

Evolving Connection Weights Between Sensors and Actuators in Robots

José M. Molina

Grupo de Agentes Inteligentes
Departamento de Informática.
Universidad Carlos III de Madrid.
28911-Leganés (Madrid), Spain.

Araceli Sanchis

Grupo de Vida Artificial.
Departamento de Informática.
Universidad Carlos III de Madrid.
28911-Leganés (Madrid), Spain.

Antonio Berlanga

Grupo de Vida Artificial.
Departamento de Informática.
Universidad Carlos III de Madrid.
28911-Leganés (Madrid), Spain.

Pedro Isasi

Grupo de Vida Artificial.
Departamento de Informática.
Universidad Carlos III de Madrid.
28911-Leganés (Madrid), Spain.

Abstract – In this paper an evolution strategy (ES) is introduced, to learn reactive behaviour in autonomous robots. An ES is used to learn high-performance reactive behaviour for navigation and collisions avoidance. The learned behaviour is able to solve the problem in a dynamic environment; so, the learning process has proven the ability to obtain generalised behaviours. The robot starts without information about the right associations between sensors and actuators, and, from this situation, the robot is able to learn, through experience, to reach the highest adaptability grade to the sensors information. No subjective information about “how to accomplish the task” is included in the fitness function. A mini-robot Khepera has been used to test the learned behaviour.

I. INTRODUCTION

Many real world problems involve interactions with dynamic environments. Some examples appear in control systems which have to be able to deal with unexpected events that occur in their environments. A particular case are robotic systems that have to react with the real physical world. The successful operation in such environments depends on the ability to adapt the system to the changes.

Autonomous robots are sometimes viewed as reactive systems; that is, as systems whose actions are completely determined by current sensorial inputs. This is the base of the subsumption architecture [1], where finite state machines are used to implement robot behaviours. Other systems use fuzzy logic controllers instead [2]. The rules of these behaviours could be designed by a human expert, designed “ad-hoc” for the problem or learned using different artificial intelligence techniques [3].

Machine learning has been applied to shape the behaviour of autonomous agents in these kinds of environments. Some of these techniques become inapplicable to the learning reactive behaviour problem because they require more information than the problem constraints allow.

The control architecture used to evolve the reaction (adaptation) is based on the Braitenberg models. Braitenberg [4] shows clearly and on strong biological grounds that behaviours which seem complex can be generated by means of a very simple control structure.

In the proposed model, the robot starts without

information about the right associations between environmental signals and actions responding to those signals. And from this situation the robot is able to learn through experience to reach the highest adaptability grade to the sensors information. The number of inputs (robot sensors), the range of the sensors, the number of outputs (number of robot motors) and its description is the only previous information.

In this paper, we present the results of a research aimed at learning reactive behaviours in an autonomous robot using an ES. In section 2, we outline the general theory of Evolution Strategies. Section 3 is related to the experimental environment and the goals of the work. The experimental features are described in Section 4. The experimental results are shown in Section 5. The last section contains some concluding remarks.

II. EVOLUTION STRATEGIES

Evolution strategies (ES) developed by Rechenberg [5] and Schwefel [6], have been traditionally used for optimization problems with real-valued vector representations. As Genetic Algorithms [7] (GA) the ES are heuristic search techniques based on the building block hypothesis. Unlike GA, however, the search is basically focused in the gene mutation. This is an adaptive mutation based on the likely the individual represents the problem solution. The recombination plays also an important role in the search, mainly in the adaptive mutation.

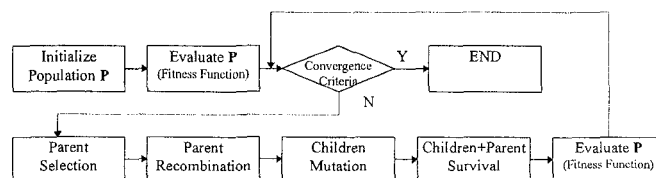


Fig. 1. Schema of an evolution strategy.

Figure 1 shows a typical evolution strategy. First, it is necessary to codify each solution of the problem in a real-valued vector. Each vector represents a solution and also an individual. The method consists in evolving solution sets, called populations, in order to find better solutions. The evolution of populations is performed by

selecting pairs of individuals (parents) that produce new individuals (children) via recombination, which are further perturbed via mutation. The best individual ($\mu+1$ selection) or the best individuals ($\mu+\lambda$ selection), in the set composed by parents and children, are selected to form the next population [8].

An individual is represented by $\alpha = (x_1, \dots, x_n, \sigma_1, \dots, \sigma_n) \in \mathbb{R}^n$, that are the n real values (x_i) and their corresponding deviations (σ_i) used in the mutation process for the ($\mu+\lambda$) ES. The mutation is represented by equations (1) and (2).

$$\sigma_i' = \sigma_i \cdot \exp(N(0, \Delta\sigma)) \quad (1)$$

$$x_i' = x_i + N(0, \sigma_i') \quad (2)$$

Where x_i' and σ_i' are the mutated values, following a normal distribution ($N(\mu, \sigma)$).

However, when a ($\mu+1$) ES is used the mutation process follows the 1/5 rule [8]. In both cases, the recombination follows the canonical GA approach [7].

III. EXPERIMENTAL ENVIRONMENT

The task faced by the autonomous robot is to reach a goal in a complex environment avoiding obstacles found in the path. Different environments have been used to find a generalised solution. The system has been developed using a simulator to prove different characteristics of the system. Finally, a real robot has been used to test the proposed solution.

A mini-robot Khepera [9] has been used, which is a commercial robot developed at LAMI (EPFL, Lausanne Switzerland). The robot characteristics are: 5.5 cm of diameter in circular shape, 3 cm of height and 70 gr. of weight. The robot has two wheels controlled by two motors that let any type of movement. The ES should specify the wheel velocity that could be read later by an odometer. Eight infra-red sensors supply two kinds of incoming information: proximity to the obstacles and ambient light. Instead of using eight sensors individually, to reduce the amount of information six sensors are used and grouped (as Figure 2 shown) giving a unique value, the average, from two input values. Representing the goal by a light source, the ambient information lets the robot know the angle (the angle position in the robot of the ambient sensor receiving more light) and the distance (the amount of light in the sensor).

A simulator developed in a previous work [10] has been used as a complete software for the simulation of mobile robot. Working with a simulation offers the possibility to evaluate several systems in different environments controlling the execution parameters.

The simulated world consists of a rectangular map of user defined dimensions, where particular objects are located. In this world it is possible to define a final position for the robot (the goal to reach), (Figure 3 (a)). In this case, the robot is represented with three proximity sensors and two special sensors to measure the distance and the angle to the goal.

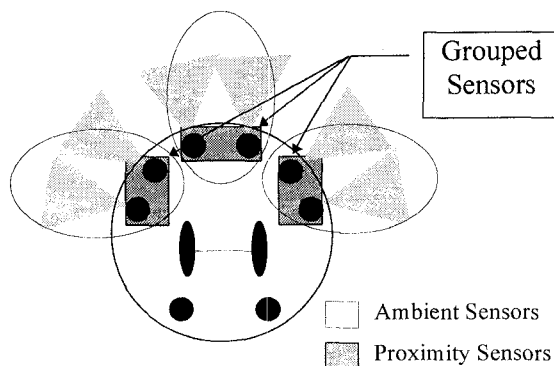


Fig. 2. Sensors considered in the real robot.

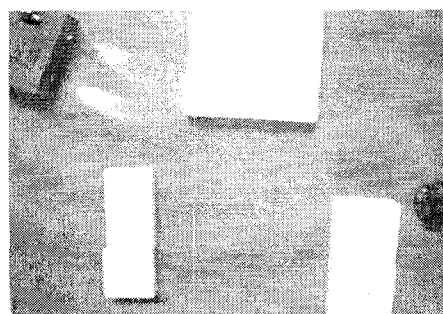
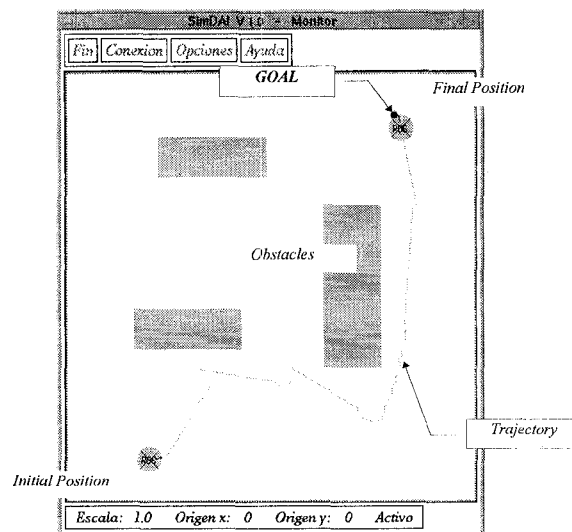


Fig. 3. (a) SimDAI Simulator (Example of one simulated environment).
(b) Example of a real experimental environment.

Different simulated worlds which resemble real ones have been defined before being implemented in the real world. An example of these environments are shown in Figure 3 (a) and Figure 3 (b). The controlled developed is the same in both cases (simulated and real) except the differences in the treatment of the sensors.

For the evaluation of the controller performance (in the real robot as well as in the simulator) some

quantitative measures have been used: the time needed to reach the goal, the trajectory length (measured using the movement of the robot centre), the number of oscillations (measured adding all the angle variations) and the number of collisions.

IV. EVOLVING WEIGHTS BY MEANS OF EVOLUTION STRATEGIES

It has been proved that by means of connections between sensors and actuators, a controller is able to solve any autonomous navigation robotic behaviour [4], just finding the right weights for those connections for each particular problem. The input sensors considered in this approach are the ambient and proximity sensors of Figure 2. The controller outputs are the wheel velocities. The velocity of each wheel is calculated by means of a linear combination of the sensor values using those weights (Figure 4):

$$v_j = f\left(\sum_{i=1}^5 w_{ij} \times s_i\right) \quad (3)$$

Where w_{ij} are searched weights, s_i are sensor input values and f is a function for constraining the maximum velocity values of the wheels.

Weight values depend on problem features. In general, these dependencies are unknown. To find them automatically, an ES is proposed. In this approach each individual is composed by a 20 dimensional-real valued vector, representing each one of the above mentioned weighs and their corresponding variances. The individual represents one robot behaviour consequence of applying the weights to the equation 3. The evaluation of behaviours is used as fitness function.

In order to make the problem more realistic no information about the location of the goal, neither direction nor distance, has been included in the evaluation function. The evaluation of the behaviour of the robot is based on the following concepts:

- Time needed to reach the goal, measuring time (seconds) in the real robot and simulation cycles in the simulator.
- Trajectory length, measured using the velocity values of the motor wheels.
- Number of oscillations using the difference between the wheels velocity.
- Number of collisions using the minimum value of the proximity sensors.

The evaluation begins with a initial value. This value decreases linearly with time, collision number and oscillations and increases, also linearly, with trajectory length. Although, taking into account the trajectory length seems contradictory it has been included not to allow static behaviours.

Each evaluated robot behaviour ends over one environment when the goal has been reached or the time exceed some time out, fixed by the time needed to reach

the goal using a fuzzy controller [3]. To evaluate a behaviour in a set of environments, the minimum value of evaluations is used as the fitness value of the individual.

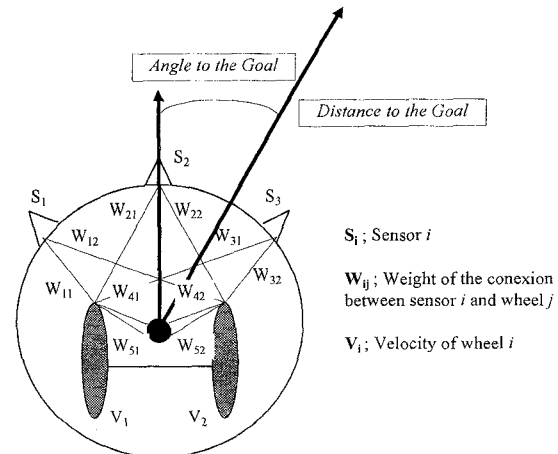


Fig. 4. Connections between sensors and actuators in the Braitenberg representation of a Khepera robot.

V. EXPERIMENTAL RESULTS

Three different experiments have been done, all of them over the same set of environments. The environments have been generated by changing the goal position, number and location of obstacles looking for a generalised environment.

- Experiment one.- Static goals are considered and a $\mu+\lambda$ -ES has been used with the following parameters: $\mu=50$, $\lambda=50$, $\Delta\sigma = 0,1$ and dynamic and preservative selection.
- Experiment two.- The goal follows a trajectory that is randomly generated in each environment. The ES characteristics are the same as in experiment one.
- Experiment three.- Environments and goals are the same as in experiment two but now a 1+1-ES has been used with the following parameters: $C_d=0,82$, $C_i=1,2$ and dynamic and preservative selection.

In Figure 5 the evolution of the fitness in experiment one is shown. The values in the y-axis represent the average of the fitness of the best individual over the whole set of environments for each iteration.

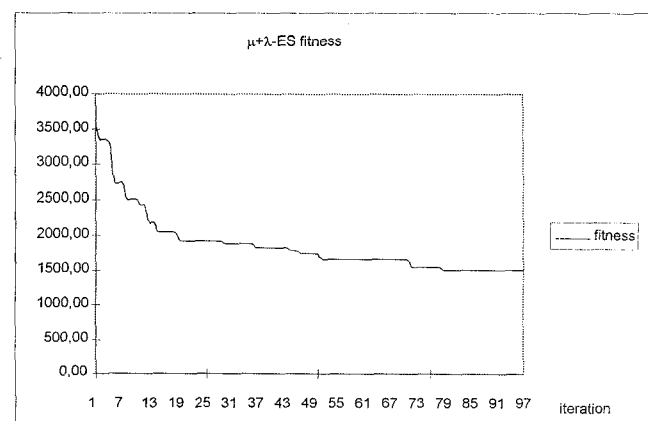


Fig. 5. Evolution of the $\mu+\lambda$ -ES fitness (exp. 1).

Figure 6 shows the evolution of the different quality measures that compose the fitness. The Figure 6 represents the average over the whole set of environments of each quality measure of the best individual (in fitness).

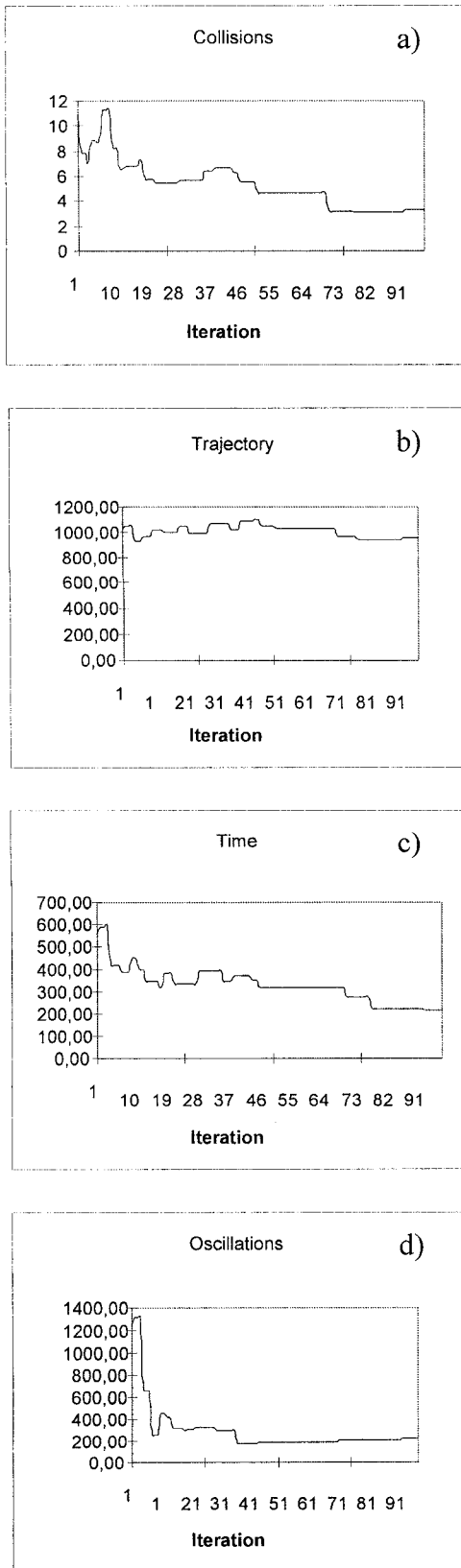


Figure 6. Evolution of the $\mu+\lambda$ -ES quality measures. (a) Collisions, (b) Trajectory length, (c) Goal reaching time and (d) Oscillations.

The plots show the different learning tendencies for each iteration. At the beginning, the system learns to avoid obstacles, and to follow a straight path, the number of collisions and the oscillations, thus, decrease fast (Figure 6 (a) and (d)) and the time needed to reach the goal is kept constant (Figure 6 (c)). This value of time is around the time out, what means that the system is unable to reach the goal.

As no information about the direction nor distance of the goal is known by the evaluation function, the robot reaches the goal, the first time, by chance. The first individuals that reaches the goal have a little fitness advantage although they have loosed the ability of obstacles avoidance, as it can be seen in plots (a) and (c) of Figure 6. In previous iterations the number of collisions and oscillations lead the evolution.

At this moment a selective pressure over the time appears. Therefore, the time needed to reach the goal decreases abruptly, and the number of collisions increases. This new selective pressure gives the ability of reaching the goal, not by chance but as a result of the behaviour of the individuals. This is shown in Figure 6 (c) where the time is continuously decreasing. When this behaviour is able to reach the goal in every environment the selective pressure over the collisions is added to the time selective pressure. From this point the system evolves a complex behaviour to reach the goal avoiding obstacles.

The fitness evolution of the second experiment can be seen in Figure 7. The system evolution is equivalent to the result obtain in the above experiment. In this experiment the complexity of the environment makes the fitness evolve less continuously. The converged fitness value is higher than in the previous experiment due to the difficulty of reaching the goal in all situations.

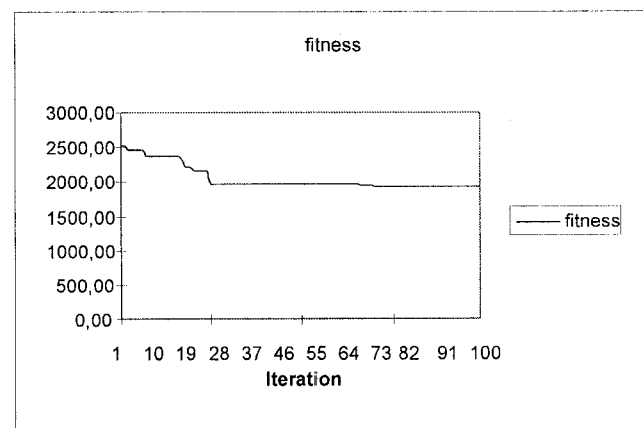


Fig. 7. Evolution of the $\mu+\lambda$ -ES fitness, with a goal in movement

In the last experiment, a simpler strategy has been used in order to improve the total time of the learning process but keeping the same performance. The fitness evolution can be seen in Figure 8.

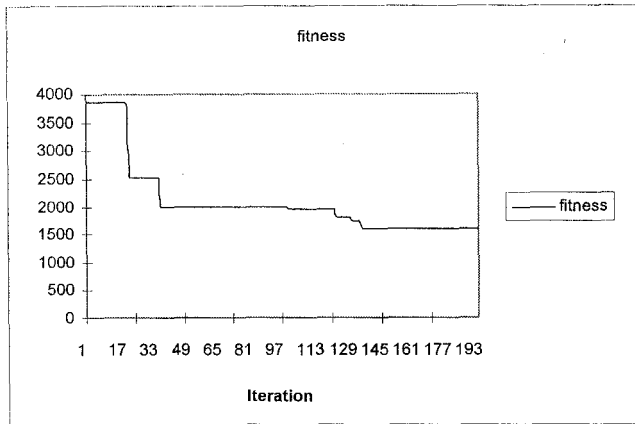


Fig. 8. Evolution of the 1+1-ES fitness, with a goal in movement

The results achieved have been tested in a real robot. The behaviour of the robot is similar to the one obtained in the simulated experiments. An example of execution of the robot is showed in Figure 9, with two frames that represents the starting point (Figure 9 (a)), and the final position (Figure 9 (b)).

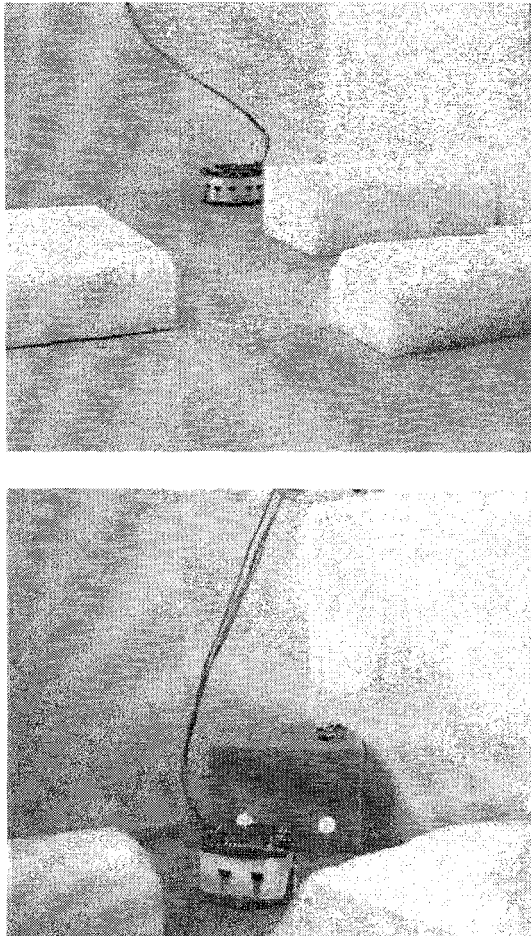


Fig. 9. System Evolution Examples in one real experimental environment. (a) Starting position, (b) Goal reached.

VI. CONCLUSIONS AND FURTHER WORK

The experiments prove the possibility of learning behaviours in an autonomous robot by means of an ES.

The process has been applied on a simple controller where the directly associations between sensors and motors allows to solve a navigation problem. It can be also extended to other more complex controllers where the association could be non-linear functions. It is important to remark that the fitness functions doesn't include any subjective information "how to accomplish the task" but objective information about "how the task has been accomplished".

As a consequence, the learning process can be easily modified in order to consider new problems that could appear such as: surrounding an obstacle, or hiding from the light. The adaptation to new problems does not require too much effort because of no inclusion of local information about the problem in the fitness function.

VII. REFERENCES

- [1] Brooks R. A. "Intelligence without Representation". *Artificial Intelligence*, 47, 1991, pp. 139-159.
- [2] Ishikawa S. "A Method of Autonomous Mobile Robot Navigation by using Fuzzy Control". *Advanced Robotics*, vol. 9, No. 1, 1995, pp. 29-52.
- [3] Matellán V., Molina J.M., Sanz J., Fernández C. "Learning Fuzzy Reactive Behaviours in Autonomous Robots", in *Proceedings of the Fourth European Workshop on Learning Robots*, Germany, 1995.
- [4] Braitenberg V. *Vehicles: experiments on synthetic psychology*. MIT Press Cambridge, Massachusetts 1984.
- [5] Rechenberg, I. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog, Stuttgart 1973.
- [6] Schwefel, H. P. *Numerical Optimization of Computer Models*. New York: John Wiley & Sons 1981.
- [7] Goldberg D., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, 1989.
- [8] Rechenberg I., *Evolution strategy: Nature's Way of Optimization*. In H. W. Bergmann, editor, "Optimization: Methods and Applications, Possibilities and Limitations", Lecture Notes in Engineering, pp. 106-26, Springer, Bonn 1989.
- [9] Mondada F. and Franzi P.I. "Mobile Robot Miniaturisation: A Tool for Investigation in Control Algorithms", in *Proceedings of the Second International Conference on Fuzzy Systems*. San Francisco, USA, 1993.
- [10] Sommaruga L., Merino I., Matellán V and Molina J. "A Distributed Simulator for Intelligent Autonomous Robots", in *Proceedings of the Fourth International Symposium on Intelligent Robotic Systems-SIRS96*, Lisboa (Portugal), 1996.