

Learning to deal with objects

María Malfaz and Miguel A. Salichs

Abstract—In this paper, a modification of the standard learning algorithm Q-learning is presented: Object Q-learning (OQ-learning). An autonomous agent should be able to decide its own goals and behaviours in order to fulfil these goals. When the agent has no previous knowledge, it must learn what to do in every state (policy of behaviour). If the agent uses Q-learning, this implies that it learns the utility value Q of each action-state pair. Typically, an autonomous agent living in a complex environment has to interact with different objects present in that world. In this case, the number of states of the agent in relation to those objects may increase as the number of objects increases, making the learning process difficult to deal with. The proposed modification appears as a solution in order to cope with this problem. The experimental results prove the usefulness of the OQ-learning in this situation, in comparison with the standard Q-learning algorithm.

Index Terms—Q-Learning, objects, decision making, autonomous agents.

I. INTRODUCTION

AN autonomous agent is a natural or artificial system in constant interaction with dynamic environments that must satisfy a set of possible goals in order to survive [1]. Moreover, according to Bellman [2], autonomy implies decision making and this implies some knowledge about the current state of the agent and its environment, including its goals. This means that the agent must have enough knowledge of itself in order to think about how to move and act in its environment, using all its properties and skills. Besides, some authors affirm that an autonomous agent has goals and motivations and it has some way to evaluate its behaviours in terms of the environment and its own motivations. Its motivations are desires or preferences that can lead to the generation and adoption of objectives. The final goals of the agent, or its motivations, must be oriented to maintain the internal equilibrium of the agent [1][3].

Learning has been denominated as one of the distinctive marks of the intelligence and introducing adaptation and learning skills in artificial systems is one of the greatest challenges of the artificial intelligence [4]. Gadanho [3] states that learning is an important skill for an autonomous agent, since it gives the agent the plasticity needed for being independent.

An autonomous agent must know what action to execute in every situation in order to fulfil its goal. In the case that this agent does not have this knowledge, the autonomous agent must learn this relation between situations and actions. The agent learns this relation by interacting with its own environment where several objects can exist. As it is going to

be shown later, learning to deal with those objects can become quite tedious. In this paper, a solution to this disadvantage is proposed.

The rest of the paper is organized as follows. In section II, a brief introduction to reinforcement learning is given. Next, in section III, one of the most commonly used reinforcement learning algorithm is presented: Q-learning. In section IV, the state is introduced as a combination of the inner and the external state of the agent and, in section V a reduced version of the state is presented. In this last section, the Q-learning algorithm is adapted in order to consider this approach and, as will be shown, this adaptation will imply some shortcomings that must be solved. The solution to this problem is proposed in section VI by considering an algorithm based on Q-learning: the Object Q-learning (OQ-learning). Next, in section VII, the experimental platform is described and later, in section VIII, the results obtained using both known algorithms in the same environment are also presented. Finally, the main conclusions of this paper and future applications are summarized in section IX.

June 4, 2009

II. REINFORCEMENT LEARNING

In a decision making process, the agent, in a certain state s , executes an action a leading him to a new state s' and generating a reinforcement r . From that new state the agent executes another action and so on. The *value* is defined as the discounted sum of all the expected reinforcements:

$$value = r_1 + \gamma \cdot r_2 + \gamma^2 \cdot r_3 + \gamma^3 \cdot r_4 + \dots \quad (1)$$

Parameter γ ($0 < \gamma < 1$) is known as the discount factor and defines how much expected future rewards affect a decision now. The goal of reinforcement learning is to maximize the total expected reward [5].

The agent that uses reinforcement learning tries to learn, through interaction with the environment, how to behave in order to fulfil a certain objective. The agent and the environment are continuously interacting, the agent selecting actions and the environment responding to those actions and presenting new situations to the agent. The environment and the proper agent also give rise to rewards that the agent tries to maximize over time. This type of learning allows the agent to adapt to the environment through the development of a policy. This policy determines the most suitable action in each state in order to maximize the reinforcement. The goal of the algorithm is to maximize the total amount of reward it receives over the long run [6].

Reinforcement learning has been successfully implemented in several virtual agents and robots [7], [8], [9], [10], [11],

M. Malfaz and M.A. Salichs are with the RoboticsLab at the Carlos III University of Madrid. 28911, Leganés, Madrid, Spain e-mail: mmalfaz@ing.uc3m.es and salichs@ing.uc3m.es

[12]. One of the most well-known reinforcement learning algorithm is the Q-learning [13], which has become one of the most used in autonomous robotics as well as in many other research areas [14].

III. Q-LEARNING ALGORITHM

The Q value is defined as the expected reward for executing action a in state s and then following the optimal policy from there. The Q-learning algorithm estimates the Q values for every state-action pair. Every $Q(s, a)$ is updated according to [15]

$$Q(s, a) = (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma V(s')) \quad (2)$$

where

$$V(s') = \max_{a \in A} (Q(s', a)) \quad (3)$$

is the value of the new state s' and is the best reward the agent can expect from the new state s' . A is the set of actions, a is every action, r is the reinforcement, γ is the discount factor, and α is the learning rate.

The learning rate α ($0 < \alpha < 1$) controls how much weight is given to the reward just experienced, as opposed to the old Q estimate [5]. This parameter gives more or less importance to the learnt Q values than new experiences. A low value of α implies that the agent is more conservative and therefore gives more importance to past experiences. If α is high, near 1, the agent values, to a greater extent, the most recent experience.

A policy π defines the behaviour of the agent. A deterministic policy $\pi : S \rightarrow \Pi(A)$ is a function that relates, with probability 1, the actions $a \in A$ that must be taken, with every state $s \in S$. The optimal policy is that one that maximizes the total expected reward.

Once the Q values are obtained, it is easy to calculate the optimal policy, $\pi^*(s)$, considering all the possible actions for a certain state and selecting the one with the highest value:

$$\pi^*(s) = \arg \max_a Q(s, a) \quad (4)$$

IV. STATE

As already stated in the introduction, it is assumed that the learning agent lives in an environment where it can interact with other objects. The agent must learn what to do in every situation in order to fulfil its goals. Let us assume that in this system the state of the agent is the combination of its inner state, S_{inner} , and its external state, $S_{external}$.

$$S = S_{inner} \times S_{external} \quad (5)$$

The inner state of the agent is related to its internal needs (for instance: the agent is hungry) and the external state is the state of the agent in relation to all the objects present in the environment (for instance: the agent has food and water):

$$S_{external} = S_{obj_1} \times S_{obj_2} \dots \quad (6)$$

This definition implies a huge number of states. For example, if there are 10 objects present in the world and it is assumed that for each object there exist 3 boolean variables: having the object, being next to the object, and knowing where the object is, we would have $2^3 = 8$ states related to every object. If the external state of the agent is its relation to all the objects, $8^{10} = 1.073.741.824$ states will exist. As previously stated, this is a huge number of states.

Moreover, if it is also assumed that there are, for example, 10 different inner states, we would have $10 * 8^{10} = 1.07 * 10^{10}$ states in total. On the other hand, considering that the number of possible actions is 30, then we would have *number of states* \times *number of actions* state-action pairs.

Therefore, if the agent uses the standard Q-learning algorithm introduced in the previous section, the number of utility values for every state-action pair, $Q(s, a)$, is $\approx 3.2 * 10^{11}$. This great number of Q values to learn presents problems since it would take a long time for those values to converge. This is because, in order to those values converge, every state-action pair must be visited by the agent a very high number of times [5].

One solution would be to use the generalization capabilities of function approximators. Feedforward neural networks are a particular case of such function approximators that can be used in combination with reinforcement learning. Although the neural networks seem to be very efficient in some cases of large scale problems, there is no guarantee of convergence [16].

Many authors have proposed several solutions to the problem of learning and planning in a large state space. According to Sprague and Ballard, this problem can be better described as a set of hierarchical organized goals and subgoals, or a problem that requires the learning agent to address several tasks at once [17]. Guestwin et al presented two algorithms for computing linear value function approximations for factored Markov Decision Processes (MDPs). The factored MDPs are one approach to represent large, structured MDPs compactly, based on the idea that a transition of a variable often depends only on a small number of other variables [18].

V. REDUCED STATE

As previously said, the definition of the state of the agent implies a huge number of different states, and the standard Q-learning seems to be not very suitable in this situation. In order to try to solve this problem, we propose a reduction of the number of states of the agent. The states related to the objects are independent of one another.

If the states related to the objects are independent, it can be considered that the external state is the state of the agent in relation to each object separately. This simplification means that the agent, at each moment, considers that its state in relation to obj_1 , for example, is independent of its state in relation to obj_2 , obj_3 , etc. Therefore, the agent learns what to do with every object separately. This simplification reduces the number of states that must be considered during the learning process of the agent.

For example, following the example presented in the previous section, for the 10 objects present in the world we would obtain $10(\text{objects}) * 8(\text{states related to every object}) = 80$ external states, those related to the objects. Finally, the total number of utility values $Q(s, a)$ would be $80(\text{external states}) * 10(\text{inner states}) * 30(\text{actions}) = 24000$, which is a great reduction.

In fact, the simplification on the learning process is even bigger if we consider that the agent learns separately what to do with every object for every inner state. For example, the agent learns what to do with food when it is hungry ($s \in S_{\text{hunger}} \times S_{\text{food}}$), what to do with food when it is thirsty ($s \in S_{\text{thirsty}} \times S_{\text{food}}$), and so on without considering its relation to the rest of objects.

Therefore, the total state of the agent in relation to each object is defined as follows:

$$s \in S_{\text{inner}} \times S_{\text{obj}_i} \quad (7)$$

From now on, the nomenclature of the Q values will be $Q^{\text{obj}_i}(s, a)$. The super-index obj_i specifies the object that the agent is dealing with, $a \in A_{\text{obj}_i}$, where A_{obj_i} is the set of actions related to object i , and s is the total state of the agent in relation to object i .

Therefore, equation (2) is adapted for the updating of the $Q^{\text{obj}_i}(s, a)$ value of the state-action pairs for an inner state and an object i :

$$Q^{\text{obj}_i}(s, a) = (1 - \alpha) \cdot Q^{\text{obj}_i}(s, a) + \alpha \cdot (r + \gamma V^{\text{obj}_i}(s')) \quad (8)$$

where

$$V^{\text{obj}_i}(s') = \max_{a \in A_{\text{obj}_i}} (Q^{\text{obj}_i}(s', a)) \quad (9)$$

is the value of object i in the new state s' , A_{obj_i} is the set of actions related to object i and s' is the new state in relation to the object i . Again, r is the reinforcement received, γ is the discount factor, and α is the learning rate.

As a consequence of this simplification, the learnt Q values, instead of being stored in a table of *total number of states* \times *total number of actions* dimension, are stored for a certain inner state and for every object in a table of *number of states related to that object* \times *number of actions related to that object* dimension.

VI. COLLATERAL EFFECTS AND OQ-LEARNING

As already stated, the considered simplification implies that the value of the actions executed in relation to a certain object are independent of its relation with the rest of objects present in the environment. This is not really true, let us consider the next example: the agent is hungry and is beside a water object, it executes the action “go for food” and at the end of this action the agent is next to food. Therefore, the agent is no longer beside the object water, so its state in relation to water has changed although the action executed was related to food.

The “collateral effects” are those effects produced by the agent on the rest of the objects when interacting with a certain

object. Therefore, in order to take into account these collateral effects, a modification of the Q-learning algorithm is proposed: The Object Q-learning.

$$Q^{\text{obj}_i}(s, a) = (1 - \alpha) \cdot Q^{\text{obj}_i}(s, a) + \alpha \cdot (r + \gamma \cdot V^{\text{obj}_i}(s')) \quad (10)$$

where

$$V^{\text{obj}_i}(s') = \max_{a \in A_{\text{obj}_i}} (Q^{\text{obj}_i}(s', a)) + \sum_m \Delta Q_{\text{max}}^{\text{obj}_m} \quad (11)$$

is the value of object i in the new state considering the possible effects of the executed action with object i , on the rest of objects. For this reason, the sum of the variations of the values of every other object is added to the value of object i in the new state, previously defined in equation (9).

These increments are calculated as follows:

$$\Delta Q_{\text{max}}^{\text{obj}_m} = \max_{a \in A_{\text{obj}_m}} (Q^{\text{obj}_m}(s', a)) - \max_{a \in A_{\text{obj}_m}} (Q^{\text{obj}_m}(s, a)) \quad (12)$$

Each of these increments measures, for every object, the difference between the best the agent can do in the new state and the best the agent could do in the previous state. In other words, when the agent executes an action in relation to a certain object, the increment or decrement of the value of the rest of objects is considered.

VII. EXPERIMENTAL PROCEDURE

A. Current application

The work presented in this paper is part of a large project whose main objective is to design a decision making system for an autonomous and social agent. The agent must learn to select the right behaviors in order to maintain its internal equilibrium. The mechanisms involved in the decision making process are inspired by those used by humans and animals.

The general idea of the proposed decision making process, see Figure 1, is that the autonomous agent has certain needs (drives) and motivations. The objective is that the agent learns how to behave in order to maintain these needs inside an acceptable range. It learns the proper action to execute in every state, that is, the policy of behavior, through its interaction with the virtual world. For this purpose, the robot/agent uses reinforcement learning algorithms to learn from its bad and good experiences.

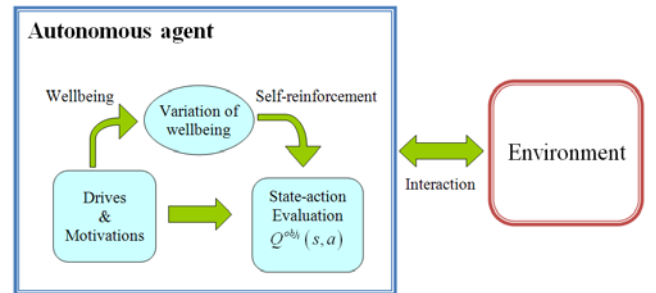


Figure 1. The decision making process

The used environment is widely described in [19]. The objects that are present in this environment are the following: food, water, medicine, box, and the world. In this environment, the food object is inside the object box and it is closed. Therefore, the agent must open the box in order to obtain food. Only when it opens the box the agent can see the food.

The boxes, the water, and the medicine are distributed in rooms in such a way that there is a room with a box, another with medicine, and another with water. It is considered that the agent has unlimited resources, i.e., the box always has food and there is an unlimited amount of water and medicine. The agent, at the beginning of its life, does not know where to find those objects. Throughout its life time, it finds the objects and remembers their position. Therefore, if the agent needs an object, it will know where to find it. This virtual world is grid-based and the agent moves around by sending 'north', 'south', 'east', and 'west' commands.

The sets of actions that the agent can execute, depending on its state in relation to the objects, are the following:

$$A_{food} = \{Eat, Get, Go\ for\}\quad (13)$$

$$A_{water} = \{Drink\ water, Get, Go\ for\}\quad (14)$$

$$A_{medicine} = \{Drink\ medicine, Get, Go\ for\}\quad (15)$$

$$A_{box} = \{Open, Go\ for\}\quad (16)$$

$$A_{world} = \{Stand\ still, Explore\}\quad (17)$$

Among all these behaviors there are some that cause an increment or decrement of some drives.

B. Inner state

As was already stated, an autonomous agent is the one that is capable of determining its own objectives and furthermore, decides which behaviors to select in order to fulfill them. Those behaviors are oriented to maintain the internal equilibrium of the agent.

The agent makes its decision based on its own motivations and drives. The internal state can be parameterized by several variables, which must be around an ideal level. When the value of these variables differs from the ideal one, an error signal occurs: the drive. The motivation is a tendency to correct the error, the drive, through the execution of behaviors. In other words, the drives are considered properties of deficit states which motivate behavior [20].

In this system, drives are related to physiological needs. The intensity of the drive increases as the need grows. Taking the hydraulic model of Lorentz [21] as an inspiration, the intensities of motivations are calculated as the combination of the intensity of the related drive (the necessity) and the presence of an external stimuli (an object). In this experiment, the drives and motivations considered are the following: Hunger, Thirst, and Weakness.

The ideal and initial value of the drives is zero. It is considered that a drive is satisfied when its value is zero, which means that there is no need.

As proposed in [22], once all the intensities of the motivations are calculated, these compete against one another. The motivation with the highest intensity is the dominant motivation and it is the one that determines the inner state, as shown in equation (18). It can happen that none of the drives of the agent has a value higher than that limit. In that case, there is not any dominant motivation and it can be considered that the agent has no needs, it is "OK".

$$S_{inner} = \begin{cases} \arg \max_i M_i \rightarrow & \text{If } \max_i M_i \neq 0 \\ OK \rightarrow & \text{In other cases} \end{cases} \quad (18)$$

In this scenario, the inner state of the agent is defined as follows:

$$S_{inner} = \{Hungry, Thirsty, Weak, OK\} \quad (19)$$

Therefore, the inner state depends on the motivations that are related to the needs of the agent, i.e., the drives. In this system, other factors that may affect the human inner state, such as psychological factors, are not considered.

C. External state

According to the definition given in section V, in this system the states related to the objects are considered to be independent of one another. This means that the agent, at each moment, considers that its state in relation to the food is independent of its state in relation to water, medicine, etc.

The state related to food, water, and medicine, is the combination of three boolean variables:

$$S_{f/w/m} = \text{Being_in_possession_of} \times \text{Being_next_to} \times \text{Knowing_where_to_find} \quad (20)$$

In the case of the box object:

$$S_{box} = \text{Being_next_to} \times \text{Knowing_where_to_find} \quad (21)$$

In our scenario, the state of the agent in relation to the world is unique, the agent is always in the world:

$$S_{world} = \text{Being_at} \quad (\text{Always True}) \quad (22)$$

D. Reinforcement function: Variation of the wellbeing

The wellbeing of the agent measures the degree of needs satisfaction. It is a function of its drives values, D_i , and some personality factors, α_i , as shown in equation (2). Therefore, when all the drives of the agent are satisfied, their values are zero and the wellbeing is maximum.

$$Wb = Wb_{ideal} - (\alpha_1 D_{hunger} + \alpha_2 D_{thirst} + \alpha_3 D_{weakness}) \quad (23)$$

where $Wb_{ideal} = 100$ is the ideal value of the wellbeing of the agent. These personality factors weigh the importance of each drive in the wellbeing of the agent. In the experiments, all the drives will have the same importance and therefore, all the personality factors are equal to one other: $\alpha_i = 1$. As the values of the drives of the agent increase as time goes on, or due to the effect of any other action, the wellbeing of the agent

decreases and it could be negative. Depending on the values of the personality factors, the increase of the drives can affect, to a certain extent, the wellbeing of the agent. Every time a drive reduction exists, there is a positive variation of the wellbeing.

The wellbeing of the agent is calculated at every simulation step, as well as its variation (ΔWb). This wellbeing variation is calculated as the current value of the wellbeing minus the value in the previous step, as shown in the next equation:

$$\Delta Wb^{k+1} = Wb^{k+1} - Wb^k \quad (24)$$

The biggest positive variation of the wellbeing will be produced when the drive related to the dominant motivation is satisfied.

At the beginning of our research, it seemed logical to think of the wellbeing as the reinforcement function, since it gives an idea about the effect of an executed action on the agent. In fact, Gadanho [23] defines a wellbeing signal generated in a similar way to ours and it was used as the reinforcement function in a reinforcement learning frame. Nevertheless, taking into account the studies carried out by Rolls [24] it seems to be more appropriate to use the variation of the wellbeing as the reinforcement function. In fact, this variation gives a clearer idea about how an action affects the wellbeing of the agent. The use of this reinforcement function makes the agent be intrinsically motivated according to the definition given by Singh et al in [25].

VIII. EXPERIMENTAL RESULTS

In this section, the experiments carried out when the agent is living alone in the virtual world using the standard Q learning as well as the OQ-learning are presented and analyzed. The goal of the agent is to learn a correct policy of behavior in order to maintain its drives inside acceptable ranges.

The performance of the agent is determined by the analysis of the wellbeing of the agent, since this information gives an accurate idea of how well the experiment go. In order to compare the performance of the agent using both reinforcement learning algorithms, two experiments are carried out.

The experiments consist of two phases: the *learning phase* (from 0 to 25000 simulation steps) and the *steady phase* (from 25000 to 30000 simulation steps).

During the learning phase, the agent starts with all the initial Q values equal to zero. The agent, through its experience in the world learns and updates its Q values, by exploring every action. It tries out actions probabilistically based on the Q values using a Boltzmann distribution [13]. At the beginning of this phase, the temperature used in that distribution is high in order to favor the exploration of every action. Along this phase this temperature decreases gradually for the exploitation of the most suitable actions. Moreover, the value of the learning rate α also decreases gradually from the value 0,3 to 0. The value of the discount factor γ is set to 0,8.

Once the learning phase has finished, the steady phase starts. In this last phase the agent “lives” according to the learnt Q values. Therefore, the value of the learning rate α is set to 0 and the temperature is very low.

The agent tries to live in the designed environment. As already stated, the food object is inside the box object. This implies that, in order to obtain food, the agent must learn to open the box and get the food. Moreover, the agent will not remember the room where it found food, just the room where the box is.

The problem presented in these experiments is that the state of the agent related to one object is independent of the rest of objects. As a consequence, if the agent does not take into account the collateral effects in its learning process, it will not be able to relate the effects of the box object on the food object.

The results presented in this paper correspond to one trial, since this is very representative of the problem.

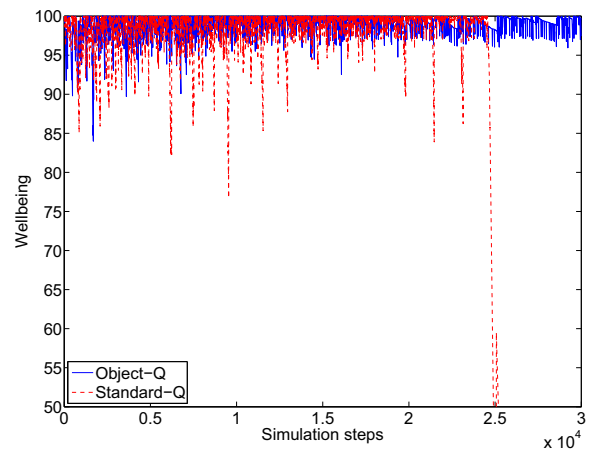


Figure 2. Wellbeing of the agent when using standard Q-learning and the OQ-learning

In figure 2 the wellbeing of the agent when using the OQ-learning as well as the standard Q-learning is shown. It can be observed that when the agent uses the standard Q-learning the wellbeing decreases in a continuous way when the steady phase starts. This is because the agent is not able to satisfy its drive Hunger, as expected.

On the other hand, when the agent uses the OQ-learning its wellbeing is very high, very near to the ideal value, during both phases of its life. This fact shows that all the drives are maintained with low values during the whole experiment. This indicates that the agent is able to learn the right sequence of actions in order to satisfy all its needs.

The reason why the agent is not able to maintain its wellbeing inside acceptable ranges, while using the standard Q-learning, is that the agent is not able to learn the right sequence of actions when it is hungry. If the agent had considered the collateral effects of the box on the food, the agent would have learnt to go where the box is, would have opened it and after that, would have gotten and eaten the food. On the contrary, as shown by the results, the agent does not value the actions related to the box. Therefore, according to (8), although the values of the actions related to food are high, the agent is not able to find food. This is because when evaluating the action “open box” the algorithm does not consider the new

state related to food: “being next to food”, whose value is very high, since the agent, from that state, can get the food. In the case of the action “go for box”, its value is very low since the value of the new state “being next to box” is also low due to the value of the action “open box”.

On the contrary, when the agent uses the OQ-learning algorithm, it happens that when the agent evaluates an action related to an object, it considers the collateral effects on the rest of objects. When using the OQ-learning, the values of the actions related to the box are high since the agent values, as the collateral effect, that when it opens the box, it is then next to food. Therefore, due to the high value of the action “get food”, the value of the new state after opening the box, that is, “being next to food”, is also high.

IX. CONCLUSIONS AND FUTURE WORKS

The OQ-learning appears as an effective learning algorithm in cases where the learning agent has to deal with objects present in its environment. Instead of considering the state of the agent in relation to every object at the same time, since this implies a huge number of states, a reduced state is introduced. The state of the agent in relation to one object is independent of its state in relation to the rest of objects. Therefore, the agent learns what to do with every object separately. Following this approach and applying the standard Q-learning, the agent ignores the possible collateral effects of the actions related to one object on the rest of objects. The OQ-learning solves this problem by considering in the value of one object, the increase or decrease of the value of rest of objects as a consequence of an action executed in relation to that object.

In the presented scenario, the agent must learn to open the box in order to get food. Using the standard Q-learning, this is impossible since the algorithm does not consider the collateral effect of the action “open the box”: “to be next to food”. Therefore, the value of the box object will never become high because the algorithm does not consider the effect on the food object. Nevertheless, we have proved that using the OQ-learning the value of the box object becomes high and the agent learns that it must open the box in order to get food.

In the next future, the decision making system introduced in this paper is intended to be implemented on a real robot. The final goal of our research is to build an autonomous and social robot, so that learning in a complex environment would be one of its basic features. Currently, we have developed a robotic platform for human-robot interaction: Maggie. Maggie is a social robot developed at RoboticsLab and is completely described in [26].

ACKNOWLEDGMENT

The authors gratefully acknowledge the funds provided by the Spanish Government through the project “Peer to Peer Robot-Human Interaction (R2H)”, of MEC (Ministry of Science and Education) and the project “A new Approach to Social Robotics (AROS)”, of MICINN (Ministry of Science and Innovation).

REFERENCES

- [1] L. Cañamero, *Emotions in Humans and Artifacts*. MIT Press, 2003, ch. Designing emotions for activity selection in autonomous agents.
- [2] K. L. Bellman, *Emotions in Humans and Artifacts*. MIT Press, 2003, ch. Emotions: Meaningful mappings between the individual and its world.
- [3] S. Gadoh, “Reinforcement learning in autonomous robots: An empirical investigation of the role of emotions,” Ph.D. dissertation, University of Edinburgh, 1999.
- [4] M. Mataric, “Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior,” *Trends in Cognitive Science*, vol. 2(3), pp. 82–87, 1998.
- [5] M. Humphrys, “Action selection methods using reinforcement learning,” Ph.D. dissertation, Trinity Hall, Cambridge, 1997.
- [6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, A Bradford Book, 1998.
- [7] C. Isbell, C. R. Shelton, M. Kearns, S. Singh, and P. Stone, “A social reinforcement learning agent,” in *the fifth international conference on Autonomous agents, Montreal, Quebec, Canada*, 2001.
- [8] E. Martinson, A. Stoytchev, and R. Arkin, “Robot behavioral selection using q-learning,” in *of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), EPFL, Switzerland*, 2002.
- [9] B. Bakker, V. Zhumatiy, G. Gruener, and J. Schmidhuber, “A robot that reinforcement-learns to identify and memorize important previous observations,” in *the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2003*, 2003.
- [10] C. H. C. Ribeiro, R. Pegoraro, and A. H. RealCosta, “Experience generalization for concurrent reinforcement learners: the minimax-q algorithm,” in *AAMAS 2002*, 2002.
- [11] A. Bonarini, A. Lazaric, M. Restelli, and P. Vitali, “Self-development framework for reinforcement learning agents,” in *the 5th International Conference on Developmental Learning (ICDL)*, 2006.
- [12] A. L. Thomaz and C. Breazeal, “Transparency and socially guided machine learning,” in *the 5th International Conference on Developmental Learning (ICDL)*, 2006.
- [13] C. J. Watkins, “Models of delayed reinforcement learning,” Ph.D. dissertation, Cambridge University, Cambridge, UK, 1989.
- [14] C. Touzet, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 2003, ch. Q-learning for robots, pp. 934–937.
- [15] W. D. Smart and L. P. Kaelbling, “Effective reinforcement learning for mobile robots,” in *International Conference on Robotics and Automation (ICRA2002)*, 2002.
- [16] J. A. Boyan and A. W. Moore, “Generalization in reinforcement learning: Safely approximating the value function,” in *NIPS*, 1994, pp. 369–376.
- [17] N. Sprague and D. Ballard, “Multiple-goal reinforcement learning with modular sarsa(0),” in *the 18th International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico*, 2003.
- [18] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman, “Efficient solution algorithms for factored mdps,” *Journal of Artificial Intelligence research (JAIR)*, vol. 19, pp. 399–468, 2003.
- [19] M. Malfaz and M. Salichs, “Learning behaviour-selection algorithms for autonomous social agents living in a role-playing game,” in *AISB’06: Adaptation in Artificial and Biological Systems. University of Bristol, Bristol, England, April 2006*.
- [20] C. L. Hull, *Principles of Behavior*. New York: Appleton Century Crofts, 1943.
- [21] K. Lorenz and P. Leyhausen, *Motivation of human and animal behaviour; an ethological view*. New York: Van Nostrand-Reinhold, 1973, vol. XIX.
- [22] C. Balkenius, “Natural intelligence in artificial creatures,” Ph.D. dissertation, Lund University Cognitive Studies 37, 1995.
- [23] S. Gadoh and L. Custodio, “Asynchronous learning by emotions and cognition,” in *From Animals to Animats VII, Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior (SAB’02), Edinburgh, UK*, 2002.
- [24] E. Rolls, *Emotions in Humans and Artifacts*. MIT Press, 2003, ch. Theory of emotion, its functions, and its adaptive value.
- [25] S. Singh, A. G. Barto, and N. Chentanez, “Intrinsically motivated reinforcement learning,” *Advances in Neural Information Processing*, vol. 18, 2004.
- [26] M. Salichs, R. Barber, A. Khamis, M. Malfaz, J. Gorostiza, R. Pacheco, R. Rivas, A. Corrales, and E. Delgado, “Maggie: A robotic platform for human-robot social interaction,” in *IEEE International Conference on Robotics, Automation and Mechatronics (RAM 2006). Bangkok, Thailand*, 2006.