Improvements on Relational Reinforcement Learning to Solve Joint Attention

Renato Ramos da Silva
Department of Computer Science
ICMC, University of Sao Paulo
São Carlos, São Paulo
ramos@icmc.usp.br

Roseli Ap. Francelin Romero
Department of Computer Science
ICMC, University of Sao Paulo
São Carlos, São Paulo
rafrance@icmc.usp.br

***Abstract*—The joint attention is an important cognitive function that human beings learn in childhood. This nonverbal communication is very important for a person understands other individuals and the environment during the interaction. Because of this, it is essential that the robots learn this skill to be inserted in the environment and interact socially. In this article, we have enhanced a robotic architecture, which is inspired on Behavior Analysis, to provide the capacity of learning joint attention on robots or agents using only relational reinforcement learning when the environment changes. Then, a set of empirical evaluations has been conducted in the social interactive simulator for performing the task of joint attention. The performance of this algorithm have been compared with the Q-Learning algorithm, contingency learning algorithm and ETG algorithm. The experimental results show that this algorithm solves the problems of learning and makes the architecture with greater flexibility to insert new modules.**

Keywords-joint attention; relational reinforcement learning; social robots; shared attention.

I. INTRODUCTION

Attention is the process whereby an agent concentrates on some features of the environment to the exclusion of others. The agent's concentration can be broken when some event happens and the attention is changed by some gaze behavior. Gaze behavior is a crucial element of social interactions and helps to establish triadic relations between self, other, and the world. In others words, this ability is very important for communication among humans because it helps a person expresses his or her intentions around external entities [1], [2].

The term joint attention (JA) is one type of gaze behavior. It is typically used to denote when the directing of attention of a person is taken to a third entity, an object or an event (e.g., a sound or a common goal), by focusing attention sequentially (not simultaneously) from another person. In the final of the process, reinforcement is assimilated. The importance of JA in humans and the greater inclusion of robots in our environment led robotics researchers seek mechanism that enables robots with this capability. The researches include creating mechanism to provide robots with the skill of JA or the ability of robots to learn it.

Reinforcement Learning (RL) is a machine learning method used by computer scientist to provide a machine learn only interacting with the environment. This approach has been used on robots to provide robots the capability of learn JA because this approach emphasizes the role of biologically plausible reward-driven learning processes. This plausible is explained with a basic set to construct an agent by Triesch et al [3]. This set includes perceptual skills and preferences, reinforcement learning, habituation and a structured social environment.

When RL algorithm is used as learning mechanism it needs additional information for learning process and it causes difficulties to create new modules in the architecture. In order to provide this capability for a robot only using reinforcement learning (i.e., without additional information) by adopting Markovian assumption is respected, might be infeasible. For JA, the problem is when the robot must to return to eye contact from a state, which is characterized as empty.

In our previous work [4], we proposed an algorithm where the last action is associated with current state to choose the next action to solve the problem cited above. But some gaps need to be solved before we start to insert new modules in our architecture. One example is a necessity of a caregiver stay fixed while the robot learn to return the attention to the caregiver on early rounds of interaction.

Then, this paper reports an ongoing work aimed at developing the robotic architecture, which is inspired on Science of Behavior Analysis [5], [6], [7]. In order to provide this, we proposed an improvement on FAIETGQ algorithm using the idea of plan of actions to select an action to response only when the environment changes. After this, we have incorporated four different learning algorithms in the robotic architecture. It has been inserted and evaluated in the simulator of social interactions. Then, the robotic architecture has been evaluated in the context of the JA.

This article is organized as it follows. We start with related work section making the case for a rigorous experimental study of JA behaviors. After, we describe the robotic architecture, in which the learning mechanism will be inserted in. In Section IV, we present the FAIETGQ algorithm and the changes proposed in this paper and the advantages of it. Then, a social interaction simulator are presented in Section V. Afterward, in Section VI, the experimental

 ISBN: 978-1-61208-155-7

results from a set of experiments carried out to evaluate the performance of the proposed architecture with each learning algorithm tested. A comparative analysis among the four learning algorithms is also been presented. Finally, in Section VII, conclusion and future works are presented.

II. Related work

Despite the biological plausibility of the RL, it is not the only way exploited by researchers to enable a robot with the ability to learn to JA.

A temporal-difference (TD) reinforcement learning scheme for learning joint visual attention was proposed by Matsuda[8]. This model is limited because the robot only gets reward when the object, operated by the observer, moves itself. Also the caregiver's face is treated separately from the objects and does not lead to any reward, that is, mutual gaze was not considered in this work. Nagai et al. [9], [10] used face edge features and motion information (optical flow) to estimate the sensor motor coordination and the motor output using two separate neural networks. Their model does not utilize the depth information of the images and thus can not handle ambiguous situations where an object appears in robot's gaze direction that may not be located within the caregiver's gaze direction. Shon et al. [11] presented a probabilistic model of gaze following imitation in which estimated gaze vectors are used in conjunction with the saliency maps of the visual scenes to produce maximum a posteriori (MAP) estimative of the object positions attended by the caregiver. In another study, Triesh proposed a basic set of structures and mechanism for gaze following [3]. This set includes perceptual skills and preferences, reward-driven learning, habituation and a structured social environment. This work is evaluated only on simulator. Kim et al. [12] improved a model that uses a basic set [3], on a robotic head. They have been used an actor-critic reinforcement learning model for learning gaze following. The drawback of this proposed method is related of using a salient map as additional information. More specifically, this map uses representations of the caregiver head direction (h) and the caregiver eye direction (e). In our previous work [13], we applied the contingency learning in architecture for JA aiming to control a real robotic head.

III. Robotic Architecture

The robotic architecture is under development and aims to build intelligent agent based on Behavior Analysis theory [5], [6], [7]. This study is motivated to help understand the human being and help someone in many parts of our day, like robots assistants and entertainment activities. Thus, it is composed by two main modules: Stimulus Perception is State Estimation and Response Emission Module is the Controller.

Figure 1 illustrates the general organization of the architecture and the interaction between modules. Arrows

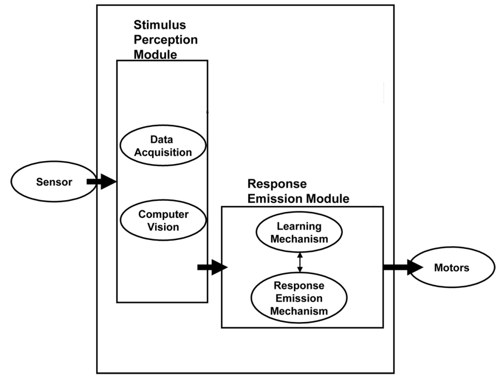

Figure 1. General organization of the architecture.

indicate the flow of information in the three modules of the architecture. The circles indicate the methods and component structures of the modules. The Stimulus Perception Module encodes stimulus from environment then it is used by Response Emission Modules for learning and exhibition appropriate behaviors.

The Stimulus Perception Module (SPM) may employ algorithms of data acquisition, a vision system, and a voice system, depending on the application domain. This module detects the state from the environment and encodes this state using an appropriate representation. The relational representation was chosen because it enabling the representation of large spaces in an economical way.

The Response Emission Module (REM) is composed by a learning mechanism that constructs a nondeterministic policy for response emission, that is, what response is to be emitted on the presence of certain antecedent stimulus. Other function of this mechanism generates a reward on the basis of the internal state estimate. Other part of REM, the response emission mechanism receives the information from learning mechanism and converts it in action to be executed by the motors.

The original version of this robotic architecture has a Motivational Module and it was proposed by Policastro [13]. This module helps to provide the ability of learn JA to robots but it create some problems to insert new modules. Now, in the process of developing this robotics architecture with new abilities to be incorporated into, we remove the Motivational Module and we are looking for a better learning mechanism for it.

The Consequence Control Module (CCM) is composed by a motivational system that simulates internal necessities of the robot. The motivational system is formed by necessity units that are implemented as a simple perceptron [14] with recurrent connections. Those necessity units simulate the homeostasis of alive organisms. A positive value of a necessity unit, above a predefined threshold, indicates the privation of the robot to certain reinforcement stimulus. In this way, the architecture supplies mechanisms to simulate privation states and satisfaction of necessities.

The motivational system works as it follows. Initially, the stimuli detected from the environment are sent to the consequence control module. Then, the *Preprocessor* encodes these stimuli to construct an appropriate input pattern. This input pattern may be or not normalized, depending on the numeric interval of the selected connection weights and problem domain. Afterwards, the necessity units calculate their activation values and their output values. After this, the *Mediator* performs a competition among all unit outputs and selects the winner. The *Mediator* checks if the winner is higher than the activation threshold. If so, the motivational system outputs the active necessity [13].

IV. Learning mechanism

A. FAIETGQ

The main idea of FAIETGQ algorithm is related with the possibility of using the previous action as a way of solving the problem of JA. Then, only with the action that resulted in the state that subsequently led to positive reinforcement are used as information to choose next action [4].

The FAIETGQ algorithm learns a control policy for an agent while it moves through the environment and receives rewards for its actions. An agent perceives a state s_i, decides to take some action a_i, makes a transition from s_i to s_{i+1} and receives the reward r_i. The task of the agent is to maximize the total reward it gets while doing actions. Agents have to learn a policy which maps states into actions. All knowledge is stored in a tree and the mechanism responsible to infer an action is called relational regression engine.

This engine is denominated relational because the representation of states. Moreover, the states use binarization by conversion of a categorical attribute to asymmetric binary attributes [15]. The regression is the mechanism of using a tree with a dependent variable action and the independent or predictor variable state [16].

The learning mechanism takes the state from the SPM and use the relational regression engine to select the action. This mechanism tries to find the current state in the intermediate nodes. If this operation is positive it takes an action which antecessor action executed in current action was positive, otherwise a random action is done. The action executed over the state changes it and the agent receives its reward. The reward can be either positive (equals to 10) or negative (equals to -1). After this occurred, the $qvalue$ is computed by:

$$\hat{Q}_i \leftarrow Q(s_i, a_i) + \alpha[r_{i+1} + \gamma \max(Q(s_{i+1}, a_{i+1})) - Q(s_i, a_i)] \quad (1)$$

Then, the relational regression engine is updated receiving a set of (state, action, $qvalue$, $last\_action$). The state is tested with internal nodes if it already exists. In the case this performance is false, the state is inserted in the tree and the leaf receives the action with the $qvalue$ and $last\_action$, forming a new branch. Otherwise, it updates the $qvalue$ for respective action in the leaf node.

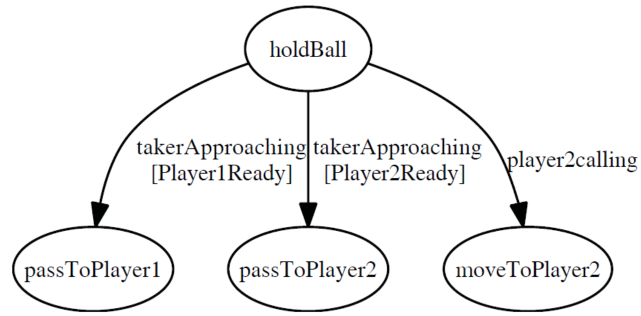

Figure 2. An example of simple plan. Actions label states, events and guards label transactions [17]

In a leaf node, more than one action can be considered. For an easier access to the most adequate action, these actions can be ordered in decreasing order according to their $qvalue$ always that an example is inserted or updated. Each leaf also has a $last\_action$ associated with action and it refers to a $last\_action$ of the robot to choose this action on this state. This process is repeated until there are not more interactions to be executed.

B. New FAIETGQ

In this section, we propose a simple modification on FAIETGQ algorithm based on the idea of agent plan by Leonetti [17]. His work used a reactive plans representation with graphs, or charts, for plan state and event. The plan states are the nodes of the graph which represent the actions that can be performed and the events are the vertices that reflect environment state. Moreover, the vertices may have guardians, which are conditions that must be satisfied to reach the other state. Thus, the agent remains on a node until the state is changed. If there is a guardian, the condition must be satisfied for the agent to move from one node to another.

The Figure 2 is an example of simple plan for robot soccer. The agent starts holding the ball until an event occurs. When *takerApproaching* or *player2calling* happen the agent can go to the next node. The agent reaches the node *passToPlayer1* or *passToPlayer2* if the condition of *Player1Ready* or *Player2Ready* is satisfied.

Our propose change the FAIETGQ to take an action only when a state changes. Then, in the beginning of the interaction the agent do not need to do a random search to find a target, object or human, and we can improve our experiments eliminating the necessity of the caregiver waiting fixed looking for a robot while it learns to return to mutual gaze. In addition this new algorithm is more natural than the original because the robot only reacts to a change in the environment. In other words, the robot, in early learning stage, do not need to search disorderly to find a correct action to establish eye contact.

However, this algorithm creates a problem during the interaction. When the caregiver turns his head to the object and the robot turns wrongly to another place. At his moment, the robot does not find the object, receive a negative reward and the state will not change. Because this problem other addition to algorithm is inserted. When the robot receives the negative reward it does not wait to change the state of the environment, it immediately takes an action.

The new FAIETGQ algorithm learns a control policy for an agent. It learns only from the interaction with the environment and receives rewards for its actions. An agent perceives a state s_i and compare with its previous s_{i-1}. If they are the same, the algorithm verify if the reward is negative to take an action a_i randomly. Otherwise, the robot wait for the next state s_{i+1}. The other case is when the states are different, then the agent decides to take some action a_i following the knowledge stored. After it takes an action it always makes a transition from s_i to s_{i+1} and receives the reward r_i. The task of the agent is to receive positive reward while takes actions. Agents have to learn a policy which maps states into actions.

All new knowledge is stored in a tree. This relational engine receives a set of (state, action, $previous\_action$) and tests the internal nodes if the state already exists. In the case this performance is false, the state is inserted in the tree and the leaf receives the action with the $previous\_action$ and necessity values, forming a new branch. The tree is updated when a positive example occur.

V. Social Interactive Simulator

To evaluate the proposed architecture, an interactive social simulator has been developed by us and it is presented here. This social interaction simulator is able to simulate an interaction between a robot and a human in a controlled social environment.

In order to simulate the JA task, it has been defined three entities that can be manipulated through functions of the simulator. They are a human, a robot, and two toys. The human being and the robot are positioned face to face, at a distance of approximately 50 cm from each other. The simulator enables that up two toys are positioned in the social environment. A toy can be positioned at any empty place of the social environment at any moment.

The social environment was modeled in the following way. Both the robot and the human can turn left or right their heads up to 90°. The robot has its central focus in 0° and has its visual field limited by a foveation parameter $\lambda°$, starting from the central focus, in $[-\lambda°, +\lambda°]$.

The position of the robot's head is given by θ_r, that can assume values in $[-90°, +90°]$. The position of the human

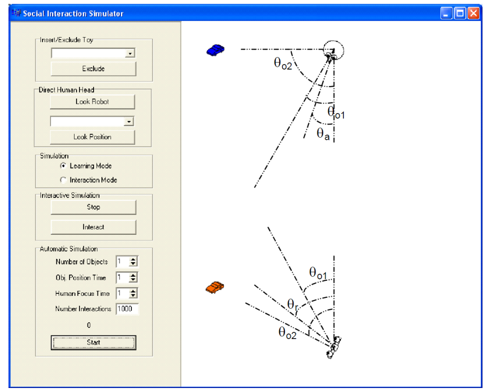

Figure 3. Positioning control [18].

being's head is given by θ_a, that also can assume values in $[-90°, +90°]$. When an object i is positioned in the social environment, the simulator maps the angle between this object and the robot's focus, that is, the distance that the robot must move its head to focus the positioned object. This mapping is given by θ_{oi}, that can assume values in $[-90°, +90°]$. In this way, if an object is positioned in the environment, the simulator verifies if the same is inside the robot's field of vision, by comparing its position in relation to the robot's focus, considering the foveation of the robot.

Figure 3 shows the interface of the developed simulator with the robot head position explained above. In this figure, on the left side of the interface is the control panel that enables interactive or automatic simulations, the human being is fixed on the upper side of the interface and the robot is fixed on the lower side of the interface.

Additionally, the simulator provides an adult attending stimulus that simulates attention from human being to the robot. The simulator provides the stimulus when the human and the robot are keeping eye contact and when the robot correctly follows the human gaze. This mechanism was incorporated in the simulator to validate the behavioral analysis presented by Dube and their colleagues [19], stating that the human serves as motivational operator in the context of JA learning.

During a simulation, the simulator executes interactions continually and each interaction takes 1 second. The simulator is able to position up to two simultaneous objects in the social environment, on places stochastically selected with probability ρ_o. These objects are positioned in the respective places for a time determined by the user (given in seconds). Additionally, the simulator is able to turn the human being's head to focus an object present in the environment or to focus the robot. The object that receives the human's focus is stochastically selected with probability ρ_{oi} and the human keeps his focus at the selected object for a time determined by the user (given in seconds), before turn his head to another object or to the robot.

VI. EXPERIMENT

In this section, the main results of the experiments carried out to evaluate the proposed learning algorithms are presented and discussed. The experiments were carried out employing the simulator previously presented, in the context of the emergence of JA. The purpose was to evaluate the capabilities of the new version of the robotic architecture on exhibit appropriate social behavior and learn from interaction.

For a complete evaluation of this previous proposed, we compared it with FAIETGQ to analyze the improvements of this algorithms. In addition, we used the old architecture with Contingency Learning, Q-learning and ETG algorithm to compare with new FAIETGQ to validate of this new architecture over the original version.

The experiments were composed by a learning phase of 10,000 time units (10,000 seconds in the simulator). During the learning phase, the human being initially kept the focus on the robot until it establish eye contact with him, characterized by 3 time units looking each other. Then, two objects were positioned in the environment and the human being turned his gaze for one of these objects, obeying the probabilities defined in the social interactive simulator. The human keeps his gaze at the object by 5 time units. Afterwards, the objects are then removed from the environment and the human turns his gaze to the robot, keeping the robot make eye contact again. This procedure is done in order to simulate an interaction where two agents are keeping eye contact and then one turns his gaze to an interesting event or object.

During the learning, the robot looks for the human. However, when an object is positioned in the environment and the human turns his gaze to it, the robot loses the human attention and starts to seek anything in the environment. Additionally, if the robot looks for a toy, which the human keeps his gaze turned on it, the robot receives a reinforcement and the human gives attention to the robot, in relation to that toy. In this way, after a history of reinforcement the robot will learn to follow the human's gaze to receive his attention.

The learning capabilities of the architecture were analyzed by observing the robot interacting with the human and the environment, and computing a measure, the *correct gaze index* (CGI). The CGI measure is based on measures prosed by Whalen [20] and is defined as the frequency of gaze shifts from the human to the correct location where the human is looking at, given by:

$$CGI = \frac{\#shifts\ from\ the\ human\ to\ correct\ location}{\#shifts\ from\ the\ human\ to\ any\ location} \quad (2)$$

To quantify the learning capabilities of the architecture through the learning of gaze following, at specific points during the learning process we temporarily interrupt the learning phase to evaluate its behavior. This evaluation was done by 10 runs of 500 time units (500 seconds in the simulator). For each run, the CGI value, given by Equation (2) was computed. After the evaluation phase, the learning process was resumed. A total of 20 interrupt points were placed. The whole procedure was performed 10 times and then a mean and standard deviation was calculated for each evaluation phase.

During the evaluation phase, the human initially kept the focus on the robot until it establishes eye contact with the human, characterized by 3 time units keeping eye contact. Then two objects were positioned in the environment and the human turned his gaze for one of these objects. However, in the evaluation phase, the object to which the human should turn his gaze was place on a position given by pre-established sequence (to prevent non determinism in the results). The second object (the distractor) was placed on an empty position, obeying the probabilities defined in the social interactive simulator. Once the robot turns its head to any direction, the simulator verifies if it is looking to the correct position in the environment (a toy which the human is looking for) or not, and update the CGI measure. This procedure takes 1 time unit. Afterwards, the objects are then removed from the environment and the human turns his gaze to the robot, keeping the robot make eye contact again.

For the experiments, the architecture knowledge was set as follows. Four stimuli were declared: *face*, *object*, *attention*, and *environment*, where attention is a reinforcement stimuli. Two facts were declared to define that red and blue objects are toys. Thirteen facts where declared in order to differentiate the human's head pose in frontal pose, six poses of left profile and six poses of right profile. Additionally, two more facts were declared to define when the robot is focusing the human or a toy.

When we are dealing with JA, a fact very important that it must be considered in all interactions is the number of times that the robot establishes eye contact with human. This is an essential fact for JA. By the simulator, the robot can choose one of the options: to find anything in the environment or pay attention to human. If the robot chooses only the first option, it could not simulate the joint attention. Because this, it is important that the learning algorithm maximizes the number of established eye contacts.

The Figure 4 shows the average number of times that the robot establishes eye contact with human for each evaluation phase by using each one of the algorithms. In this figure, it is showed the beginning of the interaction between human and robot, in a total of 125 possible opportunities to establish eye contact each other.

In performing the analysis of the Figure 4 we can verify that the new FAIETGQ achieved a better final result than the other techniques.

It shows that the robot has an increasing learning in the beginning of process until find a threshold, after that, the agent no longer learns how to establish eye contact and will

 ISBN: 978-1-61208-155-7

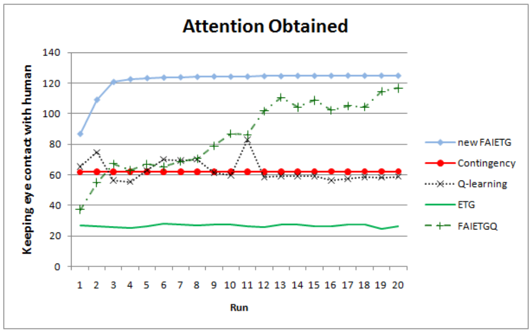

Figure 4. Average of attention obtained by human from the robot during evaluation phase.

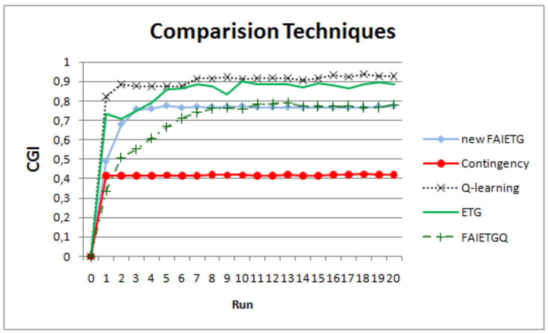

Figure 5. Learning evolution during the experiments.

only use the knowledge learned. This shows that it could learn to give attention to human. The graphics also show that the ETG technique achieves a lower level of the other algorithms and the Q-learning has irregular behavior.

Figure 5 shows the performance, the learning progress over the time, of five different learning algorithms used as learning mechanism in the architecture to solve JA. It plots the CGI average value measured for each evaluation phase, at specific points during the learning process.

Initially, it can be seen that all of the algorithms have not any knowledge about the problem. In the first run, all of them have a great improve your knowledge attaining at least 40% of maximum CGI value. In this stage, the robot or the agent learns a lot about the problem. After this, the contingency learning does not improve your knowledge until the end, remaining constant. In contrast, other algorithms have a reasonable growth until to attain a stabilization level.

In the end, the ETG and Q-learning algorithms have the best results for CGI. The new FAIETGQ and FAIETGQ algorithms have a good performance and has results close to the best. The contingency learning has result of CGI lower than the others.

A deeper analysis can be made considering Figures 4 and 5. Considering the factors of learning and the number of established eye contact, you can say that the new FAIETGQ algorithm achieved better results. This means that the robot can establish eye contact with some frequency and follows the attention to the object that is of caregiver's interest.

The experimental results also showed that the ETG and Q-learning algorithms have a high quality to follow the human gaze, but they have poor quality to establish eye contact. The ETG is the worst of this two. And finally, the contingency learning that can make half of the relations of JA, which is very low compared to other.

One good improvement can be seeing here if we compare the way as the experiment was made before. In other experiments performed so far was necessary that in the first 100 times units of learning phase, no objects are positioned in the environment and the human kept his focus on the robot the whole time, so the robot have learned that it may obtain the human attention by keeping eye contact with him. In others words, at this moment the robot explores the possibilities while the caregiver remains in the static way. This change decreases the performance of Contingency, Q-learning and ETG algorithms.

VII. Conclusion and Future works

In this paper, we presented an ongoing work for the development of a robotic architecture inspired on Behavior Analysis [7]. Five different learning algorithms, RL, contingency, ETG, FAIETGQ and this improvement on FAIETGQ proposed, were incorporated to robotic architecture to provide to the robot the ability of sharing attention. The learning mechanism were evaluated on a social interactive simulator and made by interacting real robotic head and the human in the context of the emergence of JA.

The experimental results show the evaluation by using new FAIETGQ algorithm compared with others algorithms for JA problem. It can be considered as step forward in a more natural interaction.

Future works include the insertion of new modules, for instance *energy*, *emotions*, *needs*. Another planned advance is to work with learning the joint attention by the robot where the caregiver does not have just one fixed position. So, we can better evaluate the learning method. Furthermore, it would be more natural and the robot would not need to move its trunk if the caregiver changes your position.

Acknowledgment

The authors would like to thank FAPESP, CNPq and CAPES for support received.

References

[1] F. Kaplan and V. Hafner, "The challenges of joint attention," pp. 67–74, 2004. [Online]. Available: http://cogprints.org/4067/

[2] L. Schilbach, M. Wilms, S. B. Eickhoff, S. Romanzetti, R. Tepest, G. Bente, N. J. Shah, G. R. Fink, and K. Vogeley, "Minds made for sharing: Initiating joint attention recruits reward-related neurocircuitry," *Journal of Cognitive Neuroscience*, vol. 22, no. 12, pp. 2702–2715, 2010. [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/jocn.2009.21401

[3] J. Triesch, C. Teuscher, G. O. Dek, and E. Carlson, "Gaze following: why (not) learn it?" *Developmental Science*, vol. 9, no. 2, pp. 125 – 147, 2006.

[4] R. R. da Silva and R. A. F. Romero, "Using only aspects of interaction to solve shared attention," in *Proceedings of the 3rd International Workshop on Evolutionary and Reinforcement Learning for Autonomous Robot Systems (ERLARS 2010) - The 19th European Conference on Artificial Intelligence (ECAI 2010)*, 2010, pp. 43–52.

[5] A. Catania, *Learning, Interim (4th) Edition*. Sloan Publishing, 2006.

[6] J. O. Cooper, T. E. Heron, and W. L. Heward, *Applied Behavior Analysis (2nd Edition)*. Prentice Hall, 2007.

[7] W. D. Pierce and C. D. Cheney, *Learning, Interim (4th) Edition*. Psychology Press, 2008.

[8] G. Matsuda and T. Omori, "Learning of joint visual attention by reinforcement learning," in *Int. Conf. on Cognitive Modeling (ICCM)*, 2001.

[9] Y. Nagai, A. Hosoda, and M. Asada, "A constructive model for the development of joint attention," *Connection Science*, vol. 15, no. 4, pp. 211–229, 2003.

[10] Y. Nagai, "The role of motion information in learning human-robot joint attention," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, April 2005, pp. 2069–2074.

[11] A. Shon, D. Grimes, C. Baker, M. Hoffman, S. Zhou, and R. Rao, "Probabilistic gaze imitation and saliency learning in a robotic head," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, April 2005, pp. 2865–2870.

[12] H. Kim, H. Jasso, G. Deak, and J. Triesch, "A robotic model of the development of gaze following," in *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*, aug. 2008, pp. 238–243.

[13] C. A. Policastro, R. A. F. Romero, G. Zuliani, and E. Pizzolato, "Learning of shared attention in sociable robotics," *J. Algorithms*, vol. 64, no. 4, pp. 139–151, 2009.

[14] S. Haykin, *Neural Networks - A Comprehensive Foundation*. Prentice Hall, 1999.

[15] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[16] R. O. L. Breiman, J. Friedman and C. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[17] M. Leonetti and L. Iocchi, "Improving the performance of complex agent plans through reinforcement learning," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, ser. AAMAS '10. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 723–730. [Online]. Available: http://portal.acm.org/citation.cfm?id=1838206.1838302

[18] C. A. Policastro, G. Zuliani, R. R. da Silva, V. R. Munhoz, and R. A. F. Romero, "Hybrid knowledge representation applied to the learning of the shared attention," in *IJCNN*, 2008, pp. 1579–1584.

[19] W. Dube, R. McDonald, R. Mansfield, W. Holcomb, and W. Ahearn, "Toward a behavioral analisys of joint attention," *The Behavior Analyst*, vol. 27, no. 2, pp. 197–207, 2004.

[20] C. Whalen and L. Schreibman, "Joint attention training for children with autism using behavior modification procedures," *Journal of Child Psychology and Psychiatry*, vol. 44, no. 3, pp. 456–468, 2003.