Adaptive Behavior Through a Darwinist Machine

F. Bellas, A. Lamas, R. J. Duro

Grupo de Sistemas Autónomos Universidade da Coruña, Spain
gsa@cdf.udc.es

Abstract. In this paper we propose a mechanism based on Darwinist principles applied on-line that provides organisms with the capability of adapting through the use of their interaction with their surroundings in order to improve the level of satisfaction obtained. The mechanism involves a two level concurrent operation of evolutionary processes. The processing carried out in the first, or unconscious level, leads to a current, or conscious model of the world and the organism, which is employed for evaluating candidate strategies, using as fitness the predicted motivation satisfaction.

1 Introduction

For real autonomy and adaptability to emerge, artificial agents must be endowed with cognitive mechanisms whose operation and adaptability to changing environments, requirements and tasks are not predetermined by the designer. This allows the artificial agent to organize its own knowledge from its own perceptions and needs.

In this paper we are going to consider a model, whose foundations were laid in [6], based on some not too well known theories within the field of cognitive science that relate the brain or neural structure with its operation through evolutionary processes in somatic time. These theories are: the Theory of Evolutionary Learning Circuits (TELC) [1]; the Theory of Selective Stabilization of Synapses (TSSS) [2]; the Theory of Selective Stabilization of Pre-Representations (TSSP) [3] and the Theory of Neuronal Group Selection (TNGS) or “Neural Darwinism” [4]. For an excellent review see [5]. Each theory has its own features but they all lead to the same concept of cognitive structure and their main ideas are that the brain adapts its neural connections in real time through evolutionary or selectionist processes and this somatic selection determines the connection between brain structure and function. As a reference of work in a similar direction we must cite Nordin et al. [7].

2 Cognitive Mechanism

The final objective of any cognitive mechanism is to link what the organism perceives and what it has done to what it must do in order to satisfy its motivations. In this line, a cognitive model must be given by three basic elements: *strategies, world models* and *internal models*. A *strategy* is just a sequence of actions applied to the effectors. An ideal *strategy* would depend on the current perceptions, internal state and motivations. On the other hand, a *world model* is a function that relates the sensory inputs of the agent in the instant of time t with the sensory inputs in t+1 after applying a *strategy*. A *world model* permits evaluating the consequences of actions on the environment as perceived by the agent (sensed values in $t+1$). Finally, an *internal model* is a function that relates the sensorial inputs t+1 with the internal state according to the motivation for the agent.

The problem is how to connect these three elements into an on line mechanism that permits the agent to generate *original solutions* making use of *previous experience* and minimizes *designer intervention* so that the motivations are fulfilled.

The way we have solved the problem is shown in Fig. 1. The final objective of the mechanism is to obtain the strategy the agent must execute in the real world to satisfy its motivations. This strategy is the *Current Strategy* and is applied to the *Environment* through the *Effectors* providing new *Sensing* values for the agent. Thus, after each iteration, we have a new action-perception pair obtained from the real world. These action-perception pairs are stored in the *Short-Term Memory* and are used as the fitness function for the evolution of the world and internal models.

We have three evolutionary processes in our mechanism (Fig. 1). A *World Model Evolver* manages world models through evolution. This structure evolves a population of world models and each moment of time selects the best one according to the information it has available about the real world. The *Internal Model Evolver* manages the internal model base in a similar manner. Finally, a *Strategy Evolver* that evolves strategies and when the agent needs one it provides it.

Each one of these evolutionary processes starts from the population stored in a memory (*World Model Memory, Internal Model Memory and Strategy Memory*). The contents of these memories are random for the first iteration of the mechanism. The individual with the highest fitness value after evolution according to the information of the short-term memory is selected as the current model. In the case of strategies these are tested during their evolution by implementing a virtual environment using the current models, that is, the current world model provides the sensory inputs in instant t+1 for the strategy we want to test. The current internal model uses these predicted values as input to predict the internal state resulting from the strategy. The strategy with the highest fitness value is selected as the *Current Strategy* and is applied to the *Environment*. This basic cycle is repeated and, as time progresses, the models become better adapted to the real world and the selected strategies improve.

The controllers these processes handle are Artificial Neural Networks, whose parameters constitute the "chromosomes" of the genetic algorithms, although any other computational structure could be used.

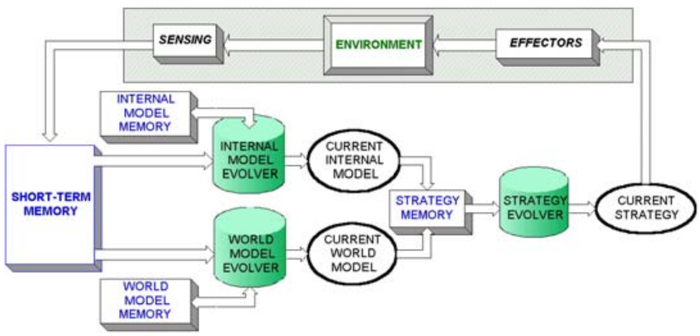

Fig. 1. Block Diagram of the cognitive mechanism where the shadowed area represents the external operation of the mechanism and the remaining area represents the internal operation.

3 Application examples

We have considered two examples. The first one is a three legged robot with two light sensors in a world with a light that moves up and down in front of it. To be satisfied, the robot must follow the light as closely as possible until its “light reservoir” is filled. The world model and internal model memories are made up of populations of artificial neural networks. The robot must learn to reach the light but initially the models are not adequate as they have not undergone enough evolution. Consequently, the strategies applied are wrong and the robot doesn’t follow the light (area A of Fig. 2). As time progresses, the models improve (area B of Fig. 2) until a point is reached where the robot follows the light closely. Once the models are good enough, the robot follows the light perfectly. In order to test the adaptability of the mechanism at this point we change the motivation and now the light hurts and the robot must escape from it. This is shown in the areas C and D of Fig. 2. It is important to see how a change of motivation implies an immediate change in the selected strategies.

As a second example, closer to reality, we have implemented the mechanism on a Hermes II hexapod robot. For the sake of processing power, we ran the mechanism on a 3-D simulation and then transferred it to the real robot. In this case, we have two different tasks, consisting in the robot learning to walk and reaching a tree like object.

First, the robot must learn to walk and the world models have one input corresponding to the distance to the tree (using a virtual sensor from the IR information) and 6 inputs corresponding to the relative phases of each leg. The output was the predicted distance. This way, the mechanism finds that the best gait (combination of phases) in order to advance quickly to the object is similar to a tripod gait. On the left area of figure 3 we represent the simulator and the range of the IR sensors as the robot walks, on the right we see the real robot. As initially the robot had no clue on how to walk or what to do to fulfill its motivations, it started moving its legs randomly producing all kinds of weird gaits until it managed to obtain some gaits that produced motion and thus provided information for the world models and internal models to improve.

Once the robot has obtained an appropriate gait, we change the problem and now the robot must turn, controlling the amplitude of leg motion, with the objective of

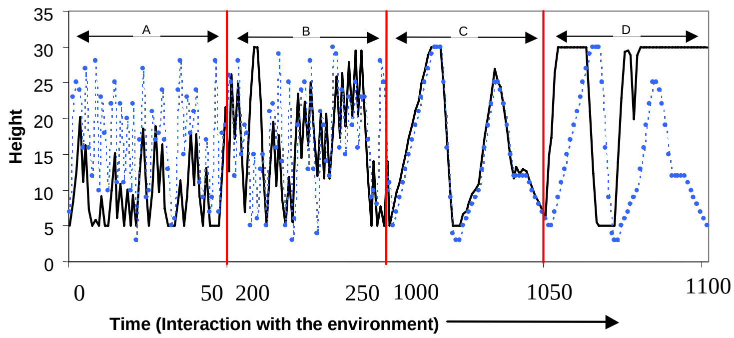

Fig. 2. In the first stages of the mechanism (area A) the robot (dotted line) doesn’t follow the light (continuous line), but a few iterations later (area B) the behavior improves a little. In area C the robot follows the light perfectly, and in area D the goal is changed to avoiding the light.

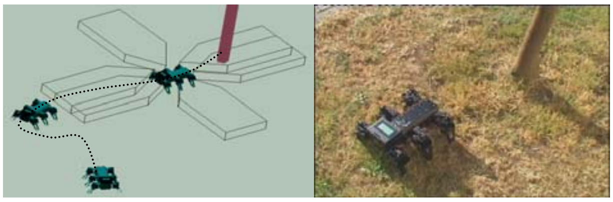

Fig. 3. The Hermes robot develops a gait and strategy to reach the tree-like object.

reaching the tree. The world model now has two sensory inputs (angle and distance to the tree), one amplitude input and two outputs (the predicted perception). The results are good after 200 iterations and the robot reaches the tree (left side of figure 3).

4 Conclusions

An evolutionary learning mechanism for artificial organisms to autonomously learn to interact with their environment without their behavior being completely predetermined has been presented. This structure permits artificial organisms to think things out before acting and provides a framework for learning to take place. It is important to note that during this thinking process, the agent is able to generate original solutions (strategies) the organism has never used before and it is able to include previous experience through evolutionary processes that allow new solutions to arise and, at the same time, permit meaningful combination of previously acquired experience in the form of world or internal models or successful strategies.

Acknowledgements

This work was supported by the FEDER through project PROTECAS N. 1FD1997-1262 MAR. and by the MCYT of Spain through project TIC2000-0739C0404.

References

1. Conrad, M.: Evolutionary Learning Circuits. J. Theor. Biol. 46, 1974, pp. 167.
2. Changeux, J., and Danchin, A.: Selective Stabilization of Developing Synapsis as a Mechanism for Specification of Neural Networks. Nature 264, 1976, pp. 705-712.
3. Changeux, J., Heidmann, T., and Patte, P.: Learning by Selection. The Biology of Learning. P. Marler and H.S. Terrace (Eds.), Springer-Verlag. 1984, pp. 115-133,
4. Edelman, G.M.: Neural Darwinism. The Theory of Neuronal Group Selection. Basic Books 1987.
5. Weiss, G.: Neural Networks and Evolutionary Computation. Part II: Hybrid Approaches on the Neurosciences. Proceedings of the First IEEE Conference on Evolutionary Computation, Vol. 1, 1994, pp. 273-277.
6. F. Prieto, R. J. Duro, J. Santos: Modelo de Mecanismo Mental Darwinista para robots móviles autónomos. Proceedings of SAAEI'96. Zaragoza, Spain, 1996
7. Nordin, P., Banzhaf, W. and Brameier, M.: Evolution of a World Model for a Miniature Robot Using Genetic Programming. Robotics and Autonomous Systems, Vol. 25, 1998, pp. 105-116.