Evolving Internal Reinforcers for an Intrinsically Motivated Reinforcement-Learning Robot

Massimiliano Schembri, Marco Mirolli, Gianluca Baldassarre

Laboratory of Autonomous Robotics and Artificial Life,
Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerche (LARAL-ISTC-CNR),
Via San Martino della Battaglia 44, I-00185 Roma, Italy

{massimiliano.schembri, marco.mirolli, gianluca.baldassarre}@istc.cnr.it

***Abstract* – Intrinsically Motivated Reinforcement Learning (IMRL) has been proposed as a framework within which agents exploit "internal reinforcement" to acquire general-purpose building-block behaviors ("skills") which can be later combined for solving several specific tasks. The architectures so far proposed within this framework are limited in that: (1) they use hardwired "salient events" to form and train skills, and this limits agents' autonomy; (2) they are applicable only to problems with abstract states and actions, as grid-world problems. This paper proposes solutions to these problems in the form of a hierarchical reinforcement-learning architecture that: (1) exploits Evolutionary Robotics techniques so to allow the system to autonomously discover "salient events"; (2) uses neural networks so to allow the system to cope with continuous states and noisy environments. The viability of the proposed approach is demonstrated with a simulated robotic scenario.**

Index Terms – Intrinsically Motivated Reinforcement Learning, Evolutionary Robotics, Actor-Critic, Surprise, Neural Networks.

I. Introduction[1]

Current robots tend to have severe limitations. The main limitation is the fact that they are programmed or evolved for accomplishing only one single task in only one kind of environment. On the contrary, natural organisms are capable of accomplishing many different tasks and can respond to novel challenges posed by the environment by reusing previously acquired general skills. In recent years there has been a growing effort in both the machine learning and the developmental robotic communities to endow robots with a similar flexibility. In this respect, many researchers have proposed that the best way to achieve this goal is to rely on robots' autonomous development [1]: rather than directly programming a behavior for each particular task of interest in robots, one should endow them with developmental programs and allow them to learn, through an autonomous interaction with the environment, general building-block behaviors later "assembled" to tackle several specific tasks.

A number of proposals have been put forward to this purpose, both within the machine learning [2, 3] and the developmental/epigenetic robotics communities [4-6] (see [7] for a brief overview). This paper presents a novel model developed within the Intrinsically Motivated Reinforcement Learning framework (IMRL) [3, 8, 9] that uses ideas and techniques of Evolutionary Robotics (ER) [10] so to overcome two important limitations of current implementations of IMRL. The result is a system which is able to solve different robotic tasks by combining, during 'adulthood', task-general skills acquired, during 'childhood' on the basis of evolved intrinsic reinforcement devices ('reinforcers').

The rest of the paper is organized as follows. Sect. II describes how the model overcomes some limits of IMRL drawing ideas from ER. Sect. III introduces the details of the architecture and of the simulated robotic experiment used to test the model. Section IV reports the main results of the tests. Finally, Sect. V discusses the novelties of the work with respect to previous proposals and illustrates future work.

II. Combining Intrinsically Motivated Reinforcement Learning and Evolutionary Robotics

A. Intrinsically Motivated Reinforcement Learning

The approach proposed here is inspired by the IMRL framework [3, 8, 9], which in turn builds upon psychological theories of motivation [11, 12], recent advances in the neuroscience of reward systems [13-16], and machine learning research on reinforcement learning [17, 18].

The basic idea behind IMRL is that natural organisms, and especially humans, are not driven only by basic motivations directly related to survival (e.g. for eating, drinking, avoiding predation and mating). Rather, they often engage in various forms of exploratory behaviors under the drive of intrinsic motivations [11]. The adaptive value of these intrinsically motivated behaviors seems to lie in *aiding the development of skills which can be subsequently combined* for accomplishing tasks directly related to fitness. The most cited candidates for such intrinsic motivations are novelty, surprise, incongruity, and complexity [12], but other possible sources of internal reward might be envisaged, especially in the case of human beings.

The hypothesis that novelty and surprise might play an important role in organisms' motivational systems has recently found empirical support in the neuroscience literature on reward. In this respect, it was suggested that the phasic release of the neuromodulator dopamine by midbrain neurons serves the function of signaling the occurrence of unpredicted reward, in a way closely similar to the temporal difference prediction error posed by standard reinforcement learning algorithms [13, 14]. Others have proposed that dopamine

[1] This research was supported by the EU Projects ICEA, contract no. FP6-IST-027819-IP, and MindRACES, contract no. FP6-511931-STREP.

release by midbrain neurons might not only signal errors in the prediction of future external rewards, but also [15], or even exclusively [16], the appearance of salient, novel stimuli.

Based on these ideas and recent advancements in machine learning Barto and co-workers have proposed new algorithms for the acquisition of general skills through a developmental process. The architecture used in this work is based on machine learning theory of "options" [18]. Basically, options are sub-routines which can be invoked just like primitive actions, and include: (1) an *initiation set*: the set of states in which the option can be invoked; (2) a *termination condition:* a mapping between states and probabilities of termination of the execution of the option; (3) a *policy*: a mapping between states and actions' probabilities. Within the framework of IMRL an option also contains an *option model*, learned from experience, which maps initiation states to: (a) the probabilities of terminating the option at any other state; (b) the total intrinsic reward obtained while executing the option.

Typically, options are hardwired by the programmer and task-specific. On the contrary, within IMRL the system autonomously develops options on the basis of the occurrence of *novel salient events*. Each time a salient event is detected for which no option is available, an option is created. Each option simultaneously learns both its policy (the option's stimulus-response associations that drive the system to accomplish the option's salient event), and its model (which tries to estimate the probability of occurrence of such an event). A key point here is that the system uses the *prediction error of the option model* as an *intrinsic reward* to decide which option to invoke and train. The effect of this is that until an option is not able to produce its salient event, it will continue to generate internal rewards and hence to be selected and trained. Once trained, the option will stop generating internal rewards and the system will focus on other options.

Although the ideas behind IMRL are very interesting and promising, there are two important drawbacks in its current implementation. First, it assumes high-level representations of states and actions, and in fact so far it has been tested only in abstract, grid-world simple environments. As also clearly recognized in [8], this is a limit because it is not clear how and whether IMRL might be used in embodied agents and robots. Second, "salient events" must be explicitly specified by the programmer. This goes against the IMRL idea of generating agents endowed with fully autonomous developmental programs. In fact, a considerable amount of task knowledge has to be used for putting the algorithm to work: how can in fact 'salient events' be defined, especially in continuous environments in which we cannot assume high-level states but only low-level sensory stimulation?

The model proposed here overcomes both these limitations by integrating IMRL with ER techniques. In particular: (1) it uses *evolved "reinforcers"* for assigning salience to explored states; (2) it uses *neural networks* in order to tackle with continuous and noisy environments such as those encountered in robotic tasks.

B. *Evolving Intrinsic Reinforcers for Simulated Robots*

Evolutionary Robotics [10] is a methodology for building robots and their controllers by artificial evolution. In a typical evolutionary robotic experiment the robot controller is an *artificial neural network* which is evolved through a *genetic algorithm*.

The first major advantage of using evolutionary robotics is that it does not require deep human knowledge about tasks and their possible solutions. By using a genetic algorithm for finding a desired solution, the programmer needs only to specify the general constrains of the control system (i.e. the neural network architecture) and a fitness function for quantifying robot's performance. This leaves the evolutionary process free of exploiting the interactions between the robot and its environment for accomplishing the task: a valuable property with robotic setups which are typically very difficult to manage with direct engineering methods [19].

A second important advantage of ER is that it uses neural-networks to implement robots' controllers. Neural networks not only have a high degree of evolvability [10]; they are also relatively robust with respect to noise and, thanks to their generalization capabilities, they can allow robots to re-use acquired skills in (partially) novel environments and tasks.

Given these desirable properties, the model presented here integrates ER with IMRL to overcome the aforementioned limitations of the latter. In particular, with respect to the original IMRL framework, the model maintains both the idea of having one sub-module for each "salient experience" and the idea of using a "prediction error" as the source of internal reward for the selection and training of sub-modules. However, five main innovations are introduced: (1) options are substituted with neural-network implementations of the actor-critic model [17] named '*experts'* [20]; (2) the action-value function which selects options is substituted by another actor-critic neural model, the '*selector'* [20]; (3) hand-coded salient events are substituted by neural-networks named '*reinforcers'*: each expert has its reinforcer whose connection weights are evolved through a genetic algorithm; (4) the life of the robot is divided into two stages, the "childhood" and "adulthood" respectively, similarly to what is done in [9]; (5) with respect to the intrinsic reward used to train the function selecting options (here implemented by the selector), the prediction error of the option's model is substituted with the *surprise* of the experts. The idea behind this is that as evaluations of each expert are an index of the *level* of the expert's skills [17], surprise (i.e. the evaluation error of the expert's critic) is an index of the *rate of improvement* of such skills. Hence, expert's surprise is a good indicator of which expert to train.

III. METHODS

This section describes the experimental setup used to test the viability of the proposed approach.

A. *The simulated robot*

The simulated robot is a "wheelchair" robot with a 30 cm diameter and a camera assumed to look at a portion of the ground located just in front of the robot (24×8 cm). In each cycle the robot perceives the ground using a grid of 6×2

sampling points associated with color-specific RGB receptors (so the system's input is a "retina" formed by a 12×3 = 36 binary values). The robot's motor system is driven by setting the orientation variation within [-30, +30] degrees, and the translation speed within [0, 2] cm.

B. *The environment and the task*

The environment is a square arena with a regularly textured floor (Fig. 1). The robot's life is divided into two phases: "childhood" and "adulthood". During childhood the robot moves around and learns a set of basic sensory-motor skills based on its intrinsic motivational system. During adulthood, the robot learns to combine the acquired skills in order to accomplish different tasks. Each task consists of a series of time steps during which the robot has to reach a given target location starting from a particular position. During each task, when the robot reaches the target it receives a reward and is placed back at the starting position (if the robot hits the wall it turns of a random angle). The model was tested in several different environments with different floor textures and with several different tasks. The results reported here refer to the environment and the tasks shown in Fig. 1.

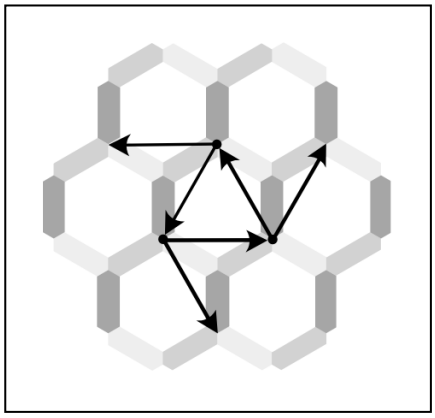

Fig. 1: The environment and the tasks. The sides of the hexagons are colored with blue (dark gray), red (gray) and green (light gray). Arrows represent the six different tasks, with each arrow's tail and head indicating, respectively, the starting and target position of one task.

C. *The controller architecture*

The controller of the robot (Fig. 2) is a hierarchical modular neural network. The system is formed by a *selector* and a number of *experts* (architectures with 3-6 experts were tested). The selector and experts are each formed by a neural-network implementation of the actor-critic model [17], with each expert including also an internal reinforcer. Hence, each expert is formed by three components (each expert functions and learns as in [21], but for the functioning of reinforcers, which is described below): (a) a *reinforcer*: this is a 2-layer neural network that maps the retina activation to a [-1, 1] sigmoid unit encoding the reward of the expert (the experts' reinforcers are evolved, see below); (b) an *actor*: this is a 2-layer neural network that maps the retinas' activation to two sigmoid units. The activation of the two units is used to set the centre of a Gaussian function used to generate noisy commands issued to the motor system (initial standard deviation = 0.3: noise is gradually reduced to zero during childhood): the first unit sets the orientation variation command and the second the translation command; (c) a *critic:* this is based on an *evaluator*, a 2-layer neural network that maps the retina activation to one linear output unit encoding the expert's evaluation. These evaluations, together with the reward produced by the expert's reinforcer, are used to compute the *surprise* of the expert in the standard way [17].

The selector is formed by two components: (a) the selector's *actor*: this is a 2-layer neural network that maps the retina activation to a number of sigmoid output units equal to the number of experts. *At each time step* [20], the activations of these output units, each corresponding to an expert, are used as pseudo-probabilities to select the expert that takes control of the motor system (i.e. selects an action) and, during childhood, learns on the basis of its reinforcer; (b) the selector's *critic,* which is a 2-layer neural network like the experts' critic. During childhood the reinforcement signals used by the selector are *intrinsic*, being formed by the *surprise* of the expert which has control on action, whereas during adulthood reinforcements are *extrinsic*, coming directly from the environment.

During childhood, at each time step the selector selects the expert that takes the control of action. This expert performs an action and trains its evaluator just as in standard function-approximation actor-critic models [17][21], that is by using surprise (i.e. the error in the prediction of future discounted rewards: discount factor = 0.9). Moreover, the expert trains its actor through a standard delta rule: if surprise is positive, it "moves" the actor's output units' activations towards target values corresponding to the executed action, whereas if surprise is negative it moves such activations away from them (see [21] for details) (learning rate of evaluator and actor = 0.009). During childhood, the selector learns, *through the expert's surprises used as reward*, to give the control to the experts which are currently maximizing the acquisition of their skill. In particular, it uses such surprise to train its evaluator in the standard way [17][21], and to train the actor with a delta rule so as to increase or decrease the probability of selecting expert just selected in the case the surprise is respectively positive or negative (learning rate = 0.05; discount factor = 0.99). Note that, during childhood, as reinforcer-based surprise needs two succeeding evaluations to be computed, both the experts and the selector learn only when an expert is selected for at least two succeeding time steps.

During adulthood, experts do not learn. At the beginning of each task, the selector is reset to random weights as the policy it learns to solve a task is not good for other tasks (also the policy it learns in childhood to train the experts is no more useful). Then it trains its evaluator and actor (as in childhood) to select experts by relying on external task-related rewards.

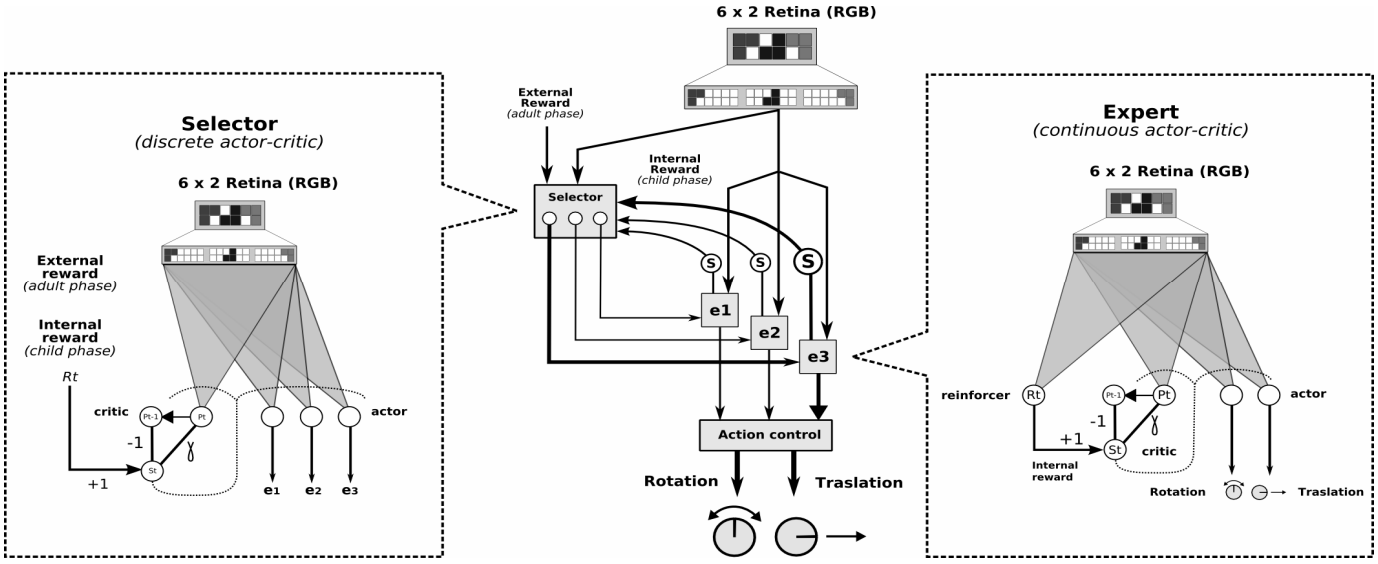

Fig. 2 Center: the whole architecture. Left: details of the selector. Right: details of one expert (see text for a detailed description).

D. *The genetic algorithm*

A genetic algorithm is used to evolve the weights of the experts' reinforcers. A population of 50 individuals is evolved for 50 generations. Each individual corresponds to a robot's genome and encodes the connection weights of the experts' reinforcers as real numbers.

Childhood lasts 100,000 steps times the number of experts (e.g., 3 in the experiment reported below). Adulthood lasts 500,000 steps times the number of tasks to be solved (e.g., 6 in the experiment reported below). The fitness is computed counting the number of times that the robot reaches the target at the end of each task and is normalized in [0, +1] by dividing such number by the maximum achievable theoretical successes. The fitness' measurement is carried out only in the last 50,000 steps of each task sub-phase.

At the end of each generation the best 10 individuals are selected and used to generate 5 offspring each with a mutation rate of 10% per connection weight. Mutation is performed by adding a random value in [-1, +1] to selected weights.

IV. Results

The system was tested in different environments, with different kinds of tasks, and with different numbers of tasks and experts. Overall, the results are quite promising: both average and best fitness rise quickly and reach a steady state value in a very few generations (typically in about 10-20 generations). Depending on the particular conditions of the tests, fitness reaches values between 0.6 and 0.8. These are quite high values, considering that a fitness of 1 would require the robot to always go from its starting position to the target following a straight line and at maximum speed: this is unlikely to happen as the robot cannot see the target and can only rely on local ground-texture information. In what follows we present a brief analysis of the typical strategies that evolved robots develop in order to solve their tasks (other conditions gave qualitatively similar results).

Fig. 3 shows the behavior of the best robot, endowed with a controller with three experts, evolved to tackle the tasks reported in *Fig. 1*. Fig. 3a shows the typical behavior displayed by "child" robots in the first cycles of life. Since at the beginning of life the connection weights of all the neural networks of the architecture are randomly set, behavior cannot but be random. In particular, the selector randomly assigns control to the various experts, which in turn act randomly. On the contrary, towards the end of childhood the robots display a very structured behavior: Fig. 3b shows how the robot has learned to robustly follow the colored lines. In particular, each expert has specialized to follow one color on the basis of evolved intrinsic reinforcer (interestingly, evolution led to the emergence of reinforcers each rewarding the perception of just one of the three colors). Moreover, the selector has learned to assign control to experts which are most rewarded (by their respective reinforcers) in following the color that the robot is currently perceiving. More precisely, the selector learnt to select experts that had the highest learning rates so enhancing their acquisition of specialized skills. The selector acquires this capability because it is reinforced by the surprise of the expert which it gives the control to. The reason is as follows. In an actor-critic architecture a critic's *evaluation*, which corresponds to the prediction of future rewards, is a good index of the actor's ability to achieve these rewards. As a consequence, the critic's *surprise*, which corresponds to the *error* in reward prediction, is a good index of the actor's rate of improvement. Hence, using surprise as the internal reward signal makes the selector learn to select the expert that is improving the most in the given context.

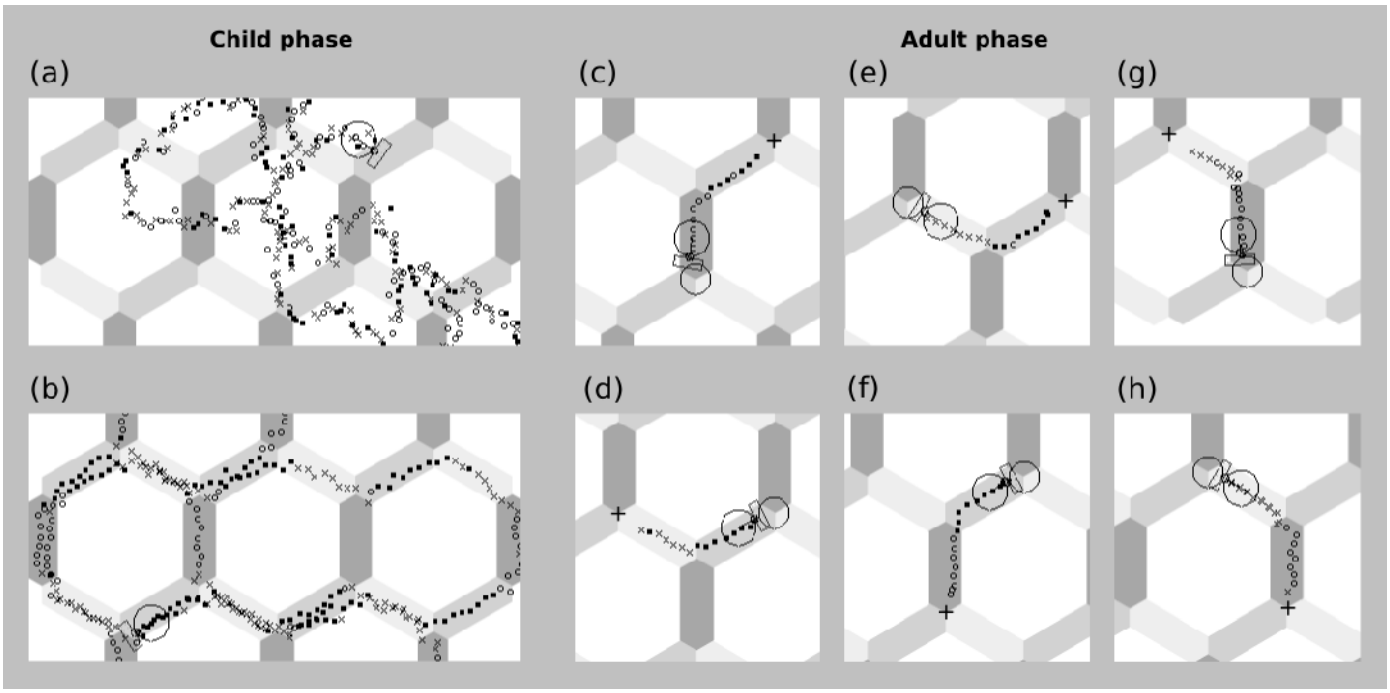

Fig. 3: Snapshots of the behavior of an evolved robot. The robot is represented as a circle with a rectangle in front of it (the retina). Small symbols (black filled boxes, empty circles, and crosses) indicate which of the three experts has been selected in a specific position and in each time step (one sample every 5 cycles). (a) About 300 steps at the beginning of robot's childhood. (b) About 300 steps at the end of childhood. (c-h) One trial for each of the six tasks at the end of the adulthood: the crosses and the circles at the junctions of colored trails represent the initial and target positions, respectively. See text for details.

The result of this developmental process is that at the end of childhood the robot has acquired a set of basic skills (sensory-motor mappings) which can subsequently be used for solving the particular tasks encountered during adulthood. This is illustrated in Fig. 3c-h, in which the behavior of the adult robot at the end of each task's learning phase is shown. As clearly shown by the graphs, whenever the robot is on one color trail the selector selects the expert which is able to follow that color (apart from rare cases due to the stochastic nature of selection). When a color trail ends and the robot arrives at junctions, the selector needs only to learn to select the expert that is best suited to follow the trail which leads to the target: this is quite easily done by the standard actor-critic algorithm which uses external rewards provided by targets during adulthood. The result is that, thanks to the skills acquired in childhood, the evolved robot is able to quickly learn to solve several different tasks. In fact the controller does not need to learn everything from scratch when solving single tasks, but can simply combine in the appropriate way the basic general skills acquired during childhood.

V. Discussion and Future Work

This paper presented an actor-critic hierarchical neural network architecture for intrinsically motivated reinforcement learning in which internal reinforcers for several experts are evolved using a genetic algorithm. We have tested the viability of the proposed approach with a robotic simulation in which a robotic agent learns to solve several different navigation tasks by combining the same basic skills learned during its developmental phase (childhood) thanks to internal rewards coming (a) by the evolved reinforcers and (b) by experts' surprise, that is, the error in the prediction of future rewards made by experts' critics. Furthermore, the system proved also to be quite robust by functioning in different environments, with different numbers of tasks, and with different numbers of available experts (provided that this number was sufficient for solving the given tasks in the given environment). Note that, for lack of space, it was not possible to compare the performance of our system with respect to other, more standard, ones. This has been done in [22], to which the interested reader is referred: suffice here to say that our system significantly outperforms both a standard neural network implementation of the actor-critic architecture and a hierarchical system similar to the one used here but in which there is no individual learning, and all the network's connection weights are evolved through the genetic algorithm.

With respect to the original proposal of IMRL [8], our model presents two important improvements. (1) Contrary to IMRL, which requires the specification of a-priori hardwired "salient events', the model uses artificial evolution for discovering experts' reinforcers that provide internal rewards. This gives the model a *high flexibility and autonomy* and requires little intervention by the researcher. In fact, by relying on the principles of self-organization, evolution is able to autonomously find internal reinforcers which can drive the development of useful. (2) Contrary to IMRL, which can currently be used only in abstract, discrete environments, our system can be used with real robotic scenarios. In fact, by using neural networks as the control system's building blocks, the system is able to cope with low-level sensory-motor

mappings in continuous and noisy. Indeed, the viability of this approach was demonstrated with a simulated yet realistic robotic set-up.

A further limitation of IMRL is related to the fact that it uses the failure in the prediction of salient events as the intrinsic reward signal. This is a limit (as recognized also by the same authors, see [8]) because it can lead the system to undesirable behaviors in environments involving areas which are intrinsically difficult or impossible to predict (see also [2, 7]). The system proposed in this paper might overcome also this limitation thanks to the use of the surprise of the experts as the intrinsic reward signal for training the selector during childhood. As discussed above, since the *evaluation* of an expert is a good index of the *level of its skill*, its *surprise* can be considered as an index of the *rate of improvement of such skill*. Hence, our system might overcome also the aforementioned problem of IMRL because in non-learnable contexts experts should tend to produce an average surprise around zero, and consequently the selector should learn to avoid selecting them.

Future work will improve the architecture under many respects, also on the basis of some appealing features of the sophisticated option framework that were lost in the current implementation of the model, for example:

1) The model's selector chooses which expert receives control *at each time step*. The possibility of assigning control to experts for prolonged periods of time might improve the performance of the system (in the options framework experts have control for the time needed to pursue their goals).
2) The neural networks used to implement the selector and experts have a simple two-layer feedforward architecture that can implement only very simple input-output mappings. However, this limitation holds only for the current implementation of the model, whose principles can in fact be used with any kind of network architecture.
3) In its current implementation, the model has a fixed number of experts. Even if the system seems to be quite robust with respect to this number, it would be interesting to let the genetic algorithm itself find the optimal number.
4) The architecture is not recursive, in the sense that each expert can only use primitive actions and not other experts to implement skills. This might be a severe limitation and is indeed one of the most interesting directions of future work.

REFERENCES

[1] J. Weng, .et al., "Autonomous Mental Development by Robots and Animals", *Science,* Vol. 291, pp. 599-600, 2001.

[2] J. Schmidhuber, "A Possibility for Implementing Curiosity and Boredom in Model-Building Neural Controllers", in J-A. Meyer, and S.W. Wilson, (Eds.), *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, The MIT Press, Cambridge, MA, pp. 222-227, 1991.

[3] A. G. Barto, S. Singh, and N. Chentanez, "Intrinsically Motivated Learning of Hierarchical Collections of Skills", *Proceedings of the Third International Conference on Development and Learning*, 2004.

[4] X. Huang, and J. Weng, "Novelty and Reinforcement Learning in the Value System of Developmental Robots", in C. G. Prince, Y. Demiris, Y. Marom, H. Kozima, and C. Balkenius, (Eds.), *Proceedings Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems* 94, Edinburgh, Scotland, pp. 47-55, 2002.

[5] F. Kaplan, and P. Oudeyer, "Motivational Principles for Visual Know-How Development", in H. Kozima D. Bullock G. Stojanov C. G. Prince, L. Berthouze & C. Balkenius, (Eds.), *Proceedings of the Third International Workshop on Epigenetic Robotics : Modeling Cognitive Development inRobotic Systems*, Lund University Cognitive Studies, Lund, pp. 73-80, 2003.

[6] J. Marshall, D. Blank, and L. Meeden, "An Emergent Framework for Self-Motivation in Developmental Robotics", in *Proceedings of the Third International Conference on Development and Learning (ICDL 2004)*, pp. 104-111, 2004.

[7] P. Oudeyer, F. Kaplan, and V.V. Hafner, "Intrinsic Motivation Systems for Autonomous Mental Development", *IEEE Transactions on Evolutionary Computation,* Vol. 11, 2007.

[8] A. Stout, G.D. Konidaris, and A.G. Barto, "Intrinsically Motivated Reinforcement Learning: A Promising Framework For Developmental Robot Learning", *in Proceedings of the AAAI Spring Symposium on Developmental Robotics*, 2005.

[9] Ö. SimSek, and A.G. Barto, "An Intrinsic Reward Mechanism for Efficient Exploration", *in Proceedings of the Twenty-Third International Conference on Machine Learning (ICML 06),* 2006.

[10] S. Nolfi and D. Floreano, *Evolutionary robotics. The biology, intelligence, and technology of self-organizing machines*, The MIT Press, Cambridge, MA., 2000.

[11] R.W. White, "Motivation Reconsidered: The Concept of Competence", *Psychological Review, Vol*. 66, no. 5, pp. 297-333, 1959.

[12] D.E. Berlyne, *Conflict, Arousal and Curiosity*, McGraw-Hill, New York, 1960.

[13] P. Montague, P. Dayan, and T. Sejnowski, "A Framework for Mesencephalic Dopamine Systems Based on Predictive Hebbian Learning", *Journal of Neuroscience,* Vol. 16, no. 5, pp. 1936-1947, 1996.

[14] W. Schultz, "Getting Formal with Dopamine and Reward", *Neuron,* Vol. 36, pp. 241-263, 2002.

[15] P. Dayan, and B. Balleine, "Reward, Motivation and Reinforcement Learning", *Neuron, Vol*. 36, pp. 285-298, 2002.

[16] P. Redgrave, and K. Gurney, "The Short-Latency Dopamine Signal: a Role in Discovering Novel Actions?", *Nature Reviews Neuroscience,* Vol. 7, no. 12, pp. 967-975, 2006.

[17] Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge MA, 1998.

[18] R. Sutton, D. Precup, and S. Singh, "Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning", *Artificial Intelligence, Vol*. 112, pp. 181-211, 1999.

[19] S. Nolfi, "Evolutionary Robotics: Exploiting the Full Power of Self-Organization", *Connection Science,* Vol. 10, no. 3-4, pp. 167-183, 1998.

[20] Baldassarre, G.: A Modular Neural-Network Model of the Basal Ganglia's Role in Learning and Selecting Motor Behaviours. *Journal of Cognitive Systems Research*, Vol. 3, pp. 5-13, 2002.

[21] Mannella, F., Baldassarre, G.: A neural-network reinforcement-learning model of domestic chicks that learn to localise the centre of closed arenas. *Philosophical Transactions of the Royal Society B – Biological Sciences*. Vol. 362, no. 1479, pp. 383-401, 2007.

[22] Schembri, M., Mirolli, M., Baldassarre, G.: Evolution and Learning in an Intrinsically Motivated Reinforcement Learning Robot. *Proceedings of the 9th European Conference on Artificial Life*. In press.