Neuro-Controllers, scalability and adaptation

José A. Fernández León[1-4], Oscar E. Goñi[1], Gerardo G. Acosta[2-4], Miguel A. Mayosky[3-5]

[1]*INCA/INTIA, Fac. Cs. Exactas, UNCPBA, 7000 Tandil, Buenos Aires, Argentina*
[2]*Grupo de Tecnología Electrónica, Physic Department, UIB, Palma de Mallorca*
[3]*LEICI Laboratory, Fac. Ing., UNLP, 1900 La Plata, Buenos Aires, Argentina*
[4]*CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina*
[5]*CICPBA – Comisión de Investigaciones Científicas de la Prov. de Buenos Aires, Argentina*

{jleon, oegoni}@exa.unicen.edu.ar   gerardo.acosta@uib.es   mayosky@ieee.org

Abstract. A Layered Evolution (LE) paradigm based method for the generation of a neuron - controller is developed and verified through simulations and experimentally. It is intended to solve scalability issues in systems with many behavioral modules. Each and every module is a genetically evolved neuro-controller specialized in performing a different task. The main goal is to reach a combination of different basic behavioral elements using different artificial neural-network paradigms concerning mobile robot navigation in an unknown environment [1][2]. The obtained controller is evaluated over different scenarios in a structured environment, ranging from a detailed simulation model to a real experiment. Finally most important implies are shown through several focuses.

Key words: Evolutionary robotics , Adaptative Systems, Scalability

1 Introduction

Several researchers in AI looks for understanding how complex biological systems are able to adapt, interact [4] and, in particular, build models that let understand and synthesize complex behaviors in autonomous robotic context. Modular neuro-controller hierarchies are studied in order to solve behaviors stability and scalability issues in evolutionary robotic [5]. In fact in this context scalability refers to creation of complex behaviors from simple ones.

Hierarchical and sequential organization of complex activities in a multiple behaviors systems, such as issues in robot navigation and movement learning to the correct target [6][7][8], establish some of real problems that robotic and bio-inspired neuroscience confront. Understanding of such systems within a robotic context aids to explain in some degree the ways of producing, in example, human motor capabilities and its influence in neural and motor system referred to adaptative and recovering capabilities.

One of the most important strategies to deal with scalability in ER is Layered evolution (LE)[9]. LE uses Subsumption[10] architecture concept in the evolutive process, supplying a smart focus in simulated based systems scalability issues. This approach offers an evolutionary sequence from lowest levels (basic behaviors) to highest ones (complex behaviors) throughout a hierarchical organization. The key concepts related to LE are modularity, adaptability and multiplicity[3].

In this paper, concepts from ER are used to get controllers that are able to adapt robot behaviors according to the sensory inputs from the environment in order to solve a navigation complex behavior in an unknown environment, either at simulated level or a real one using a mini-robot Khepera® [11]. This work is within a larger project [3] referred to generation of an adaptative autonomous control for an AUVI [12].

This article is organized as follow: In section 2 a methodology for creation of unknown a priori environment oriented neuro-controllers is presented. The Experimentally obtained results are shown in section 3. Then perspectives and tendencies of this work are exposed in *section 4*. Finally, in *section 5*, conclusions of this work are presented.

2 Initial supposes and experimental setups

Test platform used in both simulated and real environment correspond to the mini – robot Khepera®. The assumed conditions were: a) Robot moves over a plain surface; b) Robot inertial effects and non - holonomic features are not considered; c) Robot moves without slipping; d) Environment is structured and unknown. Some elements of it are static (i.e. Walls, corridors, labyrinths, doors) while others, such as target position references and obstacles, may me modified in each run; e) Variable environmental conditions (i.e., outdoor light influence) are not controlled directly and they are considered as controller noise.

2.1 Validation Experiments

Tasks selection is done in order to reach robotic autonomous mobile navigation in a controlled environment to solve autonomous navigation, focusing on getting sure tracks (sub optimal in general)[13]. Navigation control can be classified into two categories: Local Strategy (reactive) and global (planned). In reactive focus, absolute position is not required and only the interaction should be considered. As exposed in [7], to develop navigation task properly is necessary to know if there is enough free space to avoid an obstacle to reach the target. Reactive techniques are proper for solve navigation task in an unknown environment. . On the other hand, in global strategies an intern world model is needed for tracing generation.

2.2 Evolutionary Algorithm Features

In order to develop basic behaviors, a genetic algorithm based on Harvey's proposal [14] was used. Each individual represents a specific type of neuro-controller and it is selected according to a fitness value. Furthermore just only gene mutation was used.

A population of 30 controllers was considered where each individual is made of a constant number of chromosomes, depending on the dimensionality of each controller (type of neural network).

Each gene is a real (or integer) value depending on the neuro-controller used. Each genotype is awarded according to its observed performance through a fitness measurement that is used as a comparison parameter, establishing a ranking. After that, those genotypes situated at the lower part of this scale (lower half) are discarded as individuals in the next generation. Copies of the upper part replace these individuals. Then, all the individuals, except the first five, are mutated. Next, defined parameters in this process are shown.

Experimental Setup:

Parameter	Value
Number of input neurons[a]	8 (+1 additional input)
Number of hidden neurons	6
Number of output neurons	2
Selection model	elitism (5 first)
Simulation model[b]	Simple
Sensors noise level	---
Number of runs	10
Number of iterations per run	200
Number of generations	300
Number of objects in the environment	0 (phototaxis | learning) 10 (obstacle avoidance)
Initial population	Random
Synaptic weight range	[-1; 1]
Initial range of synaptic weight[c]	[0; 0.01] or [-1; 1]
Selection percentage	50%
Mutation rate	5%
Sensors light range	[0; 1]
Sensors proximity range	[0; 1]
Population size	30
Type of activation function	sigmoid
Type of mutation	Uniform
Type of selection	Ranking

2.3 Neuro-Controller Topologies

The following neural network topologies were used as alternative controller architectures. They were selected to allow performance comparisons among different recurrence and learning paradigms. In what follows, i, j refer to node index and w to synaptic weights in networks:

- *Feed-Forward Neural Network (FFNN):* no recurrence in any level; $w_{ii}=0$ for all layers; without learning associated to synaptic updates.
- *Recurrent Feed-Forward Neural Network (RFFNN):* $w_{ii}\neq 0$ in the hidden and output layer; without learning associated to synaptic updates.
- *Continuous-Time Recurrent Neural Network (CTRNN)*: $w_{ii}\neq 0$ in the hidden and output layer; without learning associated to synaptic updates; temporary activation of neurons.
- *Plastic Neural Network (PNN)*: $w_{ii}\neq 0$ in the hidden and output layer; learning associated to the synaptic updates (Hebb rules).
- *Homeostatic Plastic Neural Network (HPNN)*: $w_{ii}\neq 0$ in the hidden and output layer; learning associated to the synaptic updates (homeostatic Hebb rules).

[a] The additional input only used when it indicates.
[b] Simulator based on [29] descriptions, and with the incorporation of modifications related to improvements about its performance.
[c] Valid for plastic nets ([0; 0.01]).

The “genetically determined controllers” (e.g., FFNN, RFFNN, and CTRNN) are characterized by the sign and weight strength for each synapses, and for “adaptive synapse controllers” (e.g., PNN, and HPNN) this implies a sign, a specific Hebb-adaptive rule and a learning rate[3].

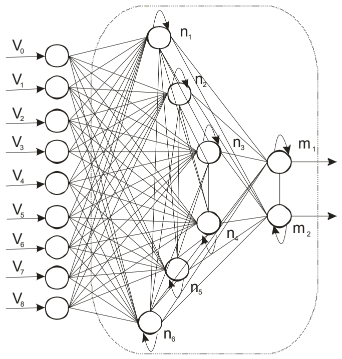

Fig. 1. Schematic diagram of the generic neuronal connectivity (2 motors or output nodes m_x, 6 hidden nodes n_x, 8 input nodes v_x, and 1 additional input v_0).

Certain parameters for each network, such as the number of neurons in each layer (as pointed out in Fig. 1 for a generic net), were considered fixed for each task. That is, instead of letting network size to be adjusted as part of the evolutionary process, a fixed network size was adopted. Besides speeding up training, a small network size results in acceptable real-time performance, an important issue in the experimental phase. However, this is in fact, a human intervention that can constrain a pure evolutionary development.

2.4 Basic Behaviors

Following the LE philosophy, different basic modules were generated at the lowest hierarchy level, using a specific fitness function. Selected tasks for this development were:

- *Phototaxis:* the robot’s ability to reach a light source or coming close to it. The reference for this part of the work was [5]. The fitness function was composed of two variables:

$$\Phi_1 = k \cdot (1 - i) \tag{1}$$

  where k is proportional to the average value of the frontal sensors measurement (the robot’s sensor V_3 and V_4 according to Fig. 2) ($0 \leq k \leq \gamma$, with γ a defaulted value); i is the absolute value of the difference among the two node-motor activities, representing the deviation angle expressed in radians. Therefore, a deviation is valid if it does not exceed 1 radian. The first component of (1) is maximized according to the proximity to a light source, while the second component is maximized when direct movements to the goal are generated.

- *Obstacle avoidance:* the robot’s skill to avoid obstacles, when going towards a particular point. The fitness measure is associated partially with Nolfi´s work [5]. The fitness function adopted Φ_2 is:

$$\Phi_2 = z \cdot \left(1 - \sqrt{\Delta z}\right) \cdot (1 - j) \tag{2}$$

where z is the difference between the output value of m_1 and m_2 (see Fig. 1). z expresses the deviation angle of the robot's trajectory in radians ($-2 \leq z \leq 2$); Δz is the absolute value of the algebraic difference between the node-motor activations m_1 and m_2, maximizing ($1-\sqrt{\Delta z}$) when both activations are equal. The term $\sqrt{\Delta z}$ enhances small differences of the motor-nodes; *(1-j)* is the difference between the maximum activation (value *1*) and the maximum sensed value *j* of the proximity infra-red sensors.

- *Wall following*: robot's abilities to follow a wall. These are intended to complement other reactive behaviors in narrow spaces and corridors. The basic objective of wall seeking (wall-following generalization track) is the generation of a trajectory parallel to a wall, or eventually an obstacle. The fitness score used is:

$$\Phi_3 = \begin{cases} \left(1-|\Theta - V_8|\right)+\left(1-|\Theta - V_2|\right) \big/ 3 & \text{if} \quad \text{left} \\ \left(1-|\Theta - V_7|\right)+\left(1-|\Theta - V_5|\right) \big/ 3 & \text{if} \quad \text{right} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

  where Θ is the minimum distance allowable between each sensor and a wall/obstacle. In all simulations a value of Θ=0.3 was adopted.

- *Learning:* the main references for the learning behavior implementation and analysis were: [5][9]. The behavior consists of the robot's approach to one of two possible light sources (targets). In half of the tests (learning stage), the objective to be reached varies without any predefined pattern. The robot does not know "a priori" which light source should reach at the beginning of each test. Therefore, it is expected that the robot learns to discriminate its objective in a trial and error paradigm. Reinforcement learning (φ) is accomplished using the following score:

$$\varphi = \begin{cases} \delta & \text{if the goal reached is right} \\ -\delta & \text{if the goal reached is wrong} \\ 0 & \text{else} \end{cases} \tag{4}$$

  where δ is a default value, being 2 for the proposed experiment. The aim is to maximize the number of times that the robot reaches the right objective in an obstacle-free environment. The fitness function is based on Togelius' proposal [9] with minimum variations:

$$\Phi_4 = \begin{cases} \sum_{i=0}^{n=200} \varphi_i & \text{if } \sum_{i=0}^{n=200} \varphi_i \geq 1 \\ \max\left(V_j\right) & \text{else} \end{cases} \tag{5}$$

  where V_f refers to the sensed value for the f^{th} sensor ($1 \leq f \leq 8$).

The nets' inputs for each configuration depend on the kind of test to carry out and the nature of the controller employed [3]. The above mentioned simple behaviors are combined for the implementation of more complex tasks, such as:

- *Conditional phototaxis:* A variant of simple phototaxis is defined when the robot can move towards one of several light sources, selecting the right goal. This is indicated by an external input (conditional variable) [9].

- *Navigation:* Inspired by different works (e.g., [5] and [15]), it is proposed to analyze whether the robot approaches to a certain source in a small closed environment avoiding obstacles. This means that the whole robot behavior is associated by one part with the goal detection and by the other part with obstacle avoidance, as simpler behaviors.

- *Conditional learning (variable goal):* The behavior associated with this task is linked to learning, but incorporating a conditional variable input, as in conditional phototaxis. The environment does not have obstacles.

This particular task selection is aimed to compare results with other available works in the literature. They present enough complexity to be studied in the outlined context (see [5][9][16]).

2.5 Coordination of Behavioral Levels

For the development of coordination among behaviors a hierarchical Subsuption Architecture [17] (see [3]) was selected. Basically, it comprises an evolutionary FFNN fed by the behavioral modules outputs and robot sensors [1]. Finally, the outputs of the coordination module control the actuators. The fitness score adopted is:

$$\Phi_{coord} = \Theta_A(1-\Theta_B) \tag{6}$$

where Θ_A is the maximum of all light sensors and Θ_B is the maximum of all proximity sensors. The coordination network is easy to evolve, and can be replaced by a simple rule-based system, a very interesting property of the approach proposed for the project[12].

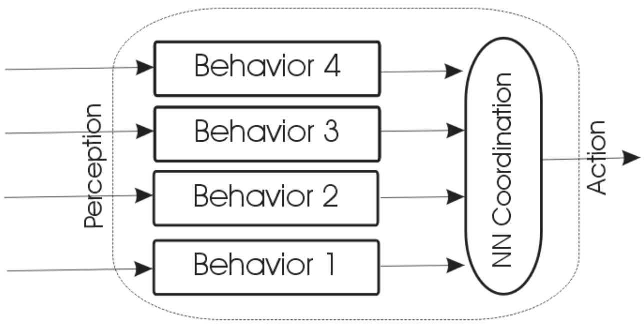

Fig. 2. Connectivity of the neuro-controllers in Subsumption Architecture.

The system doesn’t need to be homogeneous: a neural network can be combined with a conventional controller, an expert system, etc., without major modifications or retraining.

3 Simulation and Experimental Results

In a previous work [3], neural controllers for each of the basic behaviors stated in *section 2.4* were developed, using monolithic evolution. Results for five types of neuro-controllers of different net-topologies were discussed. These results are summarized here for comparison with the proposed LE approach. During the generational development, the relationships among the internal components of each controller were adapted to the desired behavior, increasing the complexity of their behavioral patterns in most cases. Specialization arises as a consequence of reactions to the environment.

Depending on network topology, this occurs in the generational process as well as during the phenotype's lifetime. Regarding the abilities of each network topology to develop the selected behaviors, the most successful in general was FFNN (in conditional phototaxis and obstacle avoidance), followed by HPNN (in learning), see Table III. Performance of the remaining architectures was significantly lower. This can be seen in Fig. 4, where fitness levels of the different neural topologies are shown. It is attractive to analyze if an incremental strategy like LE can reach similar results than more conventional approaches, like those cited in *section 2.5*.

One of the advantages of LE over monolithic evolution to obtain a neuro-controller is a significant reduction on evolution time, due to the smaller size of the networks required on each layer. In the experiments presented on this paper, this benefit was evident. An additional advantage was that behaviors were preserved without the risk of "unlearning".

3.1 LE simulation results

A layered strategy was devised using a simplified simulated model of the Khepera© robot and its environment. Considering the results of section 2 A, a FFNN topology was used for each layer. First, a bottom layer of *Conditional Phototaxis* was generated, followed by an *Obstacle Avoidance* layer, a *Wall Following* layer and finally a *Conditional Learning* (upper) layer, using the methodology described in *section 2*. The emergent behaviors were developed successfully, and the quality of the solutions was, at least, equivalent to those obtained with the monolithic approach in the developed tasks. Tests shown that FFNN topologies outperform more "adaptive" architectures suggests that basic behaviors are developed in the evolution phase, instead of during controller's life, as happens with HPNN. It was found that the FFNN and HPNN controller implementation in the learning level presents a similar final behavior.

Lately, these controllers are tested on a detailed simulator of the Khepera© while robot moves on a labyrinth-like environment. As the available real robot does not have any sensors to recognize different light sources, the conditional learning layer was omitted.

In other words, general reactive navigation strategy was tested under a complex environment, having into account several kinds of obstacles and safety issues. Final controller's robustness was analyzed based on sensors imperfection (i.e. Robustness to environmental and morphologic variations)[3]. The experiments carried out shown that the controller was able to beat environmental changes, but couldn't overcame morphologic ones in most experimental tests.

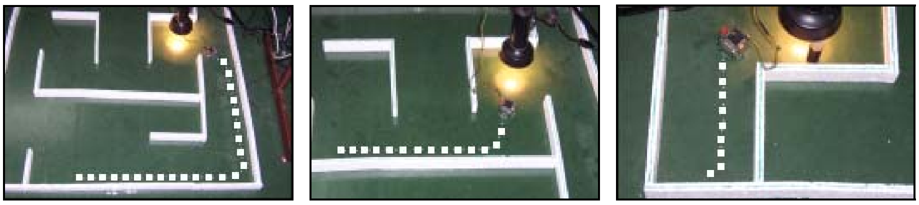

Fig. 3. Multilevel controller samples in real experiments

3.2 Experimental results in real environments

Controller presented at section 3.1 was kept in order to be used at a real robot Khepera©. The environment was represented by a physic labyrinth like the one used at simulated stage. It consisted of a 105x105 cm. arena delimited by walls, with different internal walls of white plastic and a compact green rubber floor (*fig 2*). A light source equipped with a bulb of 75 Watts was placed perpendicularly to the floor of the arena at approximately 3 cm. high. The test room presented other non-controlled light sources, such as fluorescent lights (zenithal) and natural light from windows.

Fig. 3 shows results of three independent experimental situations, requiring avoidance of concave and convex obstacles, phototaxis, and wall seeking behaviors. The robot is shown in its final position after each movement in the labyrinth. The agent was capable of performing the appropriate tasks in a partially unknown environment. The white squares indicate the path followed by the robot in each case.

The results show that the combination between behaviors generated through simulated evolution is a feasible strategy that gives rise to emergent behaviors, and controllers that can solve real-world problems efficiently. From a pure evolutionary perspective, however, it must be noted that the methodology presented is too much dependent on expert knowledge, with evolution taking place in a rigid, prescribed framework. Subdivision in atomic tasks, individual fitness functions and coordination rules are user-guided, leaving small chance for self-organization and feature discovering. Scalability in the general case is still an open subject, having into account successful results reached in this article.

4 Discussion, Perspectives and Tendencies

As previously expressed, LE focus implies human intervention mainly in the coordination of behavioral modules. Fitness scores used for each individual behavior come from expertise knowledge, like as networks size (fixed), being restrictive if a pure evolutionist focus is considered, but does not inside an engineer one. However, the main idea of this job was to compare results of applying LE based controllers to monolithic ones which was able to be carried out (see [3]).

After comparing experiments using module evolutionary coordination to rule based coordination, it is possible to say that generated results were quite similar. Probably much the rule based coordination as evolutionary one was suitable to generate the desired behavior.

A possible solution to synthesize behaviors, as done in this article, is to let evolutive process to potentially discover rules in a progressive way, and be able to generate and modify them through a random variation process, being refined during controller lifetime as an adaptative process part. So that lets several emergent properties to be kept and maintained without identify the relationships that control the behaviors.

Therefore it is valid to guess: While in classic design methods are not simple the full features inference that arise from agent-environment interaction (see [5]), the evolutive focus makes that this identification possible in an emergent way.

5 Contributions and conclusions

LE is a technique that allows controller generation in ER. This focus is able to be performed and its performance is so appropriated like one gotten in more traditional ways (i.e. monolithic evolution). This permits to scale in a structure in a consistent way. LE is useful in complex situations such as navigation in unknown environment. Although, it inherits some rigid features from Subsumption Architecture, which depends more on designer abilities that on emergency of process and pure emergent behavior coordination. This implies pre-imposed rules which let speculate if the behavior emergency really exists in LE.

From designer perspective, the described perspective enables an incremental development and a clear way to generate complex controller in engineering. It is reasonable to say that a major analysis should be done to formalize stability and performance issues related to scalability, being nowadays a vanguard topic. Finally, one of the most contribution described here (and [3]), was to develop an incremental study of adaptative-evolutive control for the project [12] and to generate practical contributions inside the ER area.

Acknowledgement. This work (as like [3]) was developed with initial support of CONICET, Argentina, Scholarship CONICET Res. 38 - 08/01/2004, and finally with the support of the Programa Alβan, Unión Europea, ID code E05D059829AR.

References

1. Fernández León, J.; Acosta, G.; Mayosky, M. An Exp. Study of Scaling Up in Mobile Robot Motion Control using Evolutive Neuro-controllers. IEEE T-RO. In evaluation. (2005).
2. Fernández León, J. A.; et al. An Experimental Study on Evolutionary Reactive Behaviors for Mobile Robots Navigation. Journal JCS&T, Vol. 5, 4, pp. 183-188 (Invited Paper). (2005).
3. Fernández León, J. A. Estudio de Neuro-Controladores Evolutivos para Navegación de Robots Autónomos. Master en Ingeniería de Sistemas. UNCPBA, Argentina. Dir.: G.G. Acosta; Co-Dir.: M. A. Mayosky. Marzo (2006). Web: http://www.exa.unicen.edu.ar/~jleon/
4. Abbott, L. F.; Regehr W. G. Synaptic Computation. Nature, Vol. 431, pp. 796-803. (2004).
5. Nolfi S.; Floreano, D. Evolutionary Robotics. MA: MIT Press/Bradford Books. (2000).
6. Fernández León, J. A.; Tosini, M. A.; Acosta, G. Evolutionary Reactive Behavior for Mobile Robots Navigation. 2004 IEEE Conf. on Cyb. and Int. Syst. (CIS), pp. 532-537. (2004).
7. Maaref, H.; Barref, C. Sensor-based navigation of a mobile robot in an indoor environment. Robotics and Autonomous Systems, 38, pp. 1-18. Elsevier. (2002).
8. Nelson, A. L.; Grant, E.; Galeotti, J. M.; Rhody, S. Maze exploration behaviors using an integrated evolutionary robotic environment. Rob. and Aut. Syst., 46, pp. 159-173. (2004).
9. Togelius, J. Evolution of the Layers in a Subsumption Architecture Robot Controller. Master of Science in Evol. and Adaptive Syst. University of Sussex, UK. (2003).
10. Brooks, R. A. A robust layered control system for a mobile robot. IEEE Journal on Rob. and Aut., 2, 14-23. (1986).
11. K-Team. Khepera User Manual, Version 5.02. Lausanne. Swiss. (1999).
12. Acosta, G. G. Proyecto 03027 – AUVI (Autonomous Underwater Vehicle for Inspections), 6th European Framework Programme for Research and Development, UE. (2005).

13. Hoppenot, P.; Benreguieg, M.; Maaref, H.; et al. Control of a medical aid mobile robot based on fuzzy navigation. IEEE/IMACS Int. Conf., CESA'96, pp. 388-393. (1996).
14. Harvey, I. The SAGA cross. The mechanics of recombination for species with variable length genotypes. Manner et al. (Eds.), Parallel Prob. Sol. from Nature 2. (1992).
15. Togelius, J. Evolution of a Subsumption Architecture Neurocontroller. Journal of Intelligent & Fuzzy Systems, 15:1, pp. 15-21. (2004).
16. Beer, R. D. Towards the evolution of dynamical neural networks for minimally cognitive behavior. 4th Int. Conf. on Sim. of Adapt. Behavior, pp. 421-429. MIT Press. (1996).
17. Arkin, R. C. Behavior-based Robotics. MIT Press, Cambridge, MA. (1998).