Proceedings of the 3rd IEEE International Conference on Development and Learning and on Epigenetic Robotics August 2013

Action Understanding using an Adaptive Liquid State Machine based on Environmental Ambiguity

Jimmy Baraglia, Yukie Nagai and Minoru Asada
Department of Adaptive Machine Systems, Graduate School of Engineering
Osaka University, Suita-city, Osaka
Email: jimmy.baraglia@gmail.com

***Abstract*—Recently, humans-robots interaction steps aside the traditional master/slave relationship to evolve in a new paradigm of cognitive robotics. Their conception requires the comprehension of human cognitive functions and how they develop. In this paper, we present how an adaptive Liquid State Machine using environment ambiguity may lead to a better emergence of action prediction abilities in a simple robot. The simulation results indicate the efficiency of the proposed method by which a simple robot develop its own prediction capability. These results are promising towards building robots able to develop more complicated capability such as understanding others' intention and further, cooperation with other agents.**

I. Introduction

During their first years, human and chimpanzee infants spontaneously start to develop altruistic behaviors toward their adult counterparts without explicit rewards [1]. Warneken et al. [2] suggested that human infants are naturally altruistic, while others proposed that individuals help others to the degree that they can anticipate being helped in return [3]. Already in 1871, the question of the emergence of cooperative behaviors has been discussed by Charles Darwin, who concluded that "natural selection could encourage altruistic behavior among kin so as to improve the reproductive potential of the family". In this study, we suppose that cooperative behaviors are not fully innate but mainly the consequence of active social interactions with other agents (e.g., [12], [13]).

Cooperation itself is the process of acting together toward a goal. For instance, if a cooperative robot is looking at a human trying to carry a heavy object inside a container, it must infer the goal of this person, understand that the person cannot lift the object and then offer its assistance. Robots that can act to help humans could enhance the life of elderly people at their home or provide mechanical strength to assist workers in difficult or dangerous environments.

The learning of action understanding and cooperative behaviors, which is the goal we are trying to reach, is a long term issue and many different approaches have already been treated. The most popular among them is the probabilistic approach where a robot tries to infer or extract the intention of a human using statistical models (e.g., [4], [5]). Others as [6], [7] and [8] implemented distributed agents that control several robots in parallel to make them cooperate toward a given task.

Besides the previously mentioned approaches, we are dealing with a more fundamental one presented by Asada et al. in 2001 [10] and surveyed in 2009 [17], named Cognitive Developmental Robotics (hereafter, CDR). Based on constructive approaches such as using infant-like robots, CDR aims to understand how humans develop their cognitive functions. The key concepts are "physical embodiment" and "social interaction", both of which are closely related to each other. Preceded by many studies in CDR, action understanding of others and the emergence of cooperative behaviors are important to the future goal of a human/robot symbiotic cohabitation.

To address the problem of the emergence of action understanding in a robot, we present an adaptive version of the liquid state machine called aLSM (for LSM see [16]) that is able to learn, for instance, the goal of an ongoing motion. In the experiments, a robot arm is placed in an environment with basic knowledge and simple action primitives (e.g., "Reaching"). During the learning phase, the robot explores its action space and learns the relationship between his arm movement and the action he is performing using the aLSM. After the learning, the robot is not only able to predict his own action, but also to predict the outcome of similar actions performed by other individuals. This paper presents the concept and advantages of using the aLSM, which adapt environmental ambiguity for the development of cognitive function such as action understanding.

II. Basic Ideas of our Approach

Breazeal et al. [9] proposed that during a cooperative task, a robot requires understanding of a person's intentions and goal in order to behave as a partner rather than just a tool. So far, robots are not able to completely understand the intentions or the goal of other's action based on interaction with the environment or using the contextual information. To address this issue, we have decided to develop a robotic system that emulates the emergence of new cognitive function through the interaction with the environment. The robot will then receive positive or negative feedback depending on whether it correctly predict the intention or not.

In this scope, we will present the roles and benefices of using the aLSM that is able to adapt its internal parameters to environmental ambiguity. This network allows a robot to predict simple actions performed by another individual. For instance, if the other is reaching for an apple, the aLSM will predict that his arm is going to reach for the apple before the completion of the action.

Fig. 1 represents the internal structure of the aLSM, which is made of an adaptive module that modifies a classic LSM. Our network takes inputs from the visual processing unit to

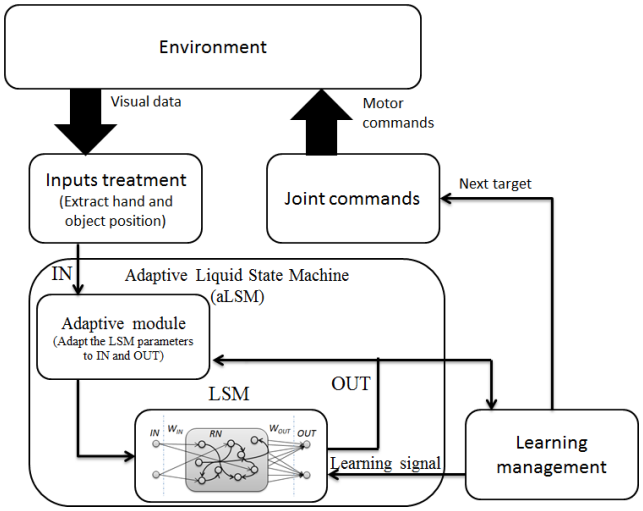


Fig. 1. Model for the emergence of action understanding using environmental ambiguity. The environment generates visual data that are treated by the Visual processing. The segmented information are then used by the aLSM. The aLSM can adapt some of this parameters using both inputs and outputs.

get information from the environment. The learning phase is triggered by the learning management unit. These units will be described in the following sections.

A. Adaptive Liquid State Machine

The LSM (see Fig. 2) is a recurrent neural network proposed by [16], which is made of spiking neurons, here Izhikevich neuron model [14] is used. It consists of a relatively large number of neurons connected to each other with different connection weights inside a reservoir (RN). Thanks to the reservoir and the recurrent nature of the network, it acquires a fading memory that turns the time varying inputs into a spatio-temporal pattern of activations in the network nodes.

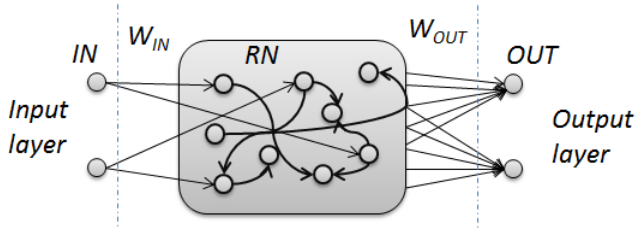


Fig. 2. Representation of a Liquid State Machine with an input layer (IN), a reservoir (RN) and an output layer (OUT).

In the LSM, the input weights (W_{in}) and the RN weights are randomly fixed between 0 and 1 during the initialization, and the stability of the reservoir is ensured using the eigenvalue of the RN weight matrix. The numbers of input and RN connections are fixed before the initialization. The output layer is connected to all RN. The ratio between inhibitory and excitatory neurons in the reservoir can be fixed.

During the learning, only the weights of the output layer (W_o) are updated. Several learning algorithms can be applied such as linear regression, reward based learning, supervised and unsupervised learning. In this study, we implement a new version of the LSM that we named aLSM. It benefits from an on-line supervised reward function using residual activation charge called Activation Charge Learning. The adaptive part of this network is done by adjusting the network parameters based on the environment ambiguity (cf. II. C).

B. Activation Charge Learning

The spiking neurons in the aLSM are activated only when their charge becomes higher than a fixed threshold according to Izhikevich model [14]. The ephemeral nature of the spikes makes it difficult to achieve efficient online learning with the existing learning methods. To solve this problem, an Activation Charge (*ACh*) is associated to each neuron in the reservoir. When a neuron spikes, the *ACh* of this neuron is incremented by a fixed value. Then, at each time steps, the charge decreases following an Activation Charge Leak factor called Charge Leak (*CL*) (see Fig. 3).

The Activation charge can be formulated by:

$$ACh(t+1) = ACh(t) + C_u \cdot RN_{Spike} - CL \qquad (1)$$

Where C_u is the charge unity (here C_u = 30mV), RN_{Spike} represents the spike activation of each neurons and *CL* is the Charge Leak factor.

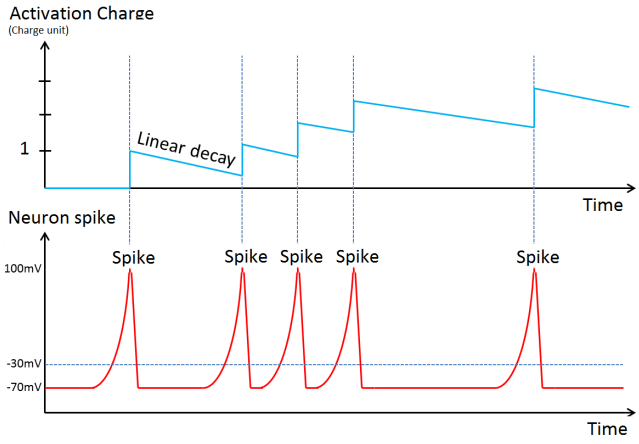


Fig. 3. Representation of activation charge. When the neuron is spiking, the activation charge increases by one charge unit. Then, each time step, the charge is reduced by a fixed factor.

Then the reward learning of the aLSM using the activation charge is represented as follow:

$$W_o(t+1) = W_o(t) + \eta \cdot LR \cdot [(OUT_{Nom} - OUT)] \qquad (2)$$

$$OUT_{nom} = OUT + Cr \cdot ACh^T \qquad (3)$$

We can then write:

$$W_o(t+1) = W_o(t) + \eta \cdot LR \cdot [Cr \cdot ACh^T] \qquad (4)$$

Where η is the fixed learning rate initialized at 0.01 and *LR* a variable learning rate.

Cr represents the correction factor. This factor depends on the reward type and on the object the robot is reaching. 3 types of reward can be generated: PosKnown is a positive reward where the good solution "i" is known; NegKnown is a negative reward where the good solution "i" is known; AllNeg in case all the outputs are negative.

$$\begin{cases} PosKnown & : & Cr(i) = 0.1 \\ & & Cr(j) = -0.1; \forall j \backslash i \\ NegKnown & : & Cr(i) = 0.05 \\ & & Cr(j) = -0.1; \forall j \backslash i \\ AllNeg & : & Cr(i) = 0.1; \forall i \end{cases}$$

The rewards represents the self satisfaction after completing an action. They allow the robot to correct his behavior when his motor commands don't match his predictions, but also to reinforce his correct predictions. They are describe as follow:

- At the initialization of the aLSM, the outputs may then generate negative values.The reward signal AllNeg is send to suppress the negative values.

- At each end of learning, if the good prediction is guessed by the system, the reward signal PosKnown will be send.

- However, if at the end of learning the good prediction is not the one predicted, the signal NegKnown is sent to the aLSM.

C. Charge leak and learning rate variation

The main goal of this paper is to analyze he influence of the two adaptive parameters on learning performance . The two parameters are the Charge Leak (*CL*) and the learning rate (*LR*), which are controlled by the Adaptive Module" presented in Fig. 1; 7 variations of these 2 parameters, given in Table. 1, are applied in the experiments to analyze results with and without the intervention of environmental information.

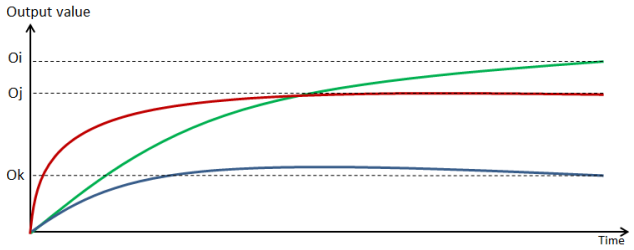
LR	Value	CL	Value
R0	1	L0	0
R1	0.02	L1	1
R2	0.001	L2	3
R3	$\exp^{(-0.0001*(tf-t))}$	L3	$\exp^{(-0.0001*(tf-t))}$
R4	1-$\exp^{(-0.0001*(tf-t))}$	L4	1-$\exp^{(-0.0001*(tf-t))}$
R5	$f(\Omega$(IN))	L5	$f(\Omega$(IN))
R6	$f(\Omega$(OUT))	L6	$f(\Omega$(OUT))

TABLE I. VALUE OF THE DIFFERENT SETTINGS OF THE CHARGE LEAK, THE VARIABLE LEARNING RATE. T IS CURRENT TIME STEP, AND TF THE FINAL LEARNING TIME STEP AND Ω REPRESENTS THE AMBIGUITY (CF. II. D.)

In Table. 1, R/L 0-4 represent a non adaptive LSM while R/L 5-6 represent the aLSM.

D. How to calculate environmental ambiguity

The environmental ambiguity Ω represents the percentage of difference between the good solution and the maximum of all others values (see Fig. 4). It can be extracted from the input and the output values. The ambiguity is calculated during the learning only. The nominal values of IN and OUT are then known by the robot. Fig. 4 shows an example of possible output values of the aLSM and how Ω is calculated in percentage. *Oi* represents here the correct prediction. *Oj* and *Ok* are two other incorrect predictions. Same function can be applied tot he inputs.



$$\Omega = (100/Oi) \cdot Max(O(n); \forall n \backslash i) \quad (5)$$

$$\Omega = 100 <= if(\Omega) > 100 \quad (6)$$

Fig. 4. How to calculate the ambiguity Ω of the output values. It represents the difference between the good solution and the maximum of all others values. The green line represents the good solution, the others are incorrect guess. See Fig. 7 to see an example of how Ω is calculated.

The measure of ambiguity gives us the information about the the precision of the input and output. For instance, Ω will be high when the precision is low. (Ω used in our study is described in III. C.)

III. IMPLEMENTATION

The aLSM have been implemented in a simple 2 degrees-of-freedom (DoF) simulated robotics arm (Fig. 5 (a)). This design allows the robot to reach for objects in a 2D space and to perform simple actions like "grasping", "pushing" or "throwing". For the current experiment, only "reaching" is used. We also started to implement a 22 DoF air pressure actuated humanoid robot (Fig. 5 (b)) to take into account the embodiment variable in the learning process. However, this robot is yet unable to perform any object manipulation and the results will not be presented in this paper.

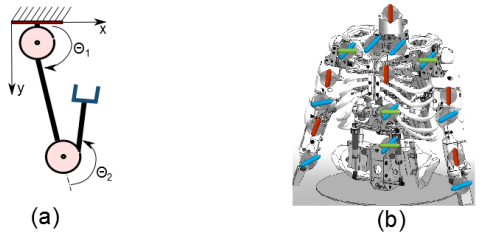


Fig. 5. (a): 2 DoFs robot arm used for the first tests. (b) a humanoid robot with 22 DoFs [11].

A. Basic setting for experiment

As shown in Fig. 6, 2 arms are placed in a simulated environment. The left arm represents the robot's arm and the right one, the experimenter's. 3 types of object can be placed in the environment: a mug, a star and a ball. The number of each object is not limited, but different color marker will be created. The position of each object can manually be changed during the experiment and may be randomly replaced at the beginning of each learning. The robot's arm is made of 2 joints and 2 sections of 100 pixels each. The experimenter's arm has also 2 joints but the sections are, from the base 200 and 150 pixels of length respectively.

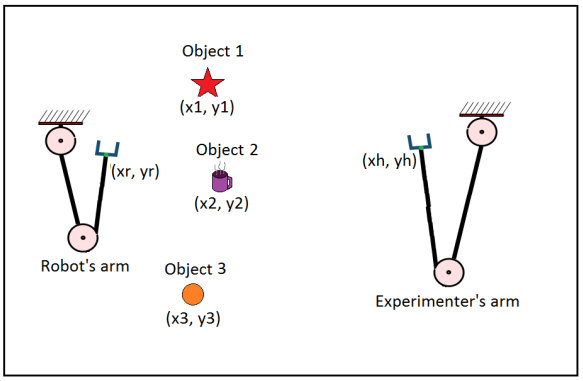


Fig. 6. Simulated environment. Each arm and object has a x and y position in the 2D space.

B. Design of the aLSM

The inputs of the aLSM (IN) are directly connected to the visual processing module. The number of inputs of the network is equal to the number of objects. In order to get similar inputs regardless of which arm (robot's of experimenter's) is moving, a perspective free value called STO (Speed Toward Object) is calculated from the position of the arm's end-effector and the objects' position. STO can be represented as follow:

$$ST_i = \frac{\partial \sqrt{(xh - x_i)^2 + (yh - y_i)^2}}{\partial t} \tag{7}$$

$$IN = STO = \begin{pmatrix} ST_0 \\ ST_1 \\ ... \\ ST_{n-1} \\ ST_n \end{pmatrix}$$

Where *n* is the number of objects. *xh* and *yh* are the humans' end effector position and *xi* and *yi* is the object's position. To calculate STO from robot arm, *xr* and *yr* are used instead of *xh* and *yh*.

The output layer (OUT) is calculated as the sum over time of the spike activations. It represents the reaching prediction for each object. For instance, if the arm is reaching for the object 1, the output number 1 should have a higher value.

$$OUT = \sum_t (W_o(t) \cdot RN_{Spike}(t)) \tag{8}$$

OUT is initialized at 0 when the reward function is activated.

The implemented aLSM is composed of 150 neurons in the reservoir. The number of input and output neurons is equal to the number of objects added in the environment. The inhibitory/excitatory ratio is 60%.

C. Design of the LR and the CL based on the environmental ambiguity

The CL is fixed between [0; 3] and the LR between [0; 1] based on the ambiguity of IN and OUT. If Ω(IN) is low, it means that the inputs values of STO are really different from each other. The task is then easier, and the learning speed may increase. However, if Ω(IN) is high, the input values are similar, which leads to more difficult learning. The learning speed should then decreases. Finally, a high Ω(OUT) means that the learning is not yet advanced and, consequently, the learning speed should increases. We can write:

$$\begin{cases} LR = \text{f}(IN) & = & 1 - (0.01 \cdot \Omega(OUT)) \\ LR = \text{f}(OUT) & = & 0.01 \cdot \Omega(IN) \\ CL = \text{f}(IN) & = & \frac{3}{100} \cdot \Omega(IN) \\ CL = \text{f}(OUT) & = & 3 - \frac{3}{100} \cdot \Omega(OUT)) \end{cases}$$

Therefor, when the LR and the CL are the function of Ω, the environmental ambiguity can compensate the learning efficiency according to the input and/or output values. Our hypothesis is that the learning performances will be better when both the CL and the LR are adaptive depending on the environment. In Fig. 7, you can see an example of the different steps to calculate the *CL* and the *LR*.

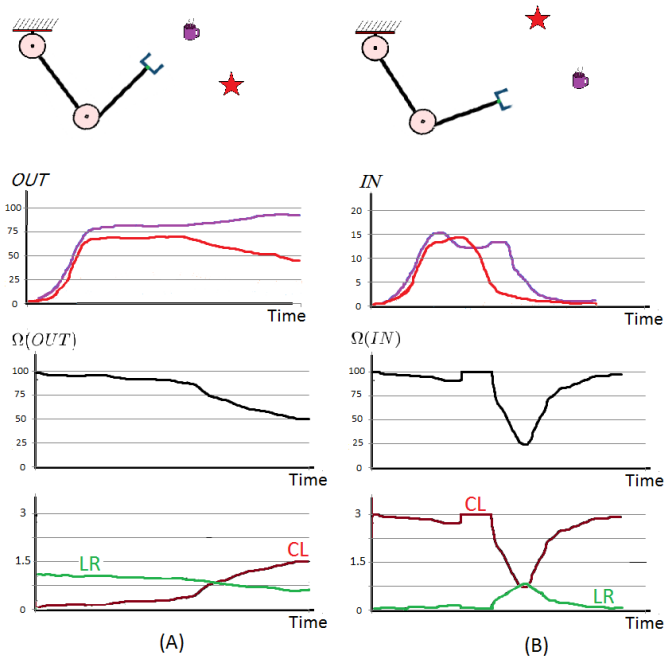


Fig. 7. Two types of adapting setting CL and LR. (A): LR and CL are function of output values (OUT). From top to bottom, the arm is reaching for the purple object. The OUT densities are measured from the output layer of the aLSM.At the third row, Ω(OUT) is calculated and finally LR and CL are calculated. (B): Same process for (A) but Ω is calculated from the input values (IN), namely STO.

IV. Experimental Setting and Results

For each different setting presented in Fig. 8 (A) to (D), and for each parameter variation,the learning experiments have been conduced at least 3 times. Each learning lasted approximately 3 minutes and the testing phase 30 seconds. In total, the learning have been performed 588 times. First, in Fig. 8 A, 2 objects were placed at fixed positions in the environment during the learning phase (100 pixels far from each other). During the testing phase, only the object that the experimenter's arm was reaching remains. These steps were reproduce for 4 objects in Fig. 8 C. Then, in Fig. 8 B with 2 objects and in Fig. 8 D with 4 items, the objects were randomly placed with a minimal distance of 70 pixels between each other, and all the objects were presented at a different position

during the testing phase than during the learning. In term of learning complexity, A was the simplest while D the more difficult.

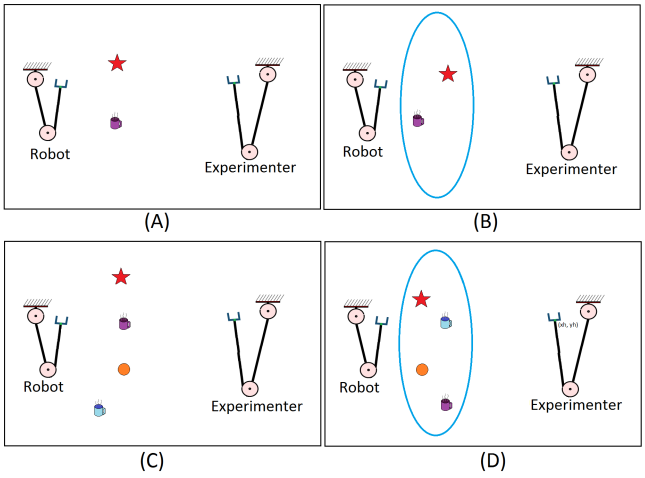


Fig. 8. Object position for the four settings. (A): 2 fixed objects. (B): 2 objects randomly positioned. (C): 4 objects fixed. (D): 4 randomly positioned objects.

Fig. 9 represents the success rate of each parameters setting *LR* and *CL* (c.f. Table 1). As we can see, the success for the parameter LR=R2 was always low. The learning rate R2 corresponds to *LR* = 0.001. Combined with η= 0.01 the global learning rate is then $10^{\wedge -5}$. The system was then not able to learn the predictions fast enough to converge to a solution.

Fig. 10 shows the average success rate for each object setting. This figure shows the difference between conditions using an aLSM (i.e., R5, R6, L5, and L6) and a non-adaptive LSM (i.e., R0 to R4 and L0 to L4). For each, multiple sets of parameters have been used (see Table. 1). The following sections discuss the results in more detail.

A. *Two objects setting, fixed positions*

The setting shown in Fig. 8 (A) is the easiest to learn. The input ambiguity remains all along low, which allows the system to learn quickly and efficiently with an average success rate of 75.5%.

As shown in Fig. 9 (A), the success rates for *CL* = L0 are lower than the other. L0 represents *CL* = 0; because the charge leak is null, even the smallest input ambiguity perturbs learning performances. Then, the lower success rate for L3 and L4 may be explained by the variable (exponential) nature of the charge leak that is most likely to affect negatively the learning.

As shown in Fig. 10 (A) the success rate is 30% higher when using the environmental ambiguity to control *CL* and *LR*.

B. *Two objects setting, random positions*

In this experimental setting shown in Fig. 8 (B) , all the objects are randomly placed in the environment during both learning and testing. The average success rate (69.38%) was

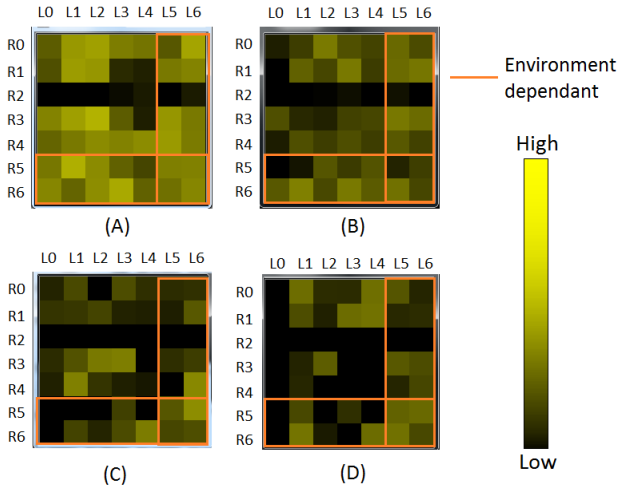


Fig. 9. Individual success rate for all set of LR, CL and experiment setting. (A): 2 objects are fixed during the learning, just one is left at the time in the environment during the testing. (B): 2 objects are randomly positioned during the learning, all the objects remain in the environment during the testing. (C): 4 objects are fixed during the learning, just one is left at the time in the environment during the testing. (D): 4 objects are randomly positioned during the learning, all the objects remain in the environment during the testing.

lower than for (A) due to the complexity of the environment. R2 and L0 lead to the lowest performance for the same reasons.

The learning rate R5 also generated an lower average of success rate. As R5 is a function of Ω(IN), it can be explained by the fact that the value of *LR* is only used when the reward function is called. Therefore, even if the input ambiguity was high all along the action, only the last value was used in the learning regardless of its previous state.

Here also, the success rate is around 20% higher (Fig. 10 (B)) when the environmental ambiguity was taken into account, namely R5, R6, L5 and L6 (see Fig. 9 (B)).

C. *Four objects setting, fixed positions*

The condition was more complex in Fig. 8 (C). The fact that 4 objects were presented to the system generates higher ambiguity at the inputs level. The objects had fixed position and only 1 object was presented at the time during the testing. As a consequence, the learning performances decreased to the average success rate of 42.85%.

Some combination with R3 produced better results. These performances can be explained by the fact that the learning rate started high and then decreased to a low value, which allowed the system to quickly learn and then not to forget (see Fig. 9 (C)).

However, in this setting the success rate was only 2% higher when the environmental ambiguity was used for the learning parameters (Fig. 10(C)). This result is not surprising as only one object at the time remains during the testing phase, which facilitates the success of the experiment for the non adaptive LSM.

D. Four objects setting, random positions

This setting is by far the most complex (Fig. 8 (D)). The inputs ambiguity is really high and the average success rate is 34.69%.

Regarding R2 and L0, the results were the lowest, which was congruent with the previous settings. However, the combination R0-L4, R1-L3 and R1-L4 also produced better results. More experiments conducted to better understand the reasons (see Fig. 9 (D)). The combination of both the *LR* and *CL* depending on the environmental ambiguity also gave better results.

The success rate is here also 19% higher when the environmental ambiguity was used for the learning phases (Fig. 10(D)).

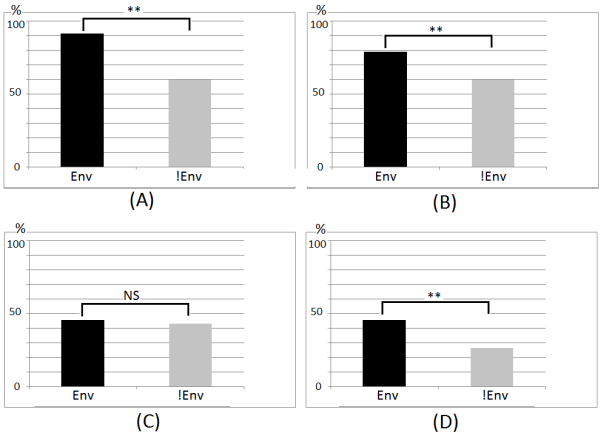


Fig. 10. Comparison of success rate for each experiment setting. (Env): Success rate for the environmental dependent settings. (!Env): Success rate for the settings that do not depend on the environment.

V. Conclusion and discussion

In this paper, we proposed an aLSM for action understanding and analyzed the effect of the environmental ambiguity on the learning performance. This is the first step toward development of robots able to read others intention and to generate cooperative behaviors. In the experiments, each object setting showed slightly different properties and sometime, opposite results. It most likely means that, as we thought, the environment plays an important role in the learning of action understanding. Indeed, as Fig. 9 and Fig. 10 show, the learning performance is all the time higher when the parameter *LR* and *CL* are depending on the environmental ambiguity. The result is even stronger when the environment is complex during the testing phase. These results clearly show that the aLSM plays its role by compensating the inaccuracies of the sensory information. We can then conclude from this set of experiments that the learning of action understanding becomes more efficient when the robotic system takes into account the environment and its own prediction ambiguity.

However, some results still have to be analyzed, for example, focusing on the learning speed with each parameter. Moreover, further experiments must be conducted with different setting to corroborate the results given in the paper. As a next step, the system will be enriched so that the robot can anticipate the intention of actions based on the contextual information as well as the goal of the actions. The system must then be able to extract all possibles action that the other could perform and to find how it can collaborate with him toward the realization of the his goal.

Acknowledgment

This study is partially supported by JSPS/MEXT Grants-in-Aid for Scientific Research (Research Project Number: 24000012, 24119003).

References

[1] WARNEKEN, Felix et al., Altruistic helping in human infants and young chimpanzees. Science, 2006, vol. 311, no 5765, p. 1301-1303.

[2] WARNEKEN, Felix et al., The roots of human altruism. British Journal of Psychology, 2009, vol. 100, no 3, p. 455-471

[3] TRIVERS, Robert L. The evolution of reciprocal altruism. Quarterly review of biology, 1971, p. 35-57.

[4] SCHMID, Andreas et al., Towards Intuitive Human-Robot Cooperation. In : Proceedings of the 2nd International Workshop on Human-Centered Robotic Systems (HCRS). 2006. p. 7-12.

[5] LALLEE, S et al., Human-robot cooperation based on interaction learning. In : From Motor Learning to Interaction Learning in Robots. Springer Berlin Heidelberg, 2010. p. 491-536.

[6] AGAH, Arvin et al., Autonomous mobile robot teams. In : Conference on Intelligent Robotics in Field, Factory, Service, and Space(CIRFFSS'94), Houston, TX. 1994. p. 246-252.

[7] CAPITAN, Jesus et al., Decentralized multi-robot cooperation with auctioned pomdps. In : Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012. p. 3323-3328

[8] HIRATA, Yasuhisa et al., Distributed robot helpers handling a single object in cooperation with a human. In : Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on. IEEE, 2000. p. 458-463.

[9] BREAZEAL, Cynthia et al., Working collaboratively with humanoid robots. In : Humanoid Robots, 2004 4th IEEE/RAS International Conference on. IEEE, 2004. p. 253-272.

[10] ASADA, Minoru et al., Cognitive developmental robotics as a new paradigm for the design of humanoid robots. Robotics and Autonomous Systems, 2001, vol. 37, no 2, p. 185-193.

[11] ISHIHARA, Hisashi et al., Realistic child robot Affetto for understanding the caregiver-child attachment relationship that guides the child development. In : Development and Learning (ICDL), 2011 IEEE International Conference on. IEEE, 2011. p. 1-5.

[12] BOYD, Robert et al., Culture and the evolution of human cooperation. Philosophical Transactions of the Royal Society B: Biological Sciences, 2009, vol. 364, no 1533, p. 3281-3288.

[13] DUFFY, Johnet al., Cooperative behavior and the frequency of social interaction. Games and Economic Behavior, 2009, vol. 66, no 2, p. 785-812

[14] IZHIKEVICH, Eugene M. Simple model of spiking neurons. Neural Networks, IEEE Transactions on, 2003, vol. 14, no 6, p. 1569-1572.

[15] HAZAN, Hananel et al., Topological constraints and robustness in liquid state machines. Expert Systems With Applications, 2012, vol. 39, no 2, p. 1597-1606.

[16] MAASS, Wofgang et al., Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Computation. 14(11), 25312560, (2002).

[17] ASADA, Minoru, et al. "Cognitive developmental robotics: A survey." Autonomous Mental Development, IEEE Transactions on 1.1 (2009): 12-34