# Induced behavior in a real agent using the Multilevel Darwinist Brain

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

Grupo de Sistemas Autónomos
Universidade da Coruña, Spain
fran@udc.es, ronin@udc.es, richard@udc.es

**Abstract.** In this paper we present a strategy for inducing a behavior in a real agent through a learning process with a human teacher. The agent creates internal models extracting information from the consequences of the actions it must carry out, and not just learning the task itself. The mechanism that permits this background learning process is the Multilevel Darwinist Brain, a cognitive mechanism that allows an autonomous agent to decide the actions it must apply in its environment in order to fulfill its motivations. It is a reinforcement based mechanism that uses evolutionary techniques to perform the on line learning of the models.

## 1 Introduction

When trying to deal with the problem of teaching a task to a robot through real time interaction, there are lots of variables that must be taken into account. In fact, the problem is normally driven by the designer and it is difficult to determine what has been autonomously learned and what has been induced by the limitations of the teaching process. As mentioned in [1] the representation reflects to a large extent the human designer's understanding of the robot task. In order to generalize the solutions obtained, the automatic design of robot controllers has become a very important field in autonomous robotics.

A natural approach to this problem is learning by demonstration (LBD) [2], but other methods such as gestures [3] or natural language [4] are also techniques that have been successfully applied. In LBD authors have tried to separate "how" to carry out a task from "what" to do in a task, that is, what the task is, focusing their research on systems that could be applied to different tasks. Within this field of learning by demonstration, relevant results have been achieved based on training and demonstration stages as in [5]. A technique that has been demonstrated as very appropriate in the automatic development of controllers is reinforcement learning [6], [7] where a robot (any agent in general) learns from rewards and punishments (which are basically some type of error signal) provided by a teacher in real time. These systems can be applied to different tasks (different "what") but they don't work if, for example, something fails in the robot (different "how").

A new approach to the automatic design process is called cognitive developmental robotics (CDR). As explained in [8], the key aspect of CDR is that

the control structure should reflect the robot's own process of understanding through interactions with the environment.

In this work we present a solution in which the main objective is to complete a task driven by a motivation and not by the way the goal is achieved. Thus, if the environment or the agent itself changes, the way the task is carried out may also be modified but the agent readapts autonomously. The cognitive mechanism we have developed is called the Multilevel Darwinist Brain and in the next section we are going to provide an introduction of its concepts and operation.

## 2 The Multilevel Darwinist Brain

The Multilevel Darwinist Brain (MDB) is a Cognitive Mechanism that allows a general autonomous agent to decide the actions it must apply in its environment in order to fulfill its motivations. The main feature of the MDB is that the acquisition of knowledge is automatic, this is, the designers do not impose their knowledge on the system. In its development, we have resorted to biopsychological theories [9], [10], [11] within the field of cognitive science that relate the brain and its operation through a Darwinist process. Each theory has its own features, as shown in the references, but they all lead to the same concept of cognitive structure based on the brain adapting its neural connections in real time through evolutionary or selectionist processes. This idea of Darwinism in the acquisition of knowledge is the basis for the development of the practical Cognitive Mechanism proposed.

### 2.1 Cognitive Model

To implement the MDB, we have used an utilitarian cognitive model [12] which starts from the premise that to carry out any task, a *motivation* (defined as the need or desire that makes an agent act) must exist that guides the behavior as a function of its degree of *satisfaction*. From this basic idea, several concepts arise:

- *Action*: set of commands applied to the actuators of the agent.
- *World model (W):* function that relates sensory inputs in time t with sensory inputs in time t+1 after applying an action.
- *Satisfaction model (S):* function that relates sensory inputs with the satisfaction the agent should perceive in the same instant of time according to the motivation for the agent.
- *Action-perception pair:* set of values made up by the sensorial inputs and the satisfaction obtained after the execution of an action in the real world. It is used when perfecting world and satisfaction models.

Two processes must take place in a real non preconditioned cognitive mechanism: models $W$ and $S$ must be obtained as the agent interacts with the world, and the best possible actions must be selected through some sort of internal optimization process using the models available at that time.

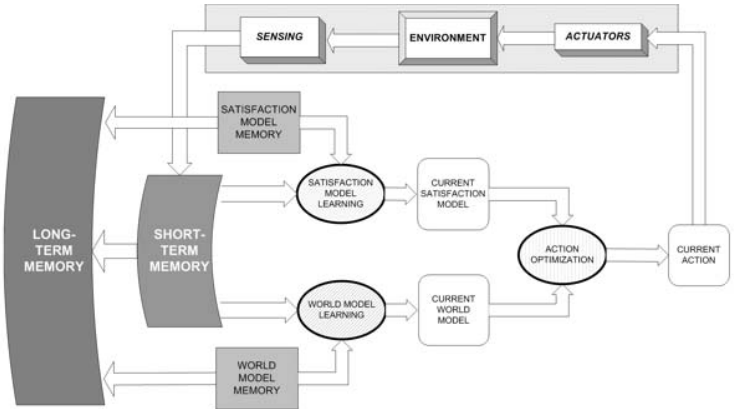


**Fig. 1.** Block diagram of the MDB

### 2.2 Constructing the MDB

The main difference of the MDB with respect to other model based cognitive mechanisms is the way the models are obtained and the actions planned from them. Its functional structure is shown in Fig. 1. The main operation can be summarized by considering that the selected action (represented by the *current action* block) is applied to the *environment* through the *actuators* obtaining new *sensing* values. These acting and sensing values provide a new action-perception pair that is stored in the action-perception memory (*Short-Term Memory* from this point forward). Then, the *model learning* processes start (for world and satisfaction models) trying to find functions that generalize the real samples (action-perception pairs) stored in the *Short-Term Memory*. The best models in a given instant of time are taken as *current world model* and *current satisfaction model* and are used in the process of *optimizing the action* with regards to the predicted satisfaction of the motivation. After this process finishes, the best action obtained is applied again to the *environment* through the *actuators* obtaining new *sensing* values.

These steps constitute the basic operation cycle of the MDB, and we will call it an iteration of the mechanism. As more iterations take place, the MDB acquires more information from the real environment (new action-perception pairs) so the models obtained become more accurate and, consequently, the action chosen using these models is more appropriate.

The block labeled *Long-Term Memory* stores those models that have provided succesful and stable results on their application to a given task in order to be reused directly in other problems or as seeds for new learning processes. Details about implementing the *Long-Term Memory* can be found in [13].

Summarizing, there are two main processes that must be solved in MDB: the learning of the best models predicting the contents of the short-term memory and the optimization of the action trying to maximize the satisfaction using the

previously obtained models. In the way these processes are carried out lies the main difference of the MDB with respect to other mechanisms.

### 2.3 Learning of models and action optimization

The model search process in the MDB is not an optimization process but a learning process (we seek the best generalization, which is different from minimizing an error function in a given instant t). Consequently, the search techniques must allow for gradual application as the information is known progressively and in real time. In addition, they must support a learning process through input/output pairs using an error function. To satisfy these requirements we have selected *Artificial Neural Networks* as the mathematical representation for the models and *Evolutionary Algorithms* as the most appropriate search technique.

Evolutionary techniques permit a gradual learning process by controlling the number of generations of evolution for a given content of the short-term memory. Thus, if evolutions last just a few generations per iteration, gradual learning by all the individuals is achieved. To obtain a general model, the populations of the evolutionary algorithms are maintained between iterations. Furthermore, the learning process takes place through input/output pairs using as fitness function the error between the predicted values provided by the models and the real values in each action-perception pair.

Strongly related to this process is the management of the Short Term Memory. The quality of the learning process depends on the data stored in this memory and the way it changes. The data stored in this memory (samples of the real world) are acquired in real time as the system interacts with the environment and, obviously, it is not practical or even useful, to store all the samples acquired in agent's lifetime. We have developed a replacement strategy for this memory that permits storing the most relevant samples for the best possible modelling. Details can be found in [12].

The search for the best action in the MDB is not a learning process because we are looking for the best possible action for a given set of conditions. That is, we must obtain the action whose predicted consequences given by the world model and satisfaction model result in the best predicted satisfaction. Consequently, for the actions, a simple optimization problem must be solved for which any optimization technique is valid.

The operation of the basic mechanism we have presented has been tested in real agents [12], so the objective of this paper is not to show the operation of the mechanism in terms of evolution of the models, action optimization or STM managment, but to study how it operates in a conceptually higher level problem.

## 3 Induced behavior in a real robot

As commented before, the operation of the MDB is based on an external process of interaction with the world where the action-perception pairs are obtained, and an internal process where models are updated and actions selected from the

current models. Both of these processes occur concurrently on the different world and satisfaction models present in the model bases. Consequently, even though, the motivation structure we have established leads the robot to maximize its "real" satisfaction in terms of rewards obtained from the teacher, internally, several satisfaction models may arise through interaction with the world that relate different sensory patterns to expected satisfactions. An example of this is the real satisfaction (teacher rewards) that can be directly modeled in terms of the relationship between actions and sounds for a given communication language through a world model that provides error values and a satisfaction model that provides the expected satisfaction. But the real satisfaction can also be modeled (if the teacher is consistent in its commands and rewards) in terms of the perceived distance decrease to the objective when performing an action (world model) and the related predicted satisfaction (satisfaction model). Obviously, if the teacher was consistent, any of these satisfaction models lead the robot to performing the same behavior, although they are not really modeling the same. One of them models the satisfaction obtained from following the teacher´s commands and has nothing to do with the task the robot is performing. The other does not take into account the teacher´s commands, but generates high satisfaction values when the behavior corresponds to the one the teacher wanted (reaching the object). This is what we call cross-induced behavior. Thus, if we take into account that any satisfaction model is only applicable when its inputs are present, and that the real satisfaction is obtained through rewards of the teacher, the MDB will use as satisfaction model the first one of these two when there is input from the teacher and when there is no input it will resort to the second one, that is, the induced satisfaction model.

Typical induced behaviors are based on the generation of a behavior pattern that arises from the task that is directly taught. In this example, the task is very simple: we want that a real agent (a robot) to reach an object. In the learning stage it must follow the commands that a teacher provides in order to reach the object. This teacher rewards or punishes the robot depending on if the action applied agrees with the command or not. The induced behavior we try to obtain appears from the fact that each time the robot applies the correct action, the distance to the object decreases. This way, once the teacher disappears, the robot can continue with the task because it just has to perform actions that reduce the distance to the object.

To carry out this example, we have designed the following experiment:

- First of all, the real agent we are going to use is a wheeled robot called Pioneer 2. It has a sonar sensor array as its main sensorial system and all the computation is carried out in a laptop that is placed on top of the robot.
- We place the robot in the central point of a simple real environment and the object it must detect in an area around the robot. In Fig. 2 (left) we display the experimental setup including the real robot and object.
- A teacher, external to the MDB, must make the robot reach the object frontally through a set of commands that guide its movements. Possible commands have been reduced to a group of seven which represent the se-

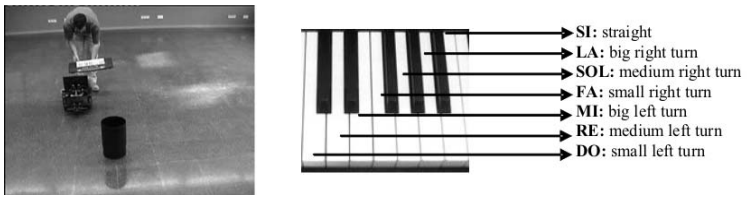


**Fig. 2.** Experimental setup (left) and translation of commands into musical notes (right)

mantics of the language used by the teacher to communicate with the robot. The sensor we use for communication purposes in the robot is a microphone, and the commands have been translated into musical notes. A possible translation is found in Fig. 2 (right) but it is not pre-established and we want the teacher to make use of any correspondence it wants.

- After perceiving a command, the robot can apply one of the following seven actions: 1-No turn, 2-Small right turn, 3-Medium right turn, 4-Large right turn, 5-Small left turn, 6-Medium left turn, 7-Large left turn. As we can see, the actions are in accordance with the possible commands.
- Depending on the degree of fulfilment of a command, the teacher must reward or punish the robot. To do this, we use a numerical value as a pain or pleasure signal that is introduced through a keyboard.

In figure 3 we display a schematic view of the two world model satisfaction model combinations that arise in this experiment. The first world model has two inputs (Fig. 3 top): command provided by the teacher and action applied; and one output: the predicted human feedback. As we can see, this model is related with robot-teacher communication, so we will call it *communication model* from this point forward. In this case, the satisfaction model is trivial because the satisfaction coincides with the output of the communication model, this is, the reward or punishment.

In the stage where the teacher is present, the communication model is the one used to select the action the robot must apply, while other world and satisfaction models are learnt in the background using the information provided by the action-perception pairs obtained by the robot through the interaction with the teacher and the environment. The second world model used has 4 inputs (Fig. 3 bottom): distance and angle to the object, continuity bit (necessary because of the symmetry of the sensed angles in this kind of circular robots) and the applied action; and 3 outputs: distance, angle to the object and continuity bit predicted after applying an action. The satisfaction model has two inputs: relative distance covered and angle after applying an action; and one output: the satisfaction value. The two operation modes are directly derived from the teaching process, because when the teacher is present the MDB can use direct sensorial information from the teacher (commands and rewards) while in the operation without teacher the MDB can use just the sensorial information obtained by the

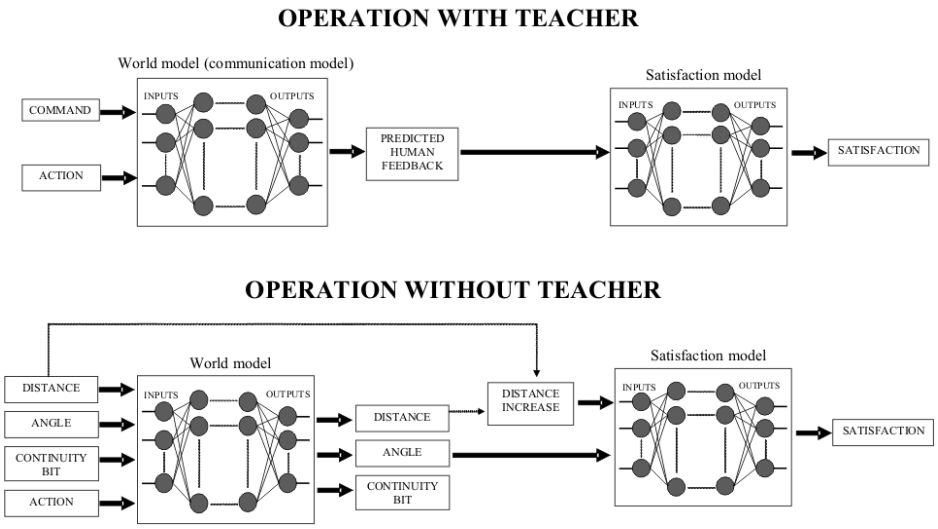


**Fig. 3.** World and satisfaction models for operation with and without teacher

robot (distance and angle to the object). We can see this operation scheme as divided into two hierachical levels: when the teacher is present, the robot follows its commands and when he has gone it must use its own perceptions.

To evolve the models we have used a promoter based genetic algorithm [14] obtaining succesful results in the modelling of the action-perception pairs with multilayer perceptron artificial neural networks as individuals. The optimization of the action is very simple in this case as a reduced set of possible actions is considered, so we test all in the communication model (or in the world model depending on the operation mode) and select the one that provides the best satisfaction value.

To show the flexibility of this operation scheme, in Fig. 4 we have represented the evolution of the mean squared error provided by the best communication model during 2000 iterations. In the first 500 iterations the teacher provides commands using the encoding (language) shown in Fig. 2 (right). From iteration 500 to iteration 1000 the teacher stops providing commands and the robot uses the world and internal models. From iteration 1000 to iteration 1500 another teacher appears using a different language (different relation between musical notes and commands) and, finally, from iteration 1500 this second teacher dissapears too and the robot must use the world and satisfaction models again. There are no data in the operation without teacher because there are neither commands nor rewards and, consequently, there is no evolution of the communication model. As we can see in the figure, in the first 500 iterations the error decreases fast to below 10% which results in a very accurate prediction of the rewards. Consequently, the robot succesfully follows the commands of the teacher. What is

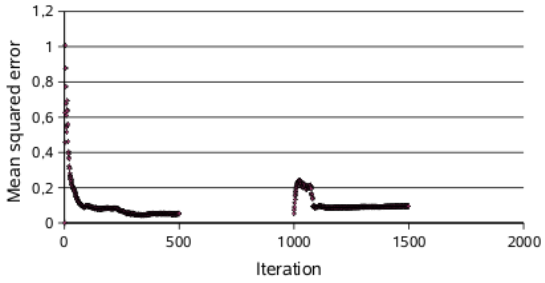


**Fig. 4.** Evolution of the mean squared error provided by the best communication model during 2000 iterations

really interesting in this graph, is that the behavior is the same from iteration 1000 to 1500, this is, when the second teacher appears. The iterations needed to follow his commands are more or less the same as in the previous case (related to the iterations needed to empty the action-perception pairs stored in the STM due to to the first teacher and fill it with the pairs of the second teacher) so the robot learns the new language and follows the commands quickly adapting itself to teacher characteristics. Finally, we want to remark that the learning of the world and internal models occurs in the same way with both teachers because the sensorial data used in these models (Fig. 3 bottom) are independent of the language selected by the teacher.

The main result from this experiment is shown on Fig. 5 that representes the number of object captures as the robot interacts with the world, taking into account that the teacher provides commands from until iteration 500 and from iteration 1000 to 1500. As shown in the figure, from this iteration on, the robot continues capturing the object in the same way the teacher had taught it, so we can say that the induced learning of the models has been successful. The decrease of the slope in the figure implies that the robot takes a large number of iterations (movements) to reach the object using the induced models. This is because it has learnt to decrease its distance to the object and not the fastest way to do it (these models aren't based on obedience but on distance increments). According to Fig. 4, when the second teacher appears there is a brief gap with no captures. In the first 200 iterations there are no captures because we have forced a random exploration stage to store relevant information in the STM. This was necessary due to the complexity of the world model motivated by the way the sonar sensor ring of the Pioneer 2 operates.

Fig. 6 displays a real execution of actions. In the left part, the robot is following commands by a teacher; in the right it is performing the behavior without any commands, just using its induced models. It can be clearly seen that the behaviour is basically the same although a little more inefficient.

Summarizing, this mechanism provides a way to teach a robot to do something where the world models and satisfaction models are separate. Thus, the

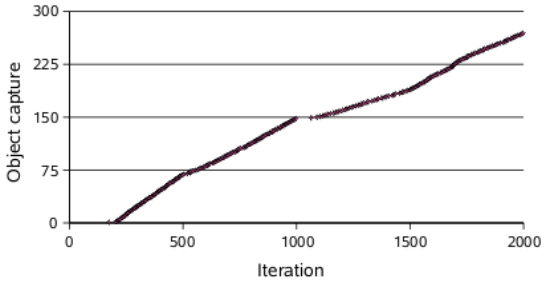


**Fig. 5.** Number of object captures through iterations

satisfaction models determine the final behavior ("what") in terms of motivation, the world models represent the circumstances of the environment and the action optimization strategy will obtain the "how" from this combination. Thus, any teacher can teach any machine any behavior it can carry out through a consistent presentation of sensorial inputs and rewards/punishments and the system will obtain a model of the world and satisfaction with regards to the teacher commands and expected actions as well as any other induced world and satisfaction models that relate other inputs with the expected satisfaction.

## 4 Conclusions

In this paper we have presented an approach, that has been implemented through the Multilevel Darwinist Brain cognitive mechanism, for performing teaching-learning processes between humans and robots in a natural manner. The MDB allows for the induction of satisfaction models from reinforcement or carrot and stick teaching processes which can be used by the system to operate when no teacher is present. In addition, the separation of the different types of models in

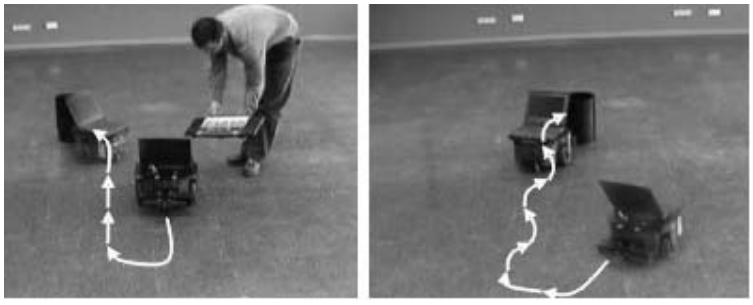

**Fig. 6.** Actions applied by the Pioneer 2 robot in the operation with (left) and without teacher (right)

the architecture provides for a straightforward implementation of the trainable communication models required for the system to be able to learn to understand what the teacher means. An example with a real agent has been presented.

## 5 Acknowledgements

This work was funded by the MCYT of Spain through project VEM2003-20088-C04-01 and Xunta de Galicia through project PGIDIT03TIC16601PR.

## References

1. Weng J., Zhang, Y.: Developmental Robots - A New Paradigm, Proceedings Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems 94, pp. 163-174 (2003).
2. Bakker P., Kuniyoshi, Y.: Robot see, robot do: An overview of robot imitation, Autonomous Systems Section, Electrotechnical Laboratory, Tsukuba Science City, Japan (1996).
3. Voylesl, R. Khosla, P.: A multi-agent system for programming robotic agents by human demonstration. In Proc., AI and Manufacturing Research Planning Workshop (1998).
4. Lauria, S., Bugmann, G., Kyriacou, T., Klein, E.:Mobile robot programming using natural language, Robotics and Autonomous Systems, 38:171–181, (2002).
5. Nicolescu, M., Mataric, M.J.: Natural Methods for Robot Task Learning: Instructive Demonstration, Generalization and Practice, Proceedings, Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp 241-248 (2003).
6. Schaal, S.: Learning from demonstration, Advances in Neural Information Processing Systems 9, pp.1040-1046, (1997).
7. Ullerstam, M.: Teaching Robots Behavior Patterns by Using Reinforcement Learning– How to Raise Pet Robots with a Remote Control, Master's Thesis in Computer Science at the School of Computer Science and Engineering, Royal Institute of Technology (2004).
8. Asada, M., MacDorman, K.F., Ishiguro H., Kuniyoshi, Y.: Cognitive Developmental Robotics As a New Paradigm for the Design of Humanoid Robots. Robotics and Autonomous System, Vol.37, pp.185-193 (2001).
9. Changeux, J., Courrege, P., Danchin, A.: A Theory of the Epigenesis of Neural Networks by Selective Stabilization of Synapses, Proc.Nat. Acad. Sci. USA 70, pp 2974-2978 (1973).
10. Conrad, M.:Evolutionary Learning Circuits. Theor. Biol. 46, pp 167-188 (1974).
11. Edelman, G.: Neural Darwinism. The Theory of Neuronal Group Selection. Basic Books (1987).
12. Bellas, F., Duro, R. J.: Multilevel Darwinist Brain in Robots: Initial Implementation, ICINCO2004 Proceedings Book (vol. 2), pp 25-32 (2004).
13. Bellas, F., Duro, R. J.: Introducing long term memory in an ann based multilevel darwinist brain, Computational methods in neural modeling, Springer-Verlag, pp 590-598 (2003).
14. Bellas, F., Duro, R.J.: Statistically neutral promoter based GA for evolution with dynamic fitness functions, Proceedings of the 2nd iasted international conference, pp 335-340 (2002).