = n \tag{3}$$

where n is the number of objects gripped during the test period of each individual.

However, grasping an object is not a simple task. It involves object recognition and sensory-motor co-ordination. One of the aspects of grasping consists in a sequence of preparatory decisions and actions. These processes include the choice of the object to be grasped, deciding about the feasibility of the action, and positioning the body system in an adequate position. In the preliminary explorations reported here, we addressed only the reaching-for-grasping task to test the feasibility of our approach.

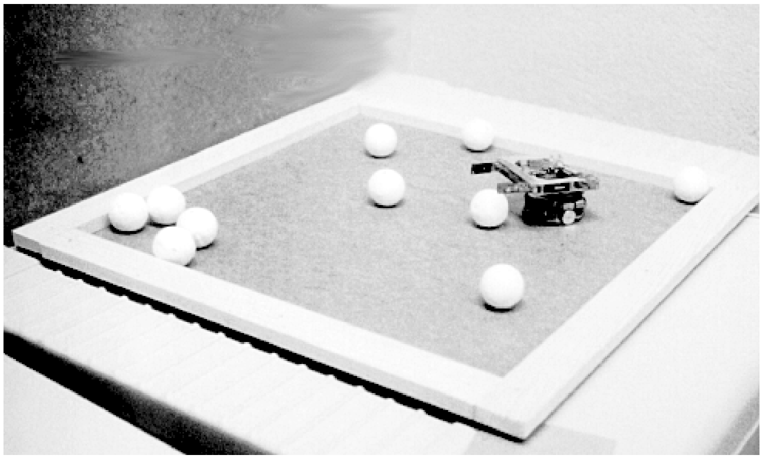


Figure 8: The environment used for evolving the gripping abilities.

4.4 Experiment and results

As for the previous experiments, we let the robot free to evolve the adequate sensory processing and control system during its own interaction with an arbitrary environment by employing the same genetic algorithm and the same neural network structure. The robot employed in our explorations was the miniature mobile robot Khepera with an added gripper module. The environment was an arena of the same size of that employed in the homing experiment. The objects were a number of white balls (38 mm in diameter) made of pressed paper (figure 8). Only the activations of the proximity sensors were given as input to the neural network. The neural network had three output units, two controlling the motor activation, and the third the initiation of a gripping action. The gripping action itself was implemented as an automatic procedure: when the neural network decided to perform a grip, the gripper was lowered, the two segments were closed, the gripper was lifted well above the robot top surface, the segments were opened again (so that the ball would fall and bounce back into the environment), and finally the gripper was positioned in the resting position. Since the gripper had an internal light barrier, the software could detect whether a ball had been gripped. The fitness function was simply the number of balls gripped.

In a number of experiments with different initial random synaptic weight values we observed an improvement across generations of the ability to grip balls (figure 9). We started by throwing 10 balls in the arena and let the robot adapt to this concentration. When its performance reached a peak level (104th generation), we subtracted 5 balls from the arena and continued evolution. After a temporary drop in performance, the agent could rapidly reach again the same performance level by improving its search strategies (see [Floreano93] for a similar procedure). Although it is difficult to plot typical trajectories because they depend very much upon the configuration of the balls, the strategy employed by the best individual was rather simple. The robot moved "backward" (i.e., in the direction where only two proximity sensors are placed) until it detected something; then it turned on itself until the frontal sensors corresponding to

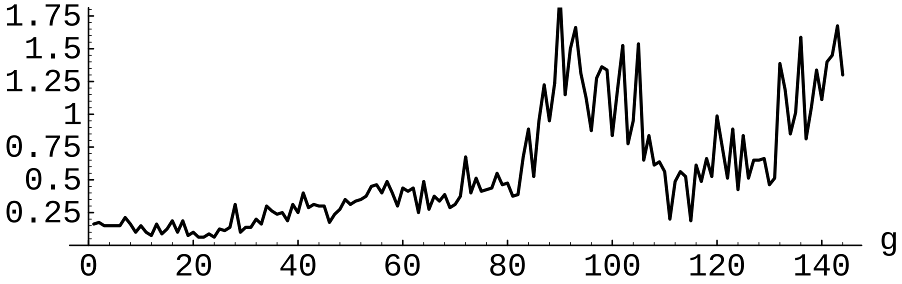


Figure 9: Average number of successful grips across generations. The temporary drop in performance around the 104th generation corresponds to the subtraction of 50% of the balls.

the gripper aperture detected the presence of the ball, and it activated the gripping sequence. If some of its sensors were still active, it performed a further turn on itself to position the gripper in the correct direction and performed another grip.

4.5 Discussion

Also in this case, the obtained behavior is a combination of perception, internal representations, and action, with a strong coherence and complementarity between layers. In fact, this complementarity was necessary because the proximity sensors could not detect objects that were more distant than the gripper segments. It was thus impossible to approach them correctly (i.e. in the direction of the gripper aperture) from a distance. The turning strategy overcame this limitation by appropriately exploiting a combination between an action and the sensory information available and processed by the robot.

If on one side with a general fitness function we gave complete freedom to the evolutionary development of the behavioral strategy, on the other side the basic gripping actions were constrained as a predefined sequence of actions. This limitation was introduced here to reduce the evolutionary time. This choice emphasized an important problem in evolutionary robotics: the choice of the appropriate building blocks. Few building blocks speed up the evolutionary process, but limit the possible range of evolved behaviors; on the other side, several simple building blocks mean a larger search space and a consequent longer development time.

These two problems (evolution time and building block structure) are strongly connected and central for improving the convergence rate, complexity, and quality of the evolved behaviors. An approach that seems interesting in this direction is the incremental one: As described above it is possible to start with simple tasks and increase the complexity during the evolution, building in an incremental way the final expected behavior (see also [Floreano93, Harvey94]). This kind of constructive approaches need to be better understood in order to improve the evolutionary technique and achieve the evolution of complex behaviors.

The experiment on grasping showed another important limitation of the evolutionary approach. During the initial tests we used the first prototype of gripper available. Since the genetic algorithms required several days of no-stop activity, the gripper broke down several times. In fact the gripper was unable to perform more than 10 - 20000 operations, and this number corresponded to only two days of continuous activity. Beside this problem in reliability which was later corrected by a new version of the gripper module, it was still necessary to protect the gripper against random and self-destructive operations, such as trying to grip the walls. Although these behaviors occurred mainly in the initial generations, there were always sub-optimal individuals even in the last generations. These individuals are indeed important for the search process in genetic algorithm which is essentially based on variations in the population, but they can become a problem when the entire population is tested on the same physical robot. In nature this is not such a problem: there is plenty of disposable biological hardware and time is not an issue.

5 Related work

Several researchers employ evolutionary techniques for the development of autonomous agents [Cliff94][Brooks94][Deneubourg93], but only few consider with care the aspect of a final implementation on a real device like a mobile robot (see [Husbands94] for an overview). Even within this small subgroup of people interested in realistic implementations, most of the research and results are based on computer simulations, rather than on real robot implementations. We believe that this trend is mainly motivated by time constraints and reliability issues, which are important problems, as clearly shown by the experiments reported in this article. Even Rodney Brooks, normally rather skeptical about simulations, acknowledged in [Brooks92] that these constraints would probably require the use of simulation, but stressing again the dangers of this approach. Pure simulation is indeed an approach where many results have been achieved (e.g., [Koza92], [Reynolds93], [Reynolds94]) but never tested and validated on a real device.

Nolfi, Floreano, Miglino and Mondada [Nolfi94] illustrate with several examples various methodologies that can be used in evolutionary robotics, spanning from careful simulation to experiments carried out entirely on real robots. The evolved structures are based on neural networks and the robot used are Khepera and a Lego robot controlled by a Miniboard computer. The simulation appears to be an interesting tool, indeed complementary to the real experimentation, but not sufficient to demonstrate the success of a methodology. The real robot plays an important role for the design of the simulation, and the performance measurements made on it are the only valid final verification. One of the examples shows how initial stages of the development could be carried out on simulations and then continued on the real robot.

Colombetti and Dorigo [Colombetti93] also applied this mixed technique to develop the control structure ALECSYS. The evolved system is based on classifier systems and can coordinate several different behaviors. The design and evaluation of this control system are strongly based on ethological consideration. The evolution has been carried out in simulation, but performances have been measured on a real robot.

Harvey, Husband, and Cliff [Harvey94] have used a gantry-frame to move a camera in a real environment for long periods of time, evolving in such a way visually-guided "robots". Despite the fact that the camera takes a real view of the environment, the motion of the robot, based on absolute cartesian coordinates, is not realistic. The movement of the wheels and the interaction wheel-ground, who is very important in mobile robotics, is not present on this "robot" and need to be simulated, introducing the same problems as for pure simulation. The control structure is based on neural networks and the evolution is applied on the morphology of the visual system as well as on the structure of the neural network. The results showed very smart, economical and efficient solutions, but the fitness functions used were specifically and carefully designed and applied incrementally.

Finally, Baluja [Baluja95] evolved neural networks for the control of an autonomous land vehicle (ALVINN). The evaluation of the neural network is based on an accurate knowledge of the response requested from the control system. When combined with gradient descent, the results are slightly better than those found by either learning algorithm alone.

6 Conclusion

We have shown that an approach that combines genetic algorithms and neural networks can be employed on a real autonomous robot and give interesting results.

The evolutionary approach is very well suited for the autonomous development of an agent in its real environment. The continuous interaction with the environment and the simulatneous autonomous development of the control structure allows the generation of an homogeneous system that fits well the requirements of the environment. The different (and often artificially separated) computational layers of this structure become strongly interdependent and work in complete interaction wich each other, with the physical characteristics of the robot, and with the environment. This interaction, found at the level of the control structure, is also present at the behavioral level, and gives rise to very efficient strategies.

The principal disadvantage of this approach is the time required: A full developement of a control structure needs much time, especially if carried out on a real robot. This problem is associated with the type of basic building blocks under evolution. New efforts need to be made to better exploit the evolutionary mechanism: this implies a better understanding of the influence of several aspects, like the genetic coding, the interaction with other adaptive mechanisms (like maturation and learning during life, for instance), the chaining of several evolutionary processes, or other sub-mechanisms that can help improving this methodology. A first simple attempt in this direction has been described in the last experiment, but this preliminary test need to be completed by a better theory and more detailed tests. In our current research we are also trying to implement a simple form of associative learning during life (with pure Darwinian inheritance) to speed up the adaptation process in complex situations (preliminary results are described in [Floreano95b]).

Finally, we have observed that the evolutionary approach, as implemented in these three examples, can be hardly scaled up to bigger or more complex robots for a very simple problem of reliability and mechanical robustness. The approach presented here, which starts from random behaviors, requires a robot which is mechanically very robust. Complex devices, like arms, that can cause damages when randomly operated, should be avoided at this stage or their actions restricted, even if limiting in this way the autonomous development of the agent. Big robots, due to the physical laws, have less relative mechanical robustness than Khepera. In fact, Khepera (5 cm of diameter, 3 cm of height) at maximal allowed speed (8 cm/s), reports no damages when it crashes into a wall. If we scale up the same situation to a robot 20 times bigger (1 m of diameter, 60 cm of height) moving at 1.6 m/s, the collision with the wall would be destructive. This problem is strongly connected with the previous one (design methodology) and stresses to the importance of the hardware building blocks.

Acknowledgments

We would like to thank Edo Franzi and André Guignard for the important work in the design of Khepera, and Yves Cheneval for the analysis software. This work has been supported by the Swiss National Research Foundation (project PNR23).

Biography

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.

References

[Baluja95] S. Baluja. Evolution of an Artificial Neural Network Based Autonomous Land Vehicle Controller. *IEEE Transactions on Systems, Man, and Cybernetics*, In press, 1995.

[Beckers94] R. Beckers, O.E. Holland, and J.L. Deneubourg. From Local Actions to Global Tasks: Stigmergy and Collective Robotics. In R. Brooks and P. Maes, editors, *Proceedings of the Fourth Workshop on Artificial Life*, pages 181–189, Boston, MA, 1994. MIT Press.

[Braitenberg84] V. Braitenberg. *Vehicles. Experiments in Synthetic Psychology.* MIT Press, Cambridge, MA, 1984.

[Brooks90] R. A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15, 1990. Special issue.

[Brooks91] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–59, 1991.

[Brooks92] R. Brooks. Artificial Life and real robots. In F. J. Varela and P. Bourgine, editors, *Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life.* The MIT Press/Bradford Books, Cambridge, MA, 1992.

[Brooks94] R. Brooks and P. Maes, editors. *Proceedings of the Fourth Workshop on Artificial Life.* MIT Press, Boston, MA, 1994.

[Burhanpurkar94] V.P. Burhanpurkar. Design of commercial autonomous service robots. In *Proceedings of the 25th ISIR, Hannover*, Hannover, 1994.

[Cliff94] D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, editors. *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA, 1994.

[Colombetti93] M. Colombetti and M. Dorigo. Learning to Control an Autonomous Robot by Distributed Genetic Algorithms. In J. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA, 1993.

[Deneubourg93] J. L. Deneubourg, S. Goss, and G. Nicolis, editors. *Preprints of the Proceedings of the Second European Conference on Artificial Life.* Brussels, 1993.

[Elman90] J. L. Elman. Finding Structure in Time. *Cognitive Science*, 14:179–211, 1990.

[Everett93] H.R. Everett, G.A. Gilbreath, T.A. Heath-Pastore, and R.T. Laird. *Coordinated control of multiple security robots*, pages 292–305. SPIE Vol. 2058 Mobile Robots VIII, 1993.

[Floreano93] D. Floreano. Emergence of Home-Based Foraging Strategies in Ecosystems of Neural Networks. In J. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA, 1993.

[Floreano94] D. Floreano and F. Mondada. Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot. In D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, editors, *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA, 1994.

[Floreano95a] D. Floreano and F. Mondada. Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics*, In press, 1995.

[Floreano95b] D. Floreano and F. Mondada. From evolution of innate behaviors to evolution of learning in robotic agents. Technical Report R95.06I, Laboratory of Microcomputing, Swiss Federal Institute of Technology at Lausanne, January 1995.

[Franceschini91] N. Franceschini, J.-M. Pichon, and C. Blanes. Real time visuomotor control: From flies to robots. In *Proceedings of the Fifth International Conference on Advanced Robotics*, pages 91–95, Pisa, June 1991.

[Goldberg89] D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Reading, MA, 1989.

[HARVEY94] I. Harvey, P. Husbands, and D. Cliff. Seeing The Light: Artificial Evolution, Real Vision. In D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, editors, *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 392–401. MIT Press-Bradford Books, Cambridge, MA, 1994.

[HUSBANDS94] P. Husbands, I. Harvey, D. Cliff, and G. Miller. The Use of Genetic Algorithms for the Development of Sensorimotor Control Systems. In J-D. Nicoud and P. Gaussier, editors, *Proceedings of the conference From Perception to Action.* IEEE Computer Press, Los Alamitos, CA, 1994.

[KOZA92] J. R. Koza, J. P. Rice, and J. Roughgarden. Evolution of Food-Foraging Strategies for the Caribbean *Anolis* Lizard Using Genetic Programming. *Adaptive Behavior*, 1:171–199, 1992.

[MCFARLAND92] D. J. McFarland. Autonomy and self-sufficiency in robots. AI-Memo 92-03, Artificial Intelligence Laboratory, Vrije Universiteit Brussel, Belgium, 1992.

[MONDADA93] F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturization: A tool for investigation in control algorithms. In *Experimental Robotics III: Proceedings of the Third International Symposium on Experimental Robotics*, pages 501–513, Kyoto, Japan, 1993. Springer-Verlag.

[MONTANA89] 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.

[NOLFI94] S. Nolfi, D. Floreano, O. Miglino, and F. Mondada. How to evolve autonomous robots: Different approaches in evolutionary robotics. In R. Brooks and P. Maes, editors, *Proceedings of the Fourth Workshop on Artificial Life*, pages 190–197, Boston, MA, 1994. MIT Press.

[REYNOLDS93] C. W. Reynolds. An Evolved, Vision-Based Behavioral Model of Coordinated Group Motion. In J. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA, 1993.

[REYNOLDS94] C. W. Reynolds. Evolution of corridor following behavior in a noisy world. In D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, editors, *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 402–410. MIT Press-Bradford Books, Cambridge, MA, 1994.

[SCHOFIELD94] M. Schofield and H. Grünke. Cleaning robots from concept to product - the users point of view. In *Proceedings of the 25th ISIR, Hannover*, Hannover, 1994.

[Wendell93] H. C. Wendell and J.W. William. *Mobile Systems: Architecture to Applications*, pages 292–305. SPIE Vol. 2058 Mobile Robots VIII, 1993.

[Yamamoto93] M. Yamamoto. Sozzy: A hormone-driven autonomous vacuum cleaner. In *SPIE Vol. 2058 Mobile Robots VIII*, pages 292–305, 1993.

[Yuta87] S. Yuta. Self-contained mobile robots and autonomous navigation using map information. In *AI'87*, pages 122–133, Japan, October 1987.