Institute of Psychology
C.N.R. - Rome

Adaptation as a more powerful tool than decomposition and integration

Stefano Nolfi
Institute of Psychology, National Research Council
15, Viale Marx - 00187 - Rome - Italy
voice: 0039-6-86090231
fax: 0039-6-824737
stefano@kant.irmkant.rm.cnr.it

March 1996

Technical Report 96-03

Department of Neural Systems and Artificial Life
15, Viale Marx
00137 - Rome - Italy
voice: 0039-6-86894596
fax: 0039-6-824737

Adaptation as a more powerful tool than decomposition and integration

Stefano Nolfi
Institute of Psychology, National Research Council
15, Viale Marx - 00187 - Rome - Italy
voice: 0039-6-86090231
fax: 0039-6-824737
stefano@kant.irmkant.rm.cnr.it

Abstract

Recently a new way of building control systems, known as behavior based robotics, has been proposed to overcome the difficulties of the traditional AI approach to robotics. Most of the work done in behavior-based robotics involves a decomposition process (in which the behavior required is broken down into simpler sub-components) and in an integration approach (in which the modules designed to produce the sub-behaviors are put together). In this paper we claim that the decomposition and integration process should be the result of an adaptation process and not of the decision of an experimenter. To support this hypothesis we show how in the case of a simple task in which a real autonomous robot is supposed to classify objects of different shapes, by letting the entire behavior emerge through an evolutionary technique, a more simple and robust solution can be obtained than by trying to design a set of modules and to integrate them.

1 Introduction

Recently a new way of building control systems, known as *behavior based robotics*, has been proposed to overcome the difficulties of the traditional AI approach to robotics. This new approach is based upon the idea of using simple sensorimotor processes, operating in parallel, to enable robots to react quickly and robustly in noisy environments.

Most of the work done in behavior-based robotics focuses on the process of breaking down the desired behavior into an appropriate set of behavioral modules and in (a) the design of a coordination technique powerful enough to produce the correct integration of the designed sub-behaviors (Brooks, 1991) or (b) the attempt to design the behavioral modules in such a way as to allow them to freely cooperate or compete to achieve a coherent behavior without the need of a coordination structure (Steels, 1994).

We believe that both these approaches are insufficient and that the process of breaking down the required behavior into sub-components should be the result of an adaptation process and not of a decision of the experimenter (see also Dorigo and Schnepf, 1993). To support this hypothesis we show how in the case of a simple task which requires the ability to classify objects of different shapes, by letting the entire behavior emerge through an evolutionary technique, a more simple and robust solution can be obtained than by trying to design a set of modules and to coordinate them.

2. The problem

Having at our disposal a Khepera robot we decided to try to develop a control system for such a robot that could to distinguish between objects with different shapes.

Khepera (Figure 1) is a miniature mobile robot developed at E.P.F.L. in Lausanne, Switzerland (Mondada, Franzi, and Ienne, 1993). It has a circular shape with a diameter of 55 mm, a height of 30 mm, and a weight of 70g. It is supported by two wheels and two small Teflon balls. The wheels are controlled by two DC motors with an incremental encoder (10 pulses per mm of advancement by the robot), and can move in both directions. The robot is provided with eight infra-red proximity sensors (six sensors are positioned on the front of the robot, and the remaining two on the back) but in this paper we will refer only to the six frontal sensors.

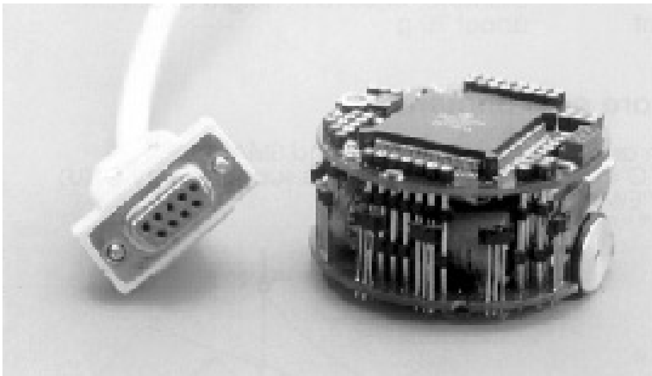

Figure 1. The Khepera robot.

The environment of the robot is a rectangular arena of 60x35 cm surrounded by walls which contains a target object. The walls are 3 cm in height, made of wood, and are covered with white paper. The target object consists of a cylinder with a diameter of 2.3 cm and a height of 3 cm. It is made of cardboard and covered with white paper. The target is positioned in the center of the arena.

The robot has to distinguish between walls and target objects. In particular we want the robot to explore the environment, avoiding walls and remaining close to targets when it finds them.

In order to train the robot we used a simulator of the robot and of the environment (see Nolfi, Floreano, Miglino, and Mondada, 1994). However, we then tested the trained controller on the real robot in the real environment.

A sampling procedure was used to calculate the activation state of the infra-red sensors. The walls and the target objects were sampled by placing the physical Khepera in front of one of them, and by letting it turn 360^{o}, recording, at the same time, the state of the infra-red sensors at different distances with respect to the objects. The activation level of each of the eight infra-red sensors was recorded for 180 different orientations and for 20 different distances. In this way two different matrices of activation were obtained for the two types of objects (walls and target). These matrices were then used by the simulator to set the activation state of the simulated sensors depending on the relative position of Khepera and of the objects in the simulated environment (when more than one object was within the range of activation of the sensors, the resulting activation was computed by summing the activation contribution of each object). This sampling procedure had the advantage of taking into account the fact that different sensors, even if identical from the electronic point of view, actually responded differently. Sampling the environment throughout the real sensors of the robot allowed us, by taking into account the characteristics of each individual sensor, to develop a simulator shaped by the actual physical characteristics of the individual robot we had.

The effects of the two motors were sampled similarly by measuring how Khepera moved and turned for a subset of the 20x20 possible states of the two motors. At the end of this process a matrix was obtained that was then used by the simulator to compute the displacements of the robot in the simulated environment.

In section 3 we describe an attempt to solve the task by using a decomposition and integration approach, while in section 4 we describe a similar attempt using an evolutionary approach in which the decomposition and integration process was performed by the training algorithm. In both cases we used neural networks with different architectures to implement the controllers.

3 Designing by decomposition and integration

In order to accomplish the task described we can break down the required behavior into simpler behaviors and develop a set or modules able to accomplish the required sub-behaviors. In our case we can break down the required behavior into: (a) moving in order to explore the environment, (b) recognizing objects, (c) avoiding objects, (d) approaching and remaining close to an object. Once we have modules able to produce these sub-behaviors we have to design a coordination mechanism that can decide which module has to take control each time step or else design the modules and the corresponding behaviors in such a way that, by letting all of them run in parallel, the interference arising from the interaction between different modules does not impair the accomplishment of the task.

The first module, the third and the fourth modules (i.e. exploring the environment, avoiding objects, and approaching objects) are easy to design and have been implemented several time on Khepera and on other robots. On the contrary, the second module (i.e. classifing sensory stimuli) is not so easy to design. However, because fortunately we know if an input pattern corresponds to a wall or to a target, we can use a supervised learning procedure in order to train a system (for example a neural network) to classify the two types of input stimuli.

We tried 3 different architectures: (a) a feedforward architecture with two layers, an input layer with 6 neurons (coding the activation of the 6 corresponding infrared sensors) and one output neuron (coding with 0 for wall and 1 for targets); (b) an architecture with an additional internal layer with four units; (c) an architecture with an additional internal layer with 8 units. For each architecture we ran 10 training processes starting with different randomly assigned initial weights. Networks were trained by back-propagation (Rumelhart, Hinton, and Williams 1986) for 5000 epochs. A learning rate of 0.02 and no momentum were used. During each epoch, networks were exposed to the sensory patterns perceived by the robot at 20 different distances and at 180 different angles with respect to a wall and to a target, for a total of 7200 different patterns.

Figure 2 shows the percentage of positions from which the networks were correctly able to classify the two types of stimuli (i.e. to produce an activation value below 0.5 in the case of a wall and above 0.5 in the case of a target) for each of the three different architectures. As can be seen, networks were able to correctly classify only 22% of the cases for networks without hidden units and about 35% of the cases for networks with the additional layer of 4 hidden units. The addition of other hidden units did not allows to obtain better performance.

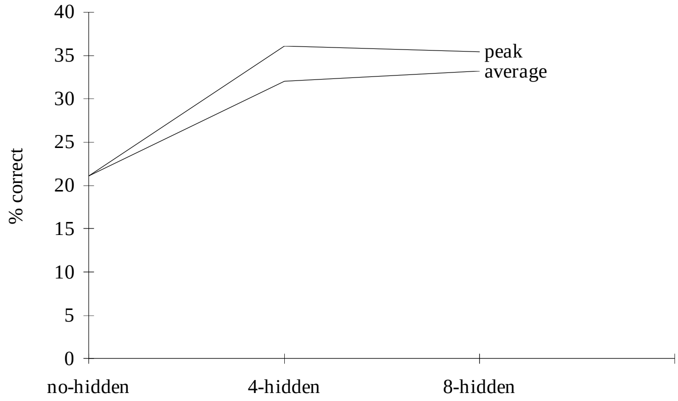

Figure 2. Percentage of positions from which sensory stimuli are correctly classified by network withs no hidden units, 4 hidden units, and 8 hidden units. Average and peak performance of 10 different simulations for each condition. For 20 different distances and 180 different angles (3600 different positions in all) networks were required to produce an output above or below the threshold, depending on the type of object. Responses were considered correct if, in a given position, the network was able to correctly classify both the wall and the target.

The fact that only some of the stimuli can be correctly classified can be explained if we consider that, given the sensory apparatus of the robot, objects can be disambiguated only at a given angle and distance. In other words they are ambiguous in the other cases. If we look at Figure 3, which represents the positions (i.e. the combination of angle and distance) from which networks are able to correctly classify the sensory patterns, we see that stimuli can be correctly classified if the objects are not more than 120^o to the left or the right of the robot face and no more than 32mm away. In addition, there are two zones in which the objects cannot be correctly disambiguated even though they correspond to positions located at about 60^o to the left or the right and a short distance away from the objects.

It is also interesting to note that the zones in which stimuli can be correctly disambiguated are not symmetrical. This has to do with the fact that different sensors, even if identical from the electronic point of view, actually respond differently. As a consequence, it is clear that whether stimuli are ambiguous or not is a function of both the structure of the stimuli themselves and of the sensory apparatus of the agent.

Given these results we can see that a simple feedforward neural networks, irrespective of the number of internal neurons, is unable to correctly classify the incoming input patterns in our environment. In fact, most of the times the network produces incorrect classifications as shown in Figure 2 and 3.

A possible solution to this problem is to add an additional module capable of deciding when the incoming stimuli can be correctly classified by a network of the type described above. When stimuli cannot be classified, the robot can continue to approach the object until a classifiable stimulus is encountered.

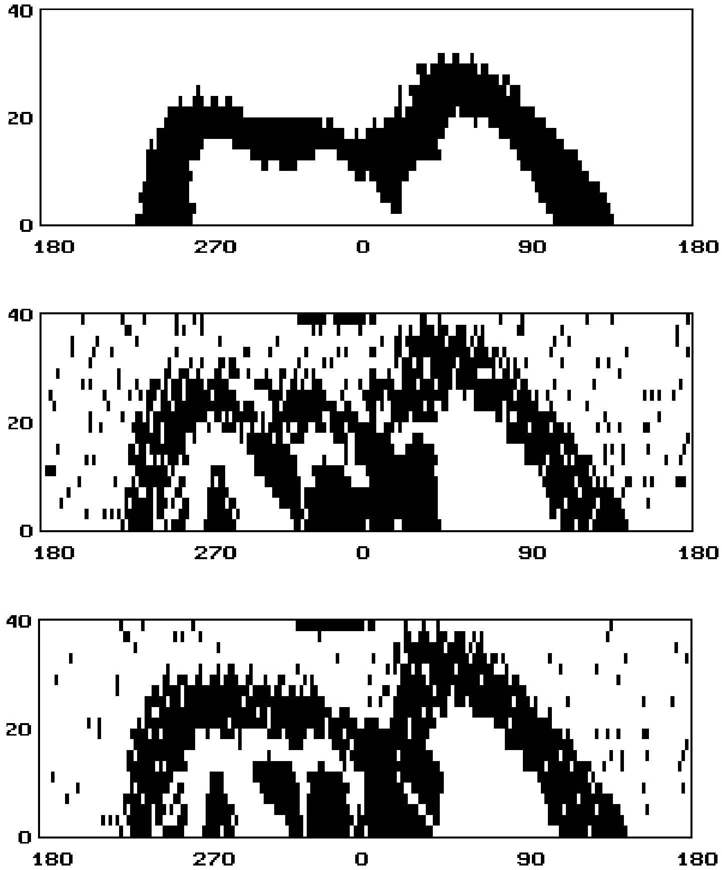

Figure 3. Stimuli correctly classified for each combination of distance and angle of the robot with respect to the objects. The three pictures (from top to bottom) represent the result for the best simulation with no hidden, 4 hidden, and 8 hidden units, respectively.

Scheier and Pfeifer (1995), who developed the control systems for a Khepera robot that performed a task very similar to the one described in this paper, proposed and implemented another interesting solution (see also Scheier and Lambrinos, 1995). Their environment is an arena surrounded by walls containing large and small pegs and a home base with a light source attached to it. The robot has to bring the small pegs to its home base and as a consequence has to distinguish between small pegs from large pegs and walls. In order to accomplish this task the authors designed a set of elementary behaviors (move forward, turn toward objects, avoid obstacles, grasp, and bring to the nest) and allowed them all to run in parallel in order to obtain the desired behavior. Because the robot was programmed to turn around objects and the shape of the object determined the angular velocity of the turning, they decided to train a network to associate the vectors of angular velocities that corresponded to small pegs with grasping behavior and the other vectors of angular velocity with avoiding behavior.

It thus appears possible to find a solution to the problem of classification in this robot-environment context. However, in the next section we will show that by using a different approach a simple feedforward network without hidden units can be trained not only to correctly classify the two types of objects but also to perform the entire required behavior without the need to add any other neural structure.

4 Designing by evolution

Another way to design the control system for an autonomous robot is to use an evolutionary robotics approach (i.e. to try to develop autonomous robots through an automatic design process involving artificial evolution). This approach does not require the required behavior to be broken down into sub-components because the complete solution of the task is expected to emerge during the evolutionary process.

Evolutionary Robotics approaches are based on the genetic algorithm technique (Holland, 1975). An initial population of different "genotypes", each codifying the control system (and possibly the morphology) of a robot are created randomly. Each robot is evaluated in the environment and to each robot is assigned a score ("fitness") corresponding to the ability of the robot to perform some desired task. The robots that have obtained the highest fitness are then allowed to reproduce (sexually or agamically) by generating copies of their genotypes with the addition of random changes ("mutations"). The process is repeated for a certain number of generations until, hopefully, desired performances are achieved (for methodological information see Nolfi, Floreano, Miglino and Mondada, 1994).

To evolve neural controllers able to perform the task described in section 2 we used a form of genetic algorithm. We began with 100 randomly generated genotypes, each representing a network with a different set of randomly assigned connection weights. This was Generation 0 (G0). G0 networks were allowed to "live" for 5 epochs, with each epoch consisting of 500 actions. At the beginning of each epoch the robot was randomly positioned in the arena far from the target. At the end of their life, individual robots were allowed to reproduce. However, only the 20 individuals which had accumulated the greatest fitness (see below) in the course of their life reproduced (agamically) by generating 5 copies of their neural networks. These 20x5=100 new robots constituted the next generation (G1). Random mutations were introduced in the copying process, resulting in possible changes of the connection weights. The process was repeated for 100 generations.

Individuals were scored for the number of cycles spent at a distance lower that 8 cm from the target. This meant that individuals had a probability to reproduce that was proportional to their ability to find the target quickly and to remain close to it.

We tried 3 different architectures and for each of them we ran 10 simulations starting with different randomly assigned weights. All architectures had 6 sensory neurons clamped to the 6 frontal infrared sensors and 2 motor neurons clamped to the left and right motors of Khepera. The three architectures differed in the number of internal units: the first architecture did not have any internal layer, the second had an internal layer with 4 units, and the third had an internal layer with 8 units.

If we look at the fitness of individuals throughout generations we can see how, after a few generations, the best individuals, by incorporating an ability to avoid walls, explore the environment and keep close to the target, are able to earn very high fitness rate by spending most of their time close to the target objects.

It should be noted that networks without internal units are able to solve the task perfectly well. As a consequence the addition of internal units does not produce any improvement in performance and, indeed, result in less efficient individuals in average (see the graph on the right side of Figure 4). This can be explained by the fact that the addition of useless internal neurons, which require longer genotypes, merely enlarge the space to be searched by the genetic algorithm.

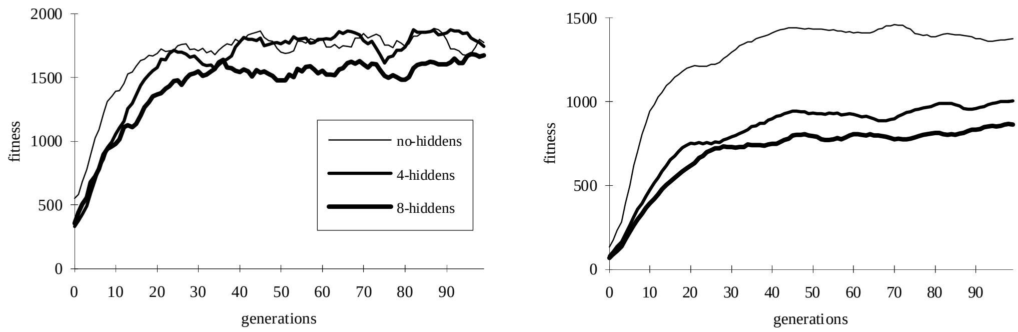

Figure 4. Fitness (i.e. number of cycles, out of 2500, spent close to the target) throughout generations for the three different architectures. The graph on the left shows the performance of the best individual in each generation, while the graph on the right shows the average performance of each generation. Each curve represents the average result of 10 simulations starting with different initial weights.

Nevertheless individual networks have been trained in simulation, the evolved individuals were then downloaded and tested on the real robot and were find to be capable of performing the task perfectly well. No significant difference was observed between behavior in the simulated and real environments in most of the individuals tested (for more information about this point see Miglino, Lund, and Nolfi, in press).

Figure 5 shows the behavior of a typical evolved individual. The robot, after being placed in front of a wall on the right hand side of the environment, recognizes and avoids the wall, recognizes and avoids the new wall it encounters, and finally finds and keeps close to the target. All individuals, like the one shown in the figure, never stop in front of the target, but start to move back and forth as well as slightly to the left and right remaining at a given angle and distance with respect to the target (see graphes ‘a’ and ‘d’ representing the angle and the distance of the robot with respect to the closest object over time).

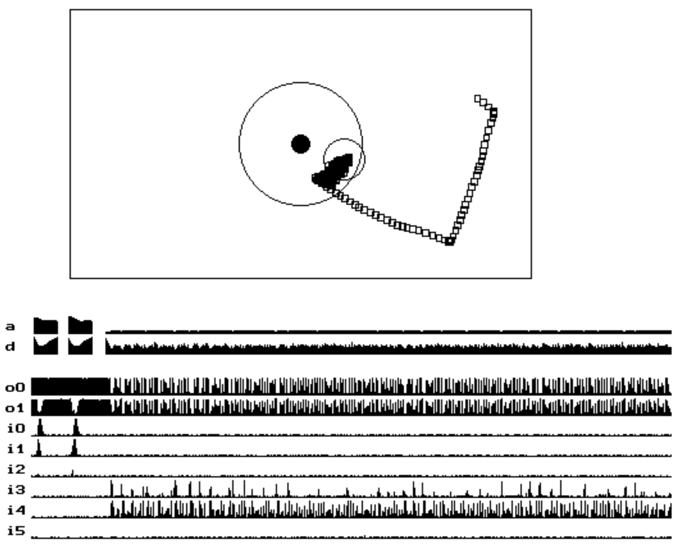

Figure 5. The top part of the figure represents the behavior of a typical evolved individual without internal units. The lines represent walls, the full circle in the center of the arena represents the target object, the large empty circle around the target represents the area in which the robot is rewarded, the

small empty circle represents the position of the robot after 500 cycles, finally the trace on the terrain represents the trajectory of the robot. The graphs in the bottom part of the figure represent the angle and the distance of the robot with respect to the closest object (a,d), the state of the two motor neurons (o0,o1), and the state of the 6 infrared sensors (i0 to i5) over time.

Interestingly enough, the positions relative to the target that individual robots (with and without internal units) maintain once the target has been found, are not located in the discriminative area, that is in the area in which targets and walls can be correctly classified by a network without hidden units (see Figure 3). On the contrary, individual networks do usually start to move back and forth when they reach an area that overlaps the border between the discriminative and non discriminative areas as is shown in Figure 6.

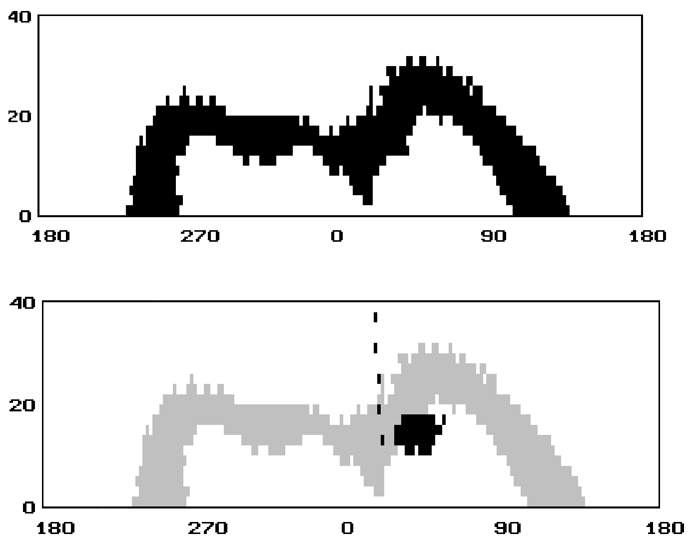

Figure 6. The top picture represents the stimuli that can be correctly classified by an evolved network without internal units (same data as Figure 3). The picture on the bottom represents, in black, the relative positions (angular and distance combinations) that a typical evolved individual assumes with respect to a target once the target has been reached. Each point represents the combination of angle and distance with respect to the target during one cycle.

By analyzing the strategy of the other evolved individuals we observed that, because of the asymmetries in the sensory system and because of the different initial conditions of the evolutionary process, they varied in the relative positions assumed in front of the target (some stop with the target in front or almost in front of them, others with the target on the right side, and only a few with the target on the left side). However all, once they reach the target, start to move back and forth in an area that, as the case shown in Figure 6, overlaps the discriminative area. Therefore all use a strategy that is different from the one we hypothesize as a solution to the problems encountered in section 3 in trying to design the control system with the decomposition and integration approach.

Conclusion

We claimed that behavior based robotics would not be able to successfully overcome the problems encountered with traditional AI by using a decomposition and integration procedure for the development of control systems for autonomous robots. The decomposition and integration process should be the result of an adaptation process and not of the decision of an experimenter who has at his disposal only a few trials. Artificial evolution can be an answer to this type of problem because it allows the supervision of the experimenter to be limited to the design of the evaluation criterion and of the evolutionary conditions without requiring the experimenter to choose how to decompose and integrate the problem.

We took the case of a real problem: trying to design the control system for an autonomous robot capable of navigating in an environment and to classify different type of objects and we showed how following the decomposition and integration approach serious problems are encountered that require complex solution while, by following the evolutionary approach, a simple and robust solution can be easily obtained. This solution has also been proven to scale up in another work in which the robot has been successfully trained to pick up the target objects and bring them out of the arena (Nolfi and Parisi, 1995; Nolfi, 1996).

We also showed how the type of solutions discovered by evolution are different from those that we and other researchers (Scheier and Pfeifer,1995) have hypothesized for the purpose of solving the task described following the decomposition and integration approach. In particular, we showed how by intensively using an active perception strategy the evolved individuals can overcome the problem posed by the fact that the two classes of stimuli are ambiguous in most cases and in particular in some of the positions which are critical because they are the positions occupied by the robot when it remains close to the target as required.

References

Brooks, R. A. 1991. Challange for complete creature architectures. In J. A. Meyer, and S.W. Wilson (eds), *From Animals to Animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior*. MIT Press, Bradford Books. Cambridge, MA.

Dorigo, M., Schnepf, U. 1993. Genetic-based machine learning and behaviour based robotics: A new synthesis. *IEEE transaction on Systems, Man, and Cybernetics*, 23, 1, pp.141-154.

Holland J. H. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, Mich., University of Michigan Press.

Miglino O., Lund, H.H. Nolfi, S. in press. Evolving mobile robots in simulated and real environments. *Artificial Life*, MIT Press.

Mondada F., E. Franzi, P. Ienne. 1993. Mobile Robot miniaturisation: A tool for investigation in control algorithms. In: *Proceedings of the Third International Symposium on Experimental Robotics,* Kyoto, Japan.

Nolfi, S., Florano D., Miglino, O., Mondada, F. 1994. How to evolve autonomous robots: different approaches in evolutionary robotics. *Artificial Life IV, Proceedings of fourth International Workshop on the Synthesis and Simulation of Living Systems*, Cambridge MA, MIT Press.

Nolfi S., Parisi D. 1995. Evolving non-trivial behaviors on real robots: an autonomous robot that picks up objects. In: M. Gori and G. Soda (eds), *Topics in Artificial Intelligence*, Proceedings of Fourth Congress of the Italian Association for Artificial Intelligence, Berlin: Springer Verlag.

Nolfi, S. 1996. Evolving non-trivial behaviors on real robots: a garbage collecting robot. *Technical Report*, Institute of Psychology, C.N.R., Rome.

Rumelhart, D.E., Hinton G.E., Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart, and J.L. McClelland (Eds.) Parallelel Distributed Processing. Cambridge, MA, MIT Press.

Scheier C., Pfeifer R. 1995. Classification as sensory-motor coordination: A case study on autonomous agents. In F. Moran, A. Moreno, J.J. Merelo, P. Chacon (Eds.) *Advances in Artificial Life: Proceedings of the Third European Conference on Artificial Life*, Springer Verlag, pp.862-875.

Scheier C., Lambrinos D. 1995. Adaptive classification in autonomous agents. *Technical Report*, AILab, Computer Science Department, University of Zurich.

Steels L. 1994. Emergent functionality in robot agents through on-line evolutions. In R. A. Brooks, and Maes P. (Eds), *Artificial Life IV*, *Proceedings of fourth International Workshop on the Synthesis and Simulation of Living Systems*, Cambridge MA, MIT Press.