# Learning Behaviors by an Autonomous Social Robot with Motivations

Álvaro Castro-González, María Malfaz, J.F. Gorostiza, Miguel A. Salichs

**Abstract**

In this paper an autonomous social robot is *living* in a laboratory where it can interact with several items (people included). Its goal is to learn by itself the proper behaviors in order to maintain its *wellbeing* as high as possible. Several experiments have been conducted to test the performance of the system.

The Object Q-Learning algorithm has been implemented in the robot as the learning algorithm. This algorithm is a variation of the traditional Q-Learning since it considers a reduced state space and *collateral effects*. The comparison of the performance of both algorithms is shown in the first part of the experiments. Moreover, two mechanisms intended to reduce the learning session durations have been included: Well-Balanced Exploration and Amplified Reward. Their advantages are justified in the results obtained in the second part of the experiments.

Finally, the behaviors learned by our robot are analyzed. The resulting behaviors have not been pre-programmed. In fact, they have been learned by real interaction in the real world, and are related to the motivations of the robot. These are *natural* behaviors in the sense that they can be easily understood by humans observing the robot.

**Index Terms**

autonomy, reinforcement learning, behavior, decision making, social robot, motivations, motivational behavior.

The authors are with the RoboticsLab at Carlos III University of Madrid, 28911, Leganés, Madrid, Spain e-mail: acgonzal@ing.uc3m.es, mmalfaz@ing.uc3m.es, jgorosti@ing.uc3m.es, salichs@ing.uc3m.es

# Learning Behaviors by an Autonomous Social Robot with Motivations

## I. Introduction

It is expected that, in a near future, robots interacting with humans will be as common as computers at home. In these situations, robots and humans will share the same areas. Therefore, during the last few years, the interest in robots integrated in our everyday environment, i.e. personal and social robots, has increased (Kubota, Nojima, Baba, Kojima, and Fukuda 2000). Since they must interact with humans, an efficient Human-robot interaction (HRI) is one of the main characteristics of these robots. In order to facilitate it, these robots must exhibit natural behaviors, i.e. behaviors which can be easily understood by people, in an autonomous manner.

An autonomous robot acts on the basis of its own decisions (Matarić 2007) in order to fulfill its goals. Thus, it must know what action to execute in each situation. In the case that this robot does not have this knowledge, it must learn this relation between situations and actions.

Learning is a cognitive ability that provides the plasticity for adapting to new situations (Gadanho 1999). Then, this is a key element for autonomy, mainly when dealing with high non-deterministic environments, like the real world. Lorenz defined learning as the adaptive changes of behavior and this is, in fact, the reason why it exists in animals and humans (Lorentz 1987). Living beings react to sensory input coming from their environment. Some of these living beings change their reactions as time goes by: given the same input (sensorial perception), the reaction may be totally different. They are able to learn and update their reactions. Learning algorithms try to imitate this ability and to explain how and why the reactions change over time.

Most of the robots existing in unstructured environments require to be as autonomous as possible. This autonomy is related to the selection of actions during the robot's *life*. The robot self-governs its behavior through the policy that determines the next action to be executed at each moment. This policy can be acquired by two different manners:

1) The policy is assigned and the robot follows this pre-designed policy.
2) The robot learns the best policy according to certain requisites.

In the first case, the policy is defined by others and it is imposed to the robot. In these situations, the available decisions of the robot are pre-programmed and limited. In order to obtain an optimal policy, all situations and possibilities should be considered in the policy. However, in unpredictable environments, like real scenarios where the robots and people coexist, this is a tedious task and sometimes it cannot be tackled.

Learning does not restrict the possible decisions but provides a flexible mechanism to adapt the robot's behavior to new or unforeseeable events. Then, learning perfectly fits the needs of the exploration of uncharted *worlds*, or situations.

In this paper, an autonomous social robot without previous knowledge is *living* in a laboratory where it can interact with several items (people included). Its goal is to learn by itself from scratch the proper behaviors in order to maintain its *wellbeing* as high as possible. Furthermore, learning must be achieved in a reasonable amount of time by interacting with the real world. Consequently, Reinforcement Learning (RL from now on) perfectly fulfills all the requisites previously presented. In RL, the teaching signal informs about the appropriateness of the response by means of the reward or reinforcement signal. It looks for a state-action mapping which maximizes the reward. The reinforcement signal just informs about whether the output is correct or incorrect and how good or bad it is. In particular, the Object Q-Learning algorithm has been implemented. Besides, two mechanisms intended for speeding up the learning process have been included. Several experiments have been conducted in order to test the performance of the system.

The rest of the paper is organized as follows: next, the related works which have inspired this work are presented (Section II); then, in Section III, the robotic platform is presented and its decision making system is described. After that, the learning algorithm (Section IV) and how it has been boosted (Section V) is explained. Section VI details the configuration of the decision making system during the experiments. All the experimental results have been included in Section VII. Finally, the results are discussed and some conclusions are extracted in Section VIII.

## II. Related works

Several works have shown how RL can be used for learning composite task; that is, the robot is endowed with a set of primitive actions and it learns how to organize them to achieve a complex behavior. Mahadevan and Connell (Mahadevan and Connell 1992) applied RL in real robots which were able to learn different behaviors for pushing boxes. Maes and Brooks (Maes and Brooks 1990) developed a 6-legged robot which learned to coordinate different actions for each leg in order to achieve a stable gate. Martinson (Martinson, Stoytchev, and Arkin 2001) developed a simulation where he achieves a behavioral coordination mechanism for an anti-tank mine robot. All these behaviors were related to low level actions for very specific tasks, and the reward signal comes from the external world: the quantity of meters the boxes have been moved, the distance the robot has walked forward, or whether the tank is *destroyed*.

Works where the learning signal comes from internal variables, some times referred as motivations, are less frequent. Blumberg (Blumberg, Todd, and Maes 1996) uses its motivational variables as the reinforcement signal for learning the behavior for each situation. These signals are independently employed, so the behaviors for each motivational variable are separately learned. This might result on situations where a certain behavior is appropriate for certain motivational variable, but rather detrimental for others. In contrast, Gadanho (Gadanho and Hallam 1998; Gadanho 1999; Gadanho and Hallam 2001) considers a broader measure of *satisfaction* as the reinforcement signal: the wellbeing, which depends on all the homeostatic variables and other values. This avoids the potential detrimental effects of Blumberg's approach. A similar idea has been considered in our system.

Barto and Singh (Barto, Singh, and Chentanez 2004; Singh, Barto, and Chentanez 2005) introduced the concept of intrinsically motivated agent's actions. That is, those actions that the agent is engaged in them for its own sake rather than trying to solve a particular external problem. Then, intrinsically motivated learning is driven by internal rewards rather than externally-directed goals. They combine intrinsically motivated learning and RL for constructing hierarchies of reusable skills that are applied to a simple artificial *playroom*. Following this line of research, Kaplan and Oudeyer (Kaplan and Oudeyer 2007; Oudeyer, Baranes, and Kaplan 2013) presented an intrinsic motivation system that can shape the developmental trajectories of a robot. The experiments presented by these authors show how a robot is able to learn how to use sensorimotor primitives to alter its surrounding environment resulting on complex self-organized developmental trajectories. Starzyk (Starzyk 2010) also considers motivated learning but he considers abstract motivations and abstract goals. For example, an abstract pain symbolizes insufficient resources that the machine needs, and it is motivated to discover new ways to find those resources. Many researchers link intrinsic motivations with concepts such as novelty, curiosity, surprise (Bolado-Gomez and Gurney 2013; Gurney, Lepora, Shah, Koene, and Redgrave 2013), and habituation (Gatsoulis, Burbridge, and McGinnity 2012), and they use them to guide the learning process and so improve it. All these works use motivations (in particular intrinsic motivations) as a mechanism to improve learning. However, as stated by Barto (Barto 2013), *"not all aspects of motivation involve learning"*. The system proposed by the authors considers motivations but they do not guide the learning process.

In relation to an efficient learning process, Thrun (Thrun 1992) already remarked the importance of the exploration during learning. He describes several techniques for exploration in finite, discrete domains like the one proposed in this work. He classifies exploration in two categories: undirected, where actions are selected based on randomness (usually this is inefficient in learning time), and directed, where exploration specific knowledge guides the exploration. In relation to the directed exploration, many works have been presented. Thrun presented the *counter-based exploration* which follows the rule "go to the least occurred adjacent state"and it was applied to simple, virtual worlds. More recent works use cognitive concepts to guide the exploration during learning. For example, (Bolado-Gomez and Gurney 2013) and (Gurney, Lepora, Shah, Koene, and Redgrave 2013) use *repetition bias* to explore novel objects, which are related with surprising outcomes. Then, the actions resulting on unpredicted outcomes are more repeated. In the work presented in (Lopes, Lang, Toussaint, and Oudeyer 2012), the exploration is driven to those areas of the state space where learning progress can indeed be made. These techniques have inspired the *Well-Balanced Exploration* (Section V-A) which is applied in this work to a real environment.

Other common strategy for reducing the learning time, is the reduction of the state space. Many authors have proposed several solutions to deal with this problem. One solution would be to use the generalization capabilities of function approximators such as feedforward neural networks combined with reinforcement learning although there is no guarantee of convergence (Boyan and Moore 1995). 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 (Sprague and Ballard 2003). In (Guestrin, Koller, Parr, and Venkataraman 2003) and (Vigorito and Barto 2010) the learning process is accelerated by structuring the environment using factored Markov Decision Processes (FMDPs), based on the idea that a transition of a variable often depends only on a small number of other variables. In (Li, Walsh, and Littman 2006), the authors present a review of other approaches which propose a state abstraction, or state aggregation, in order to deal with large state spaces. Abstraction can be thought of as a process that maps the original description of a problem to a much compact and easier one to work

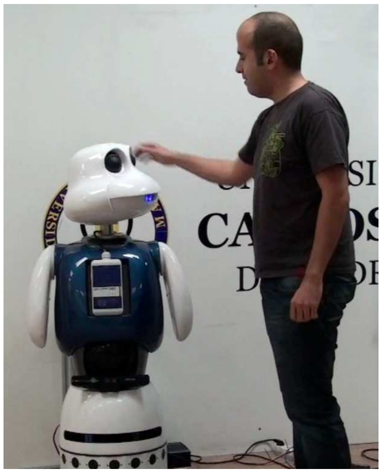


Fig. 1. The social robot Maggie during the experiments

with. In these approaches the states are grouped together if they share, for example, the same probability transition and the reward function (Boutilier, Dearden, and Goldszmidt 2000; Givan, Dean, and Greig 2003), or the same optimal action, or similar Q-values (McCallum 1996). In this work the applied method follows the idea proposed by some of the authors in (Malfaz 2007): the states related to different objects are going to be treated as if they were independent of one another (Section IV-A).

## III. The robot Maggie and its Decision Making System

The work presented in this paper has been implemented in the research robotic platform named Maggie (Salichs, Barber, Khamis, Malfaz, Gorostiza, Pacheco, Rivas, Corrales, Delgado, and Garcia 2006). Maggie is a social and personal robot intended to perform research on HRI and improving robots autonomy (Figure 1). It is controlled by the Automatic-Deliberative architecture (Barber and Salichs 2002; Barber 2000; Barber and Salichs 2001; Rivas, Corrales, Barber, and Salichs 2007; Malfaz and Salichs 2011) where the elemental component is the skill. Skills endow the robot with different sensory and motor capacities, and process information. These skills are coordinated by a Decision Making System (DMS) based on drives, motivations, emotions, and self-learning.

In our approach, the autonomous robot has certain needs (or drives) and motivations. Drives range from $0$, no need, to a maximum value, the saturation value. The intensities of the motivations of the robot are modeled as a function of its drives and some external stimuli. The general idea is that, for example, we are motivated to eat when we are hungry and also when we have food in front of us, although we do not really need it. The motivations compete among themselves for being the dominant one (i.e. the highest motivation). The dominant motivation determines the inner state of the robot.

In this work, the DMS and its learning process are intended for acquiring the right relationship between states and actions. That is, to learn the best action to execute in every state in order to maximize its wellbeing (by satisfying its drives). In order to do it, the robot learns how to behave in order to maintain its needs (the drives) within an

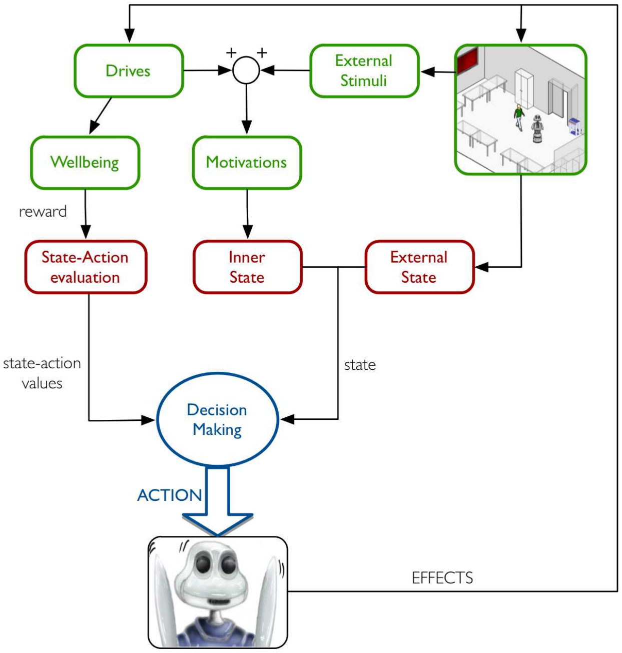


Fig. 2. Decision making system and how its elements are related to each other.

acceptable range. For this purpose, it uses RL algorithms to learn from its bad and good experiences (section IV). The reward signal is related to the wellbeing of the robot. This wellbeing is defined as a function of all its drives and it measures the degree of satisfaction of its internal needs. As the values of the needs of the robot increase, its wellbeing decreases.

In this proposed DMS, the variation of the wellbeing of the robot is used as the reward signal during the learning process. This means that an increment in the robot's wellbeing is a positive reward, and a reduction means a negative reward.

The outline of the decision making elements can be seen in Figure 2. Motivations determine the internal state. Together with the state related to the objects in the robot's environment (i.e. the external state), both determines the state which is used to make a decision. After an action is selected and executed, its consequences affect to the world where the robot is "living" and to its drives. Thus, the wellbeing is affected and used as the rating to evaluate the performance of an action in a state. This experience is considered in future decision making.

## IV. Learning

As mentioned in Section I, learning is a possible solution to dynamic environments where responses to all different situations can not be pre-programmed or predefined. An example of dynamic environment is the changing surroundings where our robot lives.

The aim is that our robot learns complex behaviors understood as a sequence of actions. Those complex behaviors optimize the adaptation of the robot to its dynamic environment. Moreover, since our robot is intended to interact with people (social robot), these complex behaviors must be a natural response to situations where people can be involved. This means that these behaviors are desired to be comprehensible because we do not desire that people avert HRI due to “weird” behaviors.

In this work, learning is achieved by RL algorithms which are appropriate to deal with motivational systems (Barto 2013). The approach adopted in this work is a model-free approach because the system knows neither the consequences of executing an action (the next state) nor the reward that will be obtained. Initially, it just knows the actions that can be executed with each object.

The learning process implemented in this work is based on two key points:

1) A reduction of the state space
2) The Object-Q-Learning and the collateral effects

which will be explained below.

The Object-Q-Learning Algorithm was extensively detailed in (Malfaz and Salichs 2009; Malfaz and Salichs 2010). In this section, the algorithm is summarized in order to provide enough knowledge to clearly understand the rest of this paper.

### *A. The reduced state space*

In this work, it is assumed that the robot lives in an environment where it can interact with objects. The goal of the autonomous robot is to learn what to do in every situation in order to survive and to maintain its needs satisfied. In this system, the state of the agent $s \epsilon S$ is the combination of its inner state and its external state. The inner state of the robot is related to its internal needs (for instance: the robot *needs* to recharge its battery so the dominant motivation is survival) and the external state is its state in relation to all the objects present in the environment. In this approach, the external state considers each object separately (Castro-González, Malfaz, and Salichs 2011). This means that the robot, at each moment, considers that its state in relation, for example, to $obj_1$ is independent from its state in relation to $obj_2$, $obj_3$, etc. so the robot learns what to do with every object by separate. This simplification reduces the number of states that must be considered during the learning process of the robot.

Using this simplification, the robot learns what to do with every object for every inner state. For example, the robot would learn what to do with the docking station when it needs to recharge without considering its relation to the rest of objects.

### *B. Object-Q Learning and Collateral Effects*

The simplification made in order to reduce the state space considers the objects in the environment as if they were independent. This assumption implies that the effects resulting from the execution of an action, in relation to a certain object, do not affect to the state of the robot in relation to the rest of objects. Let us give an example: if the robot decides to move towards the music player (an interactive object in the robot’s environment), this action will not affect to the state in relation to the rest of objects. Nevertheless, if the robot was previously recharging its battery in the docking station, this action (to go to the music player), which is related to the object music player, has affected to its state in relation to the docking station. Moreover, if a person is nearby the robot, after it moves, this person is not present anymore. As result, an action (approaching the music player) related to a particular object (the music player) may influence its state in relation to other items (the docking station and a person). These are known as collateral effects.

Therefore, in order to take into account these collateral effects, the Object Q-learning considers how the action related to a particular object affects to the rest of the objects. Using this viewpoint, the Q values are updated according to Equation (1). $Q$ values can be interpreted as a measure of how suitable is to execute action $a$ in state $s$.

$$Q^{obj_i}(s,a) = (1-\alpha)\cdot Q^{obj_i}(s,a) + \alpha \cdot \big(r + \gamma \cdot V^{obj_i}(s')\big) \tag{1}$$

where:

$$V^{obj_i}(s') = \max_{a \in A_{obj_i}} \big(Q^{obj_i}(s',a)\big) + \sum_{m \neq i} \Delta Q^{obj_m}_{\max} \tag{2}$$

The super-index $obj_i$ indicates that the learning process is made in relation to the object $i$; therefore, $s \in S_i$ is the state of the robot in relation to the object $i$, $A_{obj_i}$ is the set of the actions related to the object $i$ and $s' \in S_i$ is the new state in relation to the object $i$. Parameter $r$ is the reinforcement received, $\gamma$ is the discount factor, and $\alpha$ is the learning rate.

Moreover, $V^{obj_i}(s')$ is the value of the object $i$ in the new state $s'$ considering the possible effects of the action $a$ executed with the 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 the object $i$ in the new state.

These increments are calculated as follows in Equation (3).

$$\Delta Q^{obj_m}_{\max} = \max_{a \in A_{obj_m}} \big(Q^{obj_m}(s',a)\big) - \max_{a \in A_{obj_m}} \big(Q^{obj_m}(s,a)\big) \tag{3}$$

Each of these increments measures, for every object ($obj_m \neq obj_i$), the difference between the best the robot can do in the new state, and the best the robot could do in the previous state. In other words, it measures if the value of the new state is better or worse than the value of the previous state in relation to each object.

## V. Enhancing the learning process

As previously exposed, learning is achieved by the robot through interaction in the real world of a laboratory. Moreover, during learning, the actions are randomly selected. This random selection is based on the theory that all situations must be experienced an infinite number of times for the learning algorithm to achieve convergence. This leads to unfeasible experiments in terms of their duration.

In order to be able to carry out full learning sessions, the reduced state space and the Object Q-Learning have been considered. However, this is not enough for experiments in the real world. Consequently, two additional mechanisms have been included:

1) Well-balanced Exploration
2) Amplified Reward

Both are intended for speeding up the learning process by reducing the duration of the learning sessions. Following, they are analyzed.

### A. Well-balanced Exploration

During exploration, due to the random selection of actions, some states can remain unexplored for long periods of time. In order to solve this problem, from time to time, these unexplored states are enforced to be discovered. This is a kind of directed exploration as mentioned in Section II.

This idea is implemented in this work and it is exposed in Figure 3: at some point, the robot is forced to a new state $s'$ which has not been visited enough in comparison with other states. By means of this mechanism, we assure that all states are visited a minimum number of times. This "guided" transition cannot be considered as an iteration in the learning process because it is not the "natural" result of an action selected by the robot itself.

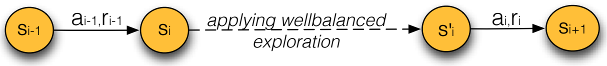


Fig. 3. Well-balanced Exploration schematic

This idea has to be applied to the particular state space of this work. Considering the ideas presented in Section IV-A, the state of the robot is composed of internal and external states. The inner state is determined by the dominant motivation at each iteration. The motivations grow due to the drive linked to each one or to the external stimuli. As a result of the random selection of actions during learning, it could happen that the required external stimuli for a particular motivation are never presented or attained; or actions that satisfy a drive are always executed when its associated motivation is not the dominant one. Moreover, drives evolve at different rates. Thereupon, the motivations associated to the slowest drives are less likely to become the dominant motivation. For all these reasons, the proper behaviors that have to be exhibited with some "slow" motivations could not be properly learned in a reasonable amount of time.

For promoting these "slow" motivations, every $f$ iterations, the least frequent dominant motivation is promoted. Promoting a motivation means that the drive linked to the motivation is artificially saturated. This implies that the drive value is set to its maximum value. Therefore, the promoted motivation will easily reach the dominance over the rest of the motivations. As a consequence, the new state is likely to be related to this promoted motivation and then the corresponding behavior will be explored and learned.

As aforementioned, when a motivation is promoted, the transition from the previous state to the new situation where its drive is artificially saturated is not considered by the learning algorithm. Otherwise, unreal effects of actions would have been taken into account and included in the learned policy.

The whole process is schematized in Algorithm 1.

**Algorithm 1** Well-balanced Exploration: promoting motivations

```
Require: $iter$ ← total number of iterations
Require: $f$ ← frequency to promote the least frequent dominant motivation
while robot is learning do
    if $iter \mod f = 0$ then
        $m$ ← least frequent dominant motivation
        $d$ ← drive associated to m
        $d$ is saturated                                ▷ promoting motivation
        Set flag to ignore this iteration at learning
    end if
    $iter = iter + 1$
end while
```

Promoting motivations forces to explore all the possible internal states (dominant motivation) an acceptable number of times, so the exploration of dominant motivations is balanced. Thus, the experiment length can be drastically reduced as it will be shown in Section VII-B2.

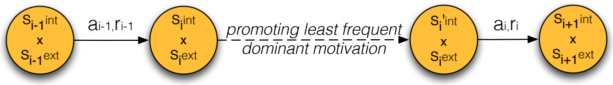

$s_{i-1}^{int}$ x $s_{i-1}^{ext}$
$a_{i-1},r_{i-1}$
$s_i^{int}$ x $s_i^{ext}$
promoting least frequent dominant motivation
$s_i'^{int}$ x $s_i^{ext}$
$a_i,r_i$
$s_{i+1}^{int}$ x $s_{i+1}^{ext}$

Fig. 4. Well-balanced Exploration applied to the internal state

In this work, Well-balanced Exploration has been applied considering just unusual internal states (Figure 4). External states are explored enough and this technique has not been applied to them.

### *B. Amplified Reward*

In order to identify as fast as possible the actions that satisfy the robot's needs, the Amplified Reward has been implemented. Living beings have been taken as the source of inspiration. Focusing on human beings, when a person is hungry and eats, the benefit is really great. However, if this person is really thirsty and also hungry, eating does not provide the same level of benefit, but a smaller one. The benefits coming from satisfying the most urgent need is always the greatest one. This is the idea behind the Amplified Reward mechanism.

In the interest of fostering this idea, positive rewards are amplified when the reward comes from correcting the drive corresponding to the dominant motivation. By means of back-propagation and the collateral effects, this amplified reward is transferred to the rest of the actions involved, even when several objects are concerned. Therefore, all the actions required to satisfy a drive will be proportionally amplified and the behavior related to its motivation will be learned faster.

Considering the previous ideas, the amplification is applied when the variation of wellbeing (the reward) is positive, and this benefit is due to the reduction of the drive connected to the dominant motivation (the most urgent need). Mathematically, it is expressed as Equation (4).

$$\text{If } \Delta_a D_{dm} < 0 \ \& \ r_a > 0 \text{ then } r \leftarrow r_a \cdot f_a \tag{4}$$

where $\Delta_a D_{dm}$ is the variation of the drive related to the dominant motivation after executing action $a$; parameter $r_a$ means the reward obtained when action $a$ has finished (this is the wellbeing variation); and $r$ is the reward used by the learning algorithm. Finally, $f_a$ is the amplification factor which determines the amount of augmentation applied to the reward. Then, after action $a$ has been executed, the obtained reward $r_a$ is amplified if it positively affects the dominant motivation.

## VI. Decision making system experimental setup

The aim of the presented DMS is to achieve an autonomous robot which learns to make right decisions. Once the learning process has finished, the most appropriated action at each moment will be selected by the decision making module. Choosing the right action depends on the value of the motivations, on previous experiences, and on the relationship with the environment. All these elements have been modeled in order to be processed by the implemented DMS.

All the parameters considered in this implementation shape a specific robot's "personality". That is, the DMS setup defines the robot's behavior during its lifespan. Changing these parameters, new "personalities" or behaviors are exhibited by the robot. The parameters which are presented in the next sections have been defined at design time by the authors.

### *A. The robot's inner world: what drives and motivations?*

This section details all the inner variables and parameters of the DMS. As mentioned, the robot's needs, the drives, are represented as an internal value. The choice about what drives (and consequently motivations too) must be implemented were made at design time considering the utility and functionality of the robot. The number of drives and motivations should be flexible and correlated to the tasks to perform (Bryson and Tanguy 2009; Kowalczuk and Czubenko 2011).

All things considered, following, the selected drives and motivations (each motivation is connected to a drive) are listed:

- **Energy:** this drive is necessary for survival and it refers to the energy dependence. It is linked to the battery by following its level. Its associated motivation is **survival**.
- **Boredom:** it is defined as the need of fun or entertainment. This drive can be satisfied when Maggie is having fun and this is achieved when it dances. It is related to the motivation of **fun**.

- **Loneliness:** this is the lack of social interaction and, then, the need of companion. The **social** motivation is related to this drive. As presented before, Maggie is a social robot so one of its main goals is to establish relationships with people. This attitude is enforced by this motivation.
- **Calm:** this is the need of peace and its associated motivation is **relax**. The *relax* motivation searches for noiseless conditions.

Drives represent the deviation from the ideal state. This ideal state corresponds to the value zero for all drives (no needs).

In addition, it could happen that none motivation can be considered as the dominant one. This situation is also contemplated in the proposed system and, consequently, the most convenient behavior for this situation will be learned and studied too. This situation is referred as **none** or **non-motivation**.

Just like human beings can become thirsty when they see water, the motivations are influenced by some objects when they are present in the environment. These are called the **external stimuli** and they will be detailed in the next section.

### *B. The external world: sensing and acting*

The world is perceived by the robot in terms of objects and the states related to these objects (the external state). In this work, the world where Maggie is living in is limited to the laboratory and the following objects: a music player, the music in the lab, the docking station for supplying energy, and the people living around the robot. Also the states related to all these items have to be defined and the transitions between states are detected by several skills running in Maggie.

Moreover, the robot interacts with its environment through the actions that can be performed with the objects. The robot has a repertory of actions and it has to learn when to execute each of them.

In Figure 5, the states related to each object, the actions, and the transitions from one state to another are shown. If an action does not appear at one state, it means that it cannot be executed from that state; e.g., Maggie cannot *play music* if it is *far* from the player; or it cannot *interact* with a person if it is alone.

Following, the available items, the states related to them, and their actions are introduced.

*1) Music player:* Maggie is able to operate a music player located in the lab (Salichs, Castro-González, and Salichs 2009). In order to operate the music player, the robot has to be located at a certain distance and facing the appliance. Therefore, in relation to the position of the robot, there are two states: *near*, when the robot is close enough to operate the player, and *far*, if the robot is in a position where it is not able to operate the player.

Moreover, related to the operational state of the music player, other two states have to be distinguished to avoid sending the same command twice to the player: *near-on* and *near-off*. When the robot is close to the player and it is already turned on, the state is *near-on*; but, when the robot is also close and the player is off, the state is *near-off*.

The possible actions with the item *music player* are:

- Go to player: Maggie approaches the music player.

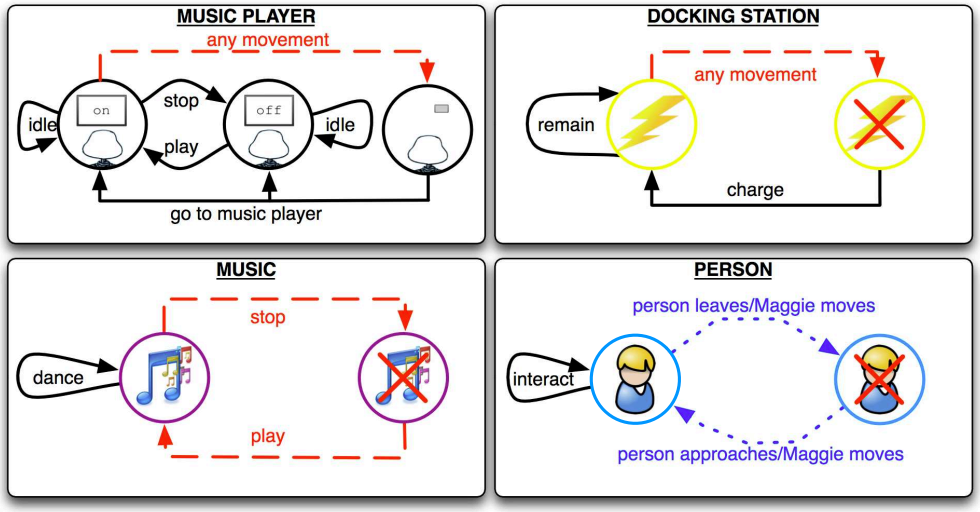


Fig. 5. States, actions and transitions related to the items of the robot's environment: a music player, the docking station, the music, and a person. Round sides rectangles represent the states related to each object, the arrows are the transitions, and the labels of the arrows are the actions which may cause the transition if no errors occur. Black arrows correspond to transitions triggered by actions executed with the object. Red dashed arrows mean transitions activated by actions with other objects. And purple dotted arrows are dedicated to transitions due to actions executed by other agents

- Play music: music is played because it turns the player on when it is off.
- Stop music: music is stopped when it is being played because the music player is turned off.
- Idle: it represents the possibility to remain next to the player for a while.

*2) Music:* The robot's environment is the lab, and *music* can be playing there. Then, the robot can be *listening*, or *not*, to *music*.

About the *music*, there is just one possible action:

- Dance: the robot moves its body with the music. This action just can be executed when Maggie is *listening* to music.

*3) Docking station:* The *docking station* is the source of energy. If the robot is *plugged*, the battery is charging, so its level increases. Otherwise, the robot is *unplugged* and the battery level decreases.

The attainable actions with the docking station are:

- Charge: Maggie approaches the docking station, plugs into it, and stays there until the battery is full. At the end of this action the robot is still *plugged* and the battery is recharged.
- Remain: it keeps plugged for a while.

*4) Person:* The robot Maggie is intended to interact with people. Hence, people are considered as "objects" of the environment. Regarding interaction, a person has to be close enough to touch, speak or being recognized. For that reason there are two states in relation to a person: *present* and *absent*.

The *person* item offers an available action:

- Interact: this action is related to the possible interaction with a person. During this action, the robot perceives the effects of the people's action over the robot's wellbeing when a user interacts with the robot. These effects are evaluated through oral and tactile interfaces: the user can offend or say compliments to the robot, or he can "stroke" or "hit" the robot.

The system provides identification for different users. Then, different users are treated as different objects of type *person*. Therefore, the robot learns what to do with each user independently.

Some of the presented objects affect the motivations, that is, they are considered as external stimuli. Table I lists all the external stimuli included in this work. Since the robot likes dancing when music is being played, the robot perceives it and the motivation to have *fun* increases. If Maggie perceives the docking station, the motivation of *survival* is augmented. Lastly, due to the fact that Maggie is a very friendly robot and loves people, the presence of a person close to it strengths its *social* motivation.

TABLE I
ALL EXTERNAL STIMULI USED IN THIS WORK

| **Motivation** | **External stimuli** | **State related to ext.stim.** |
|---|---|---|
| fun | music | listening |
| survival | docking station | plugged |
| social | any person | close |

### *C. The consequences of the robot's actions*

Once an action is selected and executed, it may disturb the robot in two manners: first, an action provokes a change in the world (e.g. *charge* action results on the robot is plugged to the charger) and second, the action causes effects over the drives (e.g. after the *charge* action the need of *energy* is reduced). In order to apply the effects over the drives, the action has to successfully end: if an error occurs during its execution, this situation is notified and its effects over the drives are not applied. The changes affecting the external state are monitored by specialized skills.

As highlighted in the previous paragraph, effects of the actions can influence the drives of the robot positively or negatively. A positive effect reduces the value of a robot's drive (this likely implies an increase in the robot's wellbeing). Actually, when the drive is set to zero (the ideal value), it is said that the action satisfies the drive. Some actions can also "damage" some drives of the robot increasing their values (so the robot's wellbeing probably drops).

As shown in table II, when the music player is switched off, the drive *calm* is satisfied; then, a quiet environment is achieved. The need of *fun* is satiated when the robot dances, so the drive *boredom* is set to zero. Since HRI involves a user, the result of this actions is not always the same. Depending on how this user behaves, the action *interact* is positive or negative. A positive interaction is related to a stroke or a compliment and satisfies the *social* drive. In contrast, a negative interaction provokes an increment of ten units in the *social* drive. This happens when the robot is damaged because of a hit or an insult.

TABLE II
EFFECTS OF ACTIONS

| **Action** | **Object** | **Drive** | **Effect** |
|---|---|---|---|
| stop | music player | calm | set to 0 |
| dance | music | boredom | set to 0 |
| positive interaction | person | social | set to 0 |
| negative interaction | person | social | +10 |

The effects of the actions over the drives are not given to the DMS, but they are applied to the drives whose value changes. In addition, the changes in the world caused by the actions, i.e. transitions in the external state, are not predefined. All in all, this means that this is a model-free approach.

## VII. EXPERIMENTAL RESULTS

In this section, several experiments prove the performance of the presented system. First, the use of the Object-Q-Learning algorithm is justified and its benefits are exposed. Following, the advantages of the modifications of the learning algorithm are shown by means of some experiments. Finally, how the robot behaves in all circumstances is analyzed.

During the experiments, the robot has learned the proper behavior in different situations. Learning has been achieved by real robot-environment interaction in the lab (Figure 6). As explained in Section III, each action will be evaluated according to its effect over the robot's wellbeing.

The fact that previous knowledge is not given in advance to the robot implies that all the Q-values have the same initial value. In these experiments this is set to 1.

### *A. Object-Q-Learning vs. Q-Learning*

At this point, the use of the Object Q-Learning is justified. Since the world is perceived in terms of objects and the robot's states in relation to these objects (Section IV-A), an agent using the traditional Q-Learning will learn the actions that satisfy the robot's needs in relation to just one object. However, it does not learn the related actions affecting other objects that are necessary. By means of the Object Q-Learning and the collateral effects, the consequences of an action over all objects in the world are considered.

The different results obtained by Object Q-Learning and Q-Learning can be seen in Figure 7. Both plots present the results obtained after learning the behavior when the dominant motivation is *fun*. That is, what the robot has to

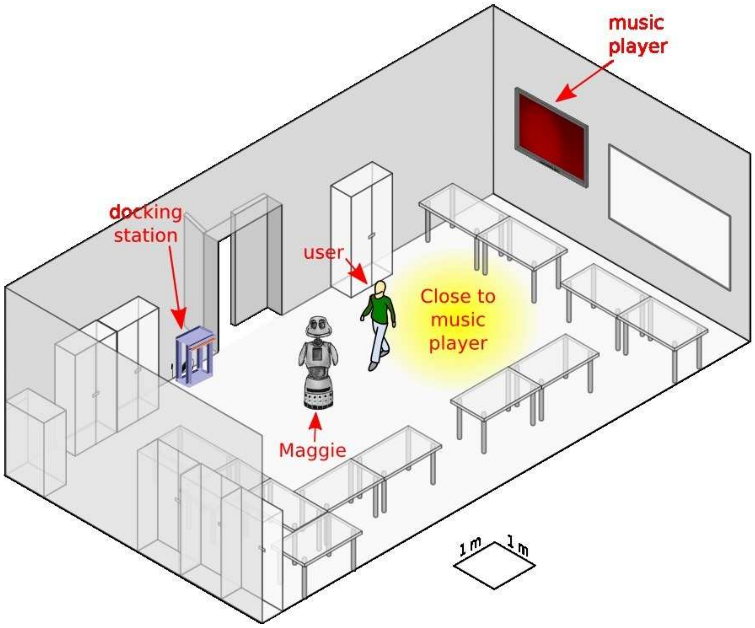
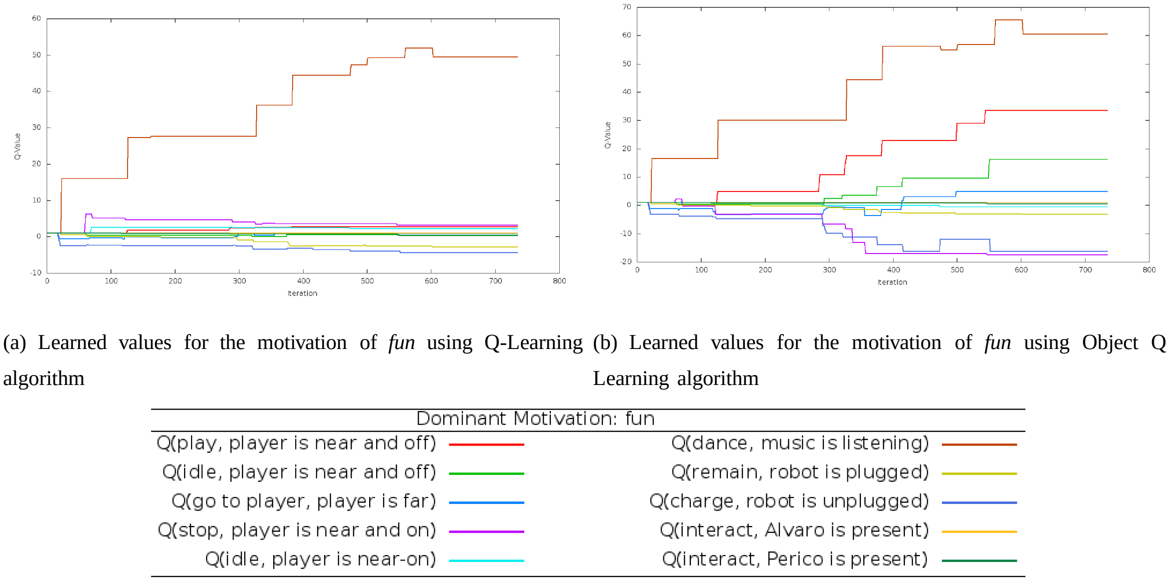


Fig. 6. The laboratory where the experiments have been conducted

do to satisfy the need of entertainment. Figure 7(a) shows the results obtained using Q-Learning. In Figure 7(b), the $Q$ values plotted have been learned by means of the Object Q-Learning algorithm.



(a) Learned values for the motivation of *fun* using Q-Learning algorithm

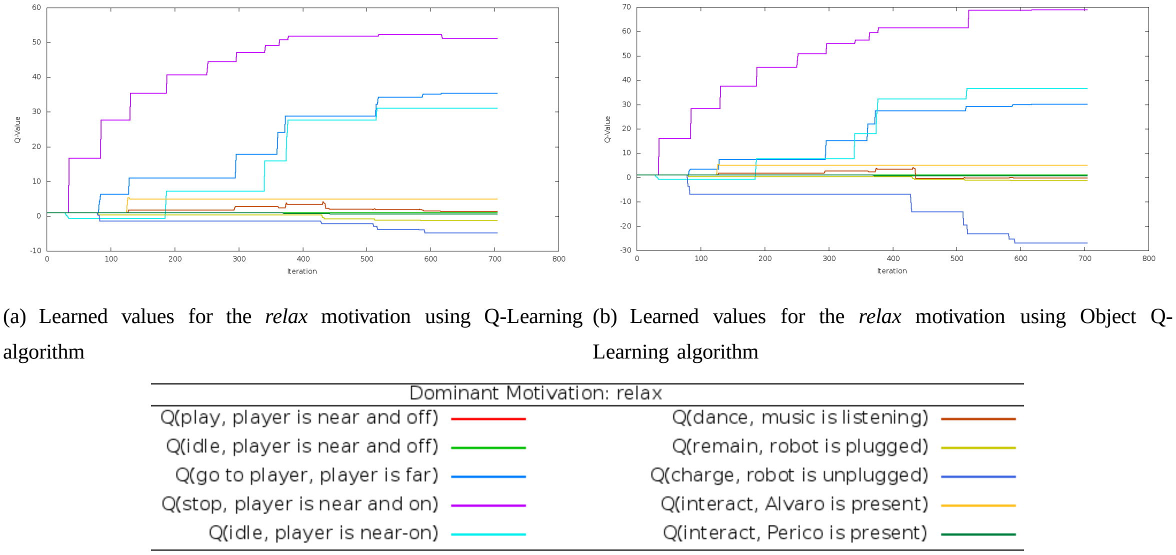
(b) Learned values for the motivation of *fun* using Object Q-Learning algorithm

Fig. 7. Comparison between traditional Q-Learning and Object Q-Learning when several objects are required for performing the behavior related to the motivation of *fun*

As expected, both methods learn that the best action to execute is *dance* because it satisfies the need of *fun*. However, in order to achieve this action, other objects are required: first, if the robot decides to dance, the music has to be on; and for turning the music on, the robot has to be close enough to the *music player*. This relationships among several objects and the states in relation to these objects cannot be learned by Q-Learning (Figure 7(a) shows how the rest of the actions have very low values).

On the other hand, the robot using the Object Q-Learning algorithm perfectly learns the correct relation among actions (even with different objects) in order to expose the proper behavior when *fun* is the dominant motivation. In Figure 7(b) the most appropriate sequence of actions can be extracted considering the highest values. As previously said, *dance* is the most valuable action and it corresponds with the highest value. Before this action can be executed, the *play music* action is required (it is the second highest value due to its collateral effects). Finally, the last required action is *go to player*, which is in charge of moving the robot close enough to the *music player*. Once there, the robot is able to *play music* and, then, to *dance*. The *go to player* action is the forth value and the last positive one.

The state-action pairs with negative Q values are not suitable for the behavior exhibited when *fun* is the dominant motivation. This means that those actions linked to a negative Q value (*stop music* and *charge* actions) drive the robot away from its objective (satiate the need of fun).



(a) Learned values for the *relax* motivation using Q-Learning algorithm

(b) Learned values for the *relax* motivation using Object Q-Learning algorithm

Fig. 8. Comparison between traditional Q-Learning and Object Q-Learning when just one object is involved in the behavior related to the motivation of *relax*

Therefore, it has been proved that Object Q-Learning performs better in relation to the collateral effects. However, when there is just one object involved in a behavior, both algorithms are able to learn the proper skills to be activated. This is the case of the behavior related to the *relax* motivation where just the *music player* is involved. Figure 8 displays the $Q$ values learned when *relax* is the dominant motivation. Figure 8(a) represents the $Q$ values determined by Q-Learning. In contrast, Figure 8(b) represents the results obtained by the Object Q-Learning algorithm. Now, in both cases, the learned values result in the proper behavior, which is formed by actions performed with the same object. The most important actions in order to *relax*, sorted by value, are: *stop music*, *idle with music on*, and *go to player*. All of them are related to the *music player* item and, therefore, both algorithms perfectly identify them.

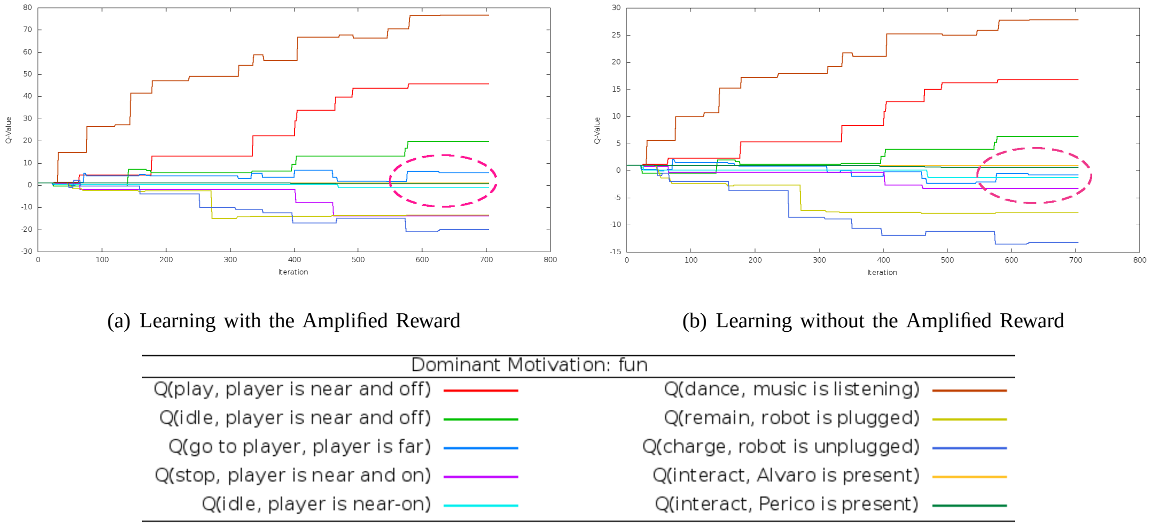
### *B. Validation of the improvements in the learning process*

The benefits obtained by the mechanisms in charge of boosting learning process (Section V) are exposed here. Both, the Amplified Reward and the Well-balanced Exploration, are analyzed comparing the results obtained with and without them in similar experiments.

*1) Amplified Reward:* In order to clearly demonstrate the advantages of using the Amplified Reward, this experiment has been focused in one dominant motivation: the *fun* motivation. In this case, a seven hundred iterations learning session has been performed. Two versions of the learning algorithm are concurrently running: a) an Object Q-Learning algorithm with Amplified Reward (Figure 9(a)), b) an Object Q-Learning without Amplified Reward (Figure 9(b)). The amplification factor has been set to 3 ($f_a$ in Equation 4).

Looking at Figure 9, at first glance, both plots seem similar: despite the fact that the amplified one (Figure 9(a)) has higher values, the policy seems to be equal. However, focusing on the *going to the player* action, the policy learned is not equal. This action is required in order to satisfy the need of entertainment. In Figure 9(a), the $Q$ value associated to this action is the forth highest positive value. In contrast, in Figure 9(b), this $Q$ value is negative and other actions not related to the motivation of *fun* are over its value. Using the Amplified Reward the learned values are higher and, therefore, the back-propagation along all successive needed actions is stronger and it reaches farther actions faster.

Probably, longer experiments will end with a positive value of the *go to the player* action. However, by means of Amplified Reward this is achieved in a shorter period of time.



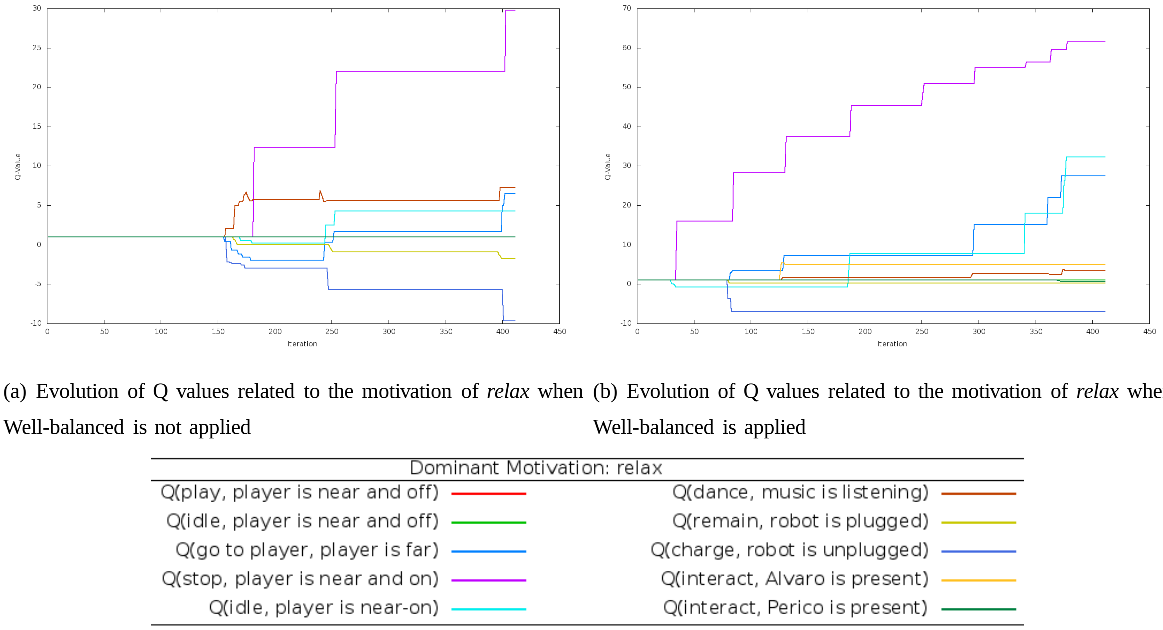
(a) Learning with the Amplified Reward

(b) Learning without the Amplified Reward

Fig. 9. Effects of Amplified Reward on the learning process when the dominant motivation is *fun*

*2) Well-balanced Exploration:* As expressed in Section V-A, an exhausted exploration of all situations in order to correctly learn the proper behaviors is needed. Next, a situation where exploration is poorly achieved is shown. Figure 10(a) presents a four hundred iteration learning session where the Well-balanced Exploration has not been considered. It corresponds to the $Q$ values related to the dominant motivation *relax*, which associated drive (*calm*) is the slowest one.

The remarkable issue extracted from Figure 10(a) is the long periods where non of the values are updated. These are the iterations ranges from $0$ to $160$ and from $250$ to $390$ which correspond to around one hour and a half periods. These long lasting periods with stability of values during a learning session means that this motivation is not explored in these periods. In other words, *relax* does not frequently become the dominant motivation. These circumstances lead to a set of state-action pairs that are not enough explored and therefore their values will not be properly learned in an acceptable amount of time.



(a) Evolution of Q values related to the motivation of *relax* when Well-balanced is not applied

(b) Evolution of Q values related to the motivation of *relax* when Well-balanced is applied

Fig. 10. Application of Well-balance

The effects of the Well-balanced Exploration when *relax* is the dominant motivation can be observed in Figure 10(b). In these experiments, every $15$ iterations the least frequent dominant motivation is promoted (i.e. in Algorithm 1, $f$ is set to $15$). During the whole learning session, there is a frequent update of any state-action pair related to the *relax* motivation. There are not more of those long periods of undesired stability in a particular motivation.

## C. Learned Motivational Behaviors

In this section, the learned behaviors are analyzed. The interactions between the robot and the environment, where experiments are accomplished, take a considerable amount of time. The learning phase has been established around 700 iterations (an iteration corresponds to the execution of an action by the robot). This represents more than seven hours that have been split in a two day experiment. After this learning phase, the robot has acquired the policy of behavior which will be studied.

During the learning, the robot has learned how to act according to its state (internal and external) in order to improve its wellbeing. Through learning, stable chains of actions have been formed and they can be considered motivational behaviors which have not been previously programmed. In this section, the learned behaviors are independently presented motivation by motivation. Moreover, the reaction of the robot when there is not a dominant motivation is also analyzed in the last part.

*1) Survival motivation. How do I get my batteries recharged?:* Figure 11 displays the learned $Q$ values related to all the objects in the robot's world when survival is the dominant motivation. This means that the need of energy is high. The best action, this is the action with the highest $Q$ value, is *charge* which is responsible for the totally recharging of the batteries. Consequently, the energy required is obtained. For that reason, after this action has finished, the *energy* drive is satiated. Then, this action is the most likely to be executed.

| **Q(action, external state)** | **Value** |
|---|---|
| Q(play, player is near and off) | 2.60409 |
| Q(idle, player is near and off) | 0.709885 |
| Q(go to player, player is far) | 41.8524 |
| Q(stop, player is near and on) | 6.64912 |
| Q(idle, player is near-on) | 2.72383 |
| Q(dance, music is listening) | 1.55256 |
| Q(remain, robot is plugged) | -0.909364 |
| **Q(charge, robot is unplugged)** | **71.712** |
| Q(interact, Alvaro is present) | 7.39612 |
| Q(interact, Perico is present) | 3.03879 |

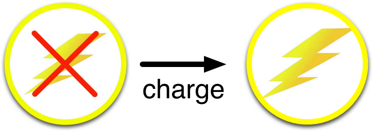


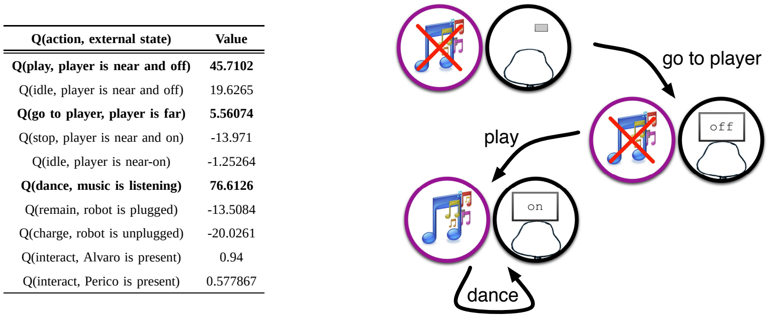
Fig. 11. Learned Q-values and the most probably behavior when **survival** is the dominant motivation

The *go to player* action is very high too because the next best action is the *charge* action. The *charge* action is executed when the robot is unplugged and far from the docking station. This situation results after the execution of the *go to player* action. In addition, *remain* just can be executed once the robot is plugged and this happens after the robot has recharged its batteries (*charge* action). Consequently, as observed in Figure 11, the learned Q-value for this action is not good.

*2) Fun motivation. Let's enjoy!:* This motivation has already been extensively studied in Section VII-A where details can be read. Summarizing, when *fun* is the dominant motivation, the robot approaches the *music player*, it turns it on, and dances. This behavior is extracted from the learned Q values and it is shown in Figure 12.

*3) Relax motivation. I need calm!:* Now, the robot demands a quiet atmosphere, so the dominant motivation is *relax*.

Firstly, it must be emphasized that, if Maggie needs calm is because the music has being playing for some time. In other words, when the music is off, Maggie does not need to relax. Consequently, the $Q$ values related to the actions executed when the *music player* is switched off do not change, so they remain at their initial value of 1.



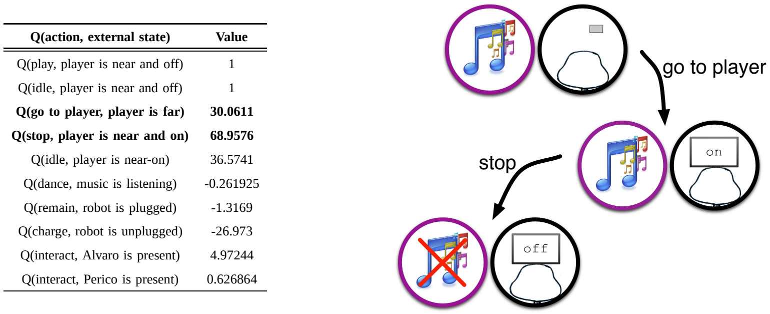
| Q(action, external state) | Value |
|---|---|
| **Q(play, player is near and off)** | **45.7102** |
| Q(idle, player is near and off) | 19.6265 |
| **Q(go to player, player is far)** | **5.56074** |
| Q(stop, player is near and on) | -13.971 |
| Q(idle, player is near-on) | -1.25264 |
| **Q(dance, music is listening)** | **76.6126** |
| Q(remain, robot is plugged) | -13.5084 |
| Q(charge, robot is unplugged) | -20.0261 |
| Q(interact, Alvaro is present) | 0.94 |
| Q(interact, Perico is present) | 0.577867 |

Fig. 12. Learned Q-values and the most probably behavior when **fun** is the dominant motivation

This means that they have not been executed ever when the dominant motivation is *relax* because it is not possible.

After music is playing for a while, the robot *feels* the need of a peaceful environment. Then, it learns that it has to *stop* music. In consequence, this is the highest $Q$ value. As it happens when *fun* is the dominant motivation, the robot must approach the *music player* to operate it. In this case, this is necessary to *stop* music. Accordingly, *go to player* action is the next best action. Once the robot is in the proximity of the *music player* (and the music is on), it can *stop* music or execute *idle* action. Since *stop* is the best action, *idle* value is very high as well. The reason is that when this action ends, the robot can *stop* music which is the highest $Q$ value.

In short, it is easy to describe the optimum behavior that the robot will exhibit when *relax* is the dominant motivation: if it is far from the music player, it will go towards it and then it will stop music (Figure 13).



| Q(action, external state) | Value |
|---|---|
| Q(play, player is near and off) | 1 |
| Q(idle, player is near and off) | 1 |
| **Q(go to player, player is far)** | **30.0611** |
| **Q(stop, player is near and on)** | **68.9576** |
| Q(idle, player is near-on) | 36.5741 |
| Q(dance, music is listening) | -0.261925 |
| Q(remain, robot is plugged) | -1.3169 |
| Q(charge, robot is unplugged) | -26.973 |
| Q(interact, Alvaro is present) | 4.97244 |
| Q(interact, Perico is present) | 0.626864 |

Fig. 13. Learned Q-values and the most probably behavior when **relax** is the dominant motivation

*4) Social motivation. Do you want to be my friend?:* As presented in Section VI-A, the *social* motivation is related to the need of positive HRI. Therefore, when the *social* motivation is the dominant one, the robot is encouraged to interact with the two users: *Alvaro* and *Perico*, who alternatively approach Maggie one by one. *Perico* always interacts with positive actions: he strokes the robot or he says compliments to Maggie. This results on the satisfaction of the social drive, which is set to 0. *Alvaro* generally acts in a positive way too. However, sporadically, he hits or offends Maggie. The consequences of the negative interactions increase some drives (Castro-González, Malfaz, and Salichs 2013).

Interactions with *Alvaro* and *Perico* have a great positive average effect over this motivation. Then, these actions are the most suitable skills to be executed: this is the reason because the highest $Q$ values among all actions, when the dominant motivation is *social*, correspond to *interact-with-Alvaro* and *interact-with-Perico* (see the highest Q values in Figure 14).

| Q(action, external state) | Value |
|---|---|
| Q(play, player is near and off) | 12.3761 |
| **Q(idle, player is near and off)** | **20.5052** |
| Q(go to player, player is far) | -2.95896 |
| Q(stop, player is near and on) | 7.81103 |
| **Q(idle, player is near-on)** | **13.6498** |
| Q(dance, music is listening) | 8.35407 |
| **Q(remain, robot is plugged)** | **25.9918** |
| Q(charge, robot is unplugged) | 9.24678 |
| **Q(interact, Alvaro is present)** | **42.9065** |
| **Q(interact, Perico is present)** | **47.824** |

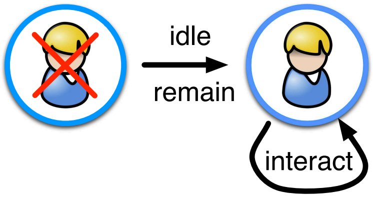


Fig. 14. Learned Q-values and the most probably behavior when **social** is the dominant motivation

Users can approach Maggie at any time. From a social point of view, this exogenous action (approaching Maggie) influences the robot's state and so the availability of endogenous actions; e.g. when a user is with the robot, it can interact with the user. However, it has been observed that users, most of the times, do not approach enough the robot when it is exhibiting a *lively* action like *dancing* or *going to player*. In contrast, they approach Maggie when it is doing other more *lethargic* actions. In particular, these *lethargic* actions are *idle* and *remain*. This is reflected on the $Q$ values of these two actions (Figure 14): the $Q$ values associated to these actions are the next highest actions after the two *interact* actions. This means, that when the robot needs to interact and there is no people around it, it will behave in a passive way by means of *idle* and *remain* actions. It seems like users are reluctant to approximate Maggie as long as it is moving.

*5) There is not a dominant motivation. I'm fine!:* An interesting result can be observed when there is no dominant motivation. This means that there is not any particular need that must be satisfied. Consequently, this situation corresponds to a *pleasant* state. But, how does Maggie behave in this case? What does it do when there is not specific needs? The results are shown in Table III.

TABLE III

LEARNED Q-VALUES WHEN THERE IS NOT A DOMINANT MOTIVATION

| Q(action, external state) | Value |
|---|---|
| **Q(play, player is near and off)** | **8.97197** |
| Q(idle, player is near and off) | 1.03933 |
| **Q(go to player, player is far)** | **6.74179** |
| Q(stop, player is near and on) | 1.60373 |
| Q(idle, player is near-on) | -2.39112 |
| **Q(dance, music is listening)** | **11.31** |
| Q(remain, robot is plugged) | -4.46195 |
| **Q(charge, robot is unplugged)** | **12.9346** |
| Q(interact, Alvaro is present) | 1.62473 |
| Q(interact, Perico is present) | 1.8989 |

The values for all actions related to the satisfaction of the need of fun are relatively high. This is because *boredom* is usually one of the highest drives due to its fast increase. Then, every time these actions are executed the robot will likely receive positive reward. However, the most valuable action is the *charge* action. This produces a pattern of behavior where either the robot charges its battery or it turns the music player on and dances, even if it is plugged. This can be interpreted as the robot satisfies two basic needs even if they are not urgent. It is like if the robot foresees the most likely future needs and it gets ready in advance. These needs do not depend on other external elements and can be satisfied by the robot itself.

The rest of the actions are either slightly positive or negative (they are all around zero), but there are not really low or high values. This means that none of these actions play a crucial role in the absence of dominant motivation.

## VIII. Discussion and conclusions

The goal of our autonomous robot is to learn what to do in every situation in order to survive and to maintain its needs satisfied. The presented work proposes a method which endows the robot with the capability to learn the proper behaviors autonomously, without any supervision, just by robot-environment interaction. The robot has learned the correct behaviors to deal with each motivation in different situations. That is, Maggie has learned when to execute the actions that lead to satiate the most urgent need. By means of the Well-balanced Exploration and the Amplified Reward mechanisms, the learning time has been significantly reduced. In addition, the robot using Q-Learning learns the direct action to deal with each motivation and the preceding actions, all of them linked to the same object. However, this is not enough to behave in complex environments where objects may be related. Object Q-Learning provides a mechanism to acquire the required knowledge in order to exhibit behaviors that satisfy motivations involving several independent objects and their states. Then, the proper action with each object at each particular state will be carried out.

Since social robots move and interact with humans sharing the same areas, one of the main requirements for social robotics is a natural behavior. That is, behaviors perfectly understandable and accepted by people, like those exhibited by animals. Consequently, from a HRI point of view, the behaviors displayed by a social robot, like Maggie, should be considered as animal-like. This will help to improve the interaction when robot is *living* with people. People would feel comfortable when they easily understand what the robot is doing and why. In contrast, people could reject a *machine* that is doing *weird* things that they do not understand. This can be observed on domestic animals: humans feel comfortable having pets at home, among other reasons, because it is easy to assess if your cat wants to be stroked, or it is hungry; when the owner does not understand what the cat is doing, he/she is worried and unpleasant. Therefore, it is important that robot's behaviors are comprehended by its *world-mates*. The experiments and all parameters have been set considering this situation. Therefore, when the robot is autonomously deciding its own behaviors based on the learned policy, the observer is able to understand what the robot is doing. Besides the robot provides a really life-like appearance which benefits the assessment of the robot and consequently the HRI, making the person to feel more *comfortable*.

When the robot exploits the learned policy, complex behaviors are shown by series of simpler actions. For example, when the robot is motivated to have fun, it approaches the music player, turns it on, and then dances. In contrast, when the dominant motivation is relax, the robot approaches the music player and switch it off. In relation to the social motivation, if the robot is alone, it decides to remain where it is until a person approaches and then they interact. Other behaviors look more elemental because just one single action is involved: when the battery are depleted the robot needs to survive so it gets its energy refilled by plugging to the docking station and remaining there. However, the mechanism under the hood is the same independently of the complexity of the consequent behaviors.

Behaviors are elicited due to the combination of the dominant motivation and the situation in the robot's world. In a situation where there is not a dominant motivation, this means that there is not an urgent need so the robot is at a pleasant state. Learning has also been carried out in these cases, so the robot has also learned how to behave when it is *comfortable*. In general, most of the resultant $Q$ values in this situation heavily fluctuate, so there is not a clear behavior. However, two state-action pairs are quite stable and have relative high stable $Q$ values associated, what gives the idea that both actions will be likely selected. These state-action pairs are: the *play* action when it is close to the *player* and the music is off, and the *dance* action when the *music* is being listened. This implies that when dominant motivation does not exist, the robot will likely turn the music player on and dance. Why is so? Both actions are related to the behavior exhibited when *fun* is the dominant motivation. Since this motivation is one of the fastest one and due to the fact that it does not depend on external agents, these actions almost always get a positive reward. Moreover, these two actions are relative short on time (specially the *play* action which takes around few seconds), and then the increment on drives is minimum. Therefore, the potential decrement in the robot's wellbeing is minimum. From other perspective, as just said, *fun* is one of the fastest motivation and, during learning, it was frequently the dominant motivation, i.e. the robot frequently needs to have fun. This reaction (dance when the dominant motivation does not exist) can be understood as a mechanism preventing from the most probable future need of entertainment.

Observing the robot's behavior when it follows the learned policy and there is not a dominant motivation (this is most of the time) gives the impression of a "dance-aholic" robot. Recalling the experiments carried on by Olds and Milner in 1950s (Olds and Milner 1954), rats rapidly became addictive to electrical self-stimulation into certain areas of their brains. This leaded to the discovery of the called pleasure centers. The behavior exhibited by the robot seems similar to how these rats acted: it is like the "*robot's pleasure center*" is being stimulated while dancing, so Maggie becomes addicted to dancing. This is an animal-like behavior that has emerged.

## ACKNOWLEDGMENT

The authors gratefully acknowledge the funds provided by the Spanish Government through the project call "Aplicaciones de los robots sociales", DPI2011-26980 from the Spanish Ministry of Economy and Competitiveness.

## REFERENCES

Barber, R. (2000). *Desarrollo de una Arquitectura para Robots Moviles Autonomos. Aplicacion a un Sistema de Navegacion Topologica.* Ph. D. thesis, Universidad Carlos III de Madrid.

Barber, R. and M. A. Salichs (2001). Mobile Robot Navigation Based on Event Maps. In *International Conference on Field and Service Robotics*, pp. 61–66.

Barber, R. and M. A. Salichs (2002). A new human based architecture for intelligent autonomous robots. In *Proceedings of The 4th IFAC Symposium on Intelligent Autonomous Vehicles*, pp. 85–90. Elsevier.

Barto, A. G. (2013). Intrinsic motivation and reinforcement learning. In G. Baldassarre and M. Mirolli (Eds.), *Intrinsically Motivated Learning in Natural and Artificial Systems*, pp. 17–47. Berlin, Heidelberg: Springer Berlin Heidelberg.

Barto, A. G., S. Singh, and N. Chentanez (2004). Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd international conference on development and learning (ICDL 2004), Salk Institute, San Diego*. Citeseer.

Blumberg, B. M., P. M. Todd, and P. Maes (1996). No Bad Dogs: Ethological Lessons for Learning in Hamsterdam. In *Collection*, Volume 01463, pp. 295–304. MIT Press.

Bolado-Gomez, R. and K. Gurney (2013). A biologically plausible embodied model of action discovery. *Frontiers in Neurorobotics 7*.

Boutilier, C., R. Dearden, and M. Goldszmidt (2000). Stochastic dynamic programming with factored representations. *Artificial Intelligence 121*(1-2), 49–107.

Boyan, J. A. and A. W. Moore (1995). Generalization in Reinforcement Learning: Safely Approximating the Value Function. *Advances in Neural Information Processing Systems 7 7*, 369–376.

Bryson, J. J. and E. Tanguy (2009). Simplifying the design of human-like behaviour: Emotions as durative dynamic state for action selection. *International Journal of Synthetic Emotions*, 355–377.

Castro-González, A., M. Malfaz, and M. A. Salichs (2011). Learning the Selection of Actions for an Autonomous Social Robot by Reinforcement Learning Based on Motivations. *International Journal of Social Robotics 3*(4), 427–441.

Castro-González, A., M. Malfaz, and M. A. Salichs (2013). An Autonomous Social Robot in Fear. *IEEE Transactions on Autonomous Mental Development*, 1–1.

Gadanho, S. C. (1999). *Reinforcement Learning in Autonomous Robots: An Empirical Investigation of the Role of Emotions*. Ph. D. thesis, University of Edinburgh.

Gadanho, S. C. and J. Hallam (1998). Exploring the Role of Emotions Autonomous Robot Learning. In *Proceedings of the AAAI Fall Symposium on Emotional Intelligence*, Number Wilson 91, pp. 84—-89. AAAI Press.

Gadanho, S. C. and J. Hallam (2001). Emotion-triggered learning in autonomous robot control. *Cybernetics & Systems 32*, 531—-559.

Gatsoulis, Y., C. Burbridge, and T. M. McGinnity (2012). Biologically inspired intrinsically motivated learning for service robots based on novelty detection and habituation. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pp. 464–469. IEEE.

Givan, R., T. Dean, and M. Greig (2003). Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence 147*(1-2), 163–223.

Guestrin, C., D. Koller, R. Parr, and S. Venkataraman (2003). Efficient Solution Algorithms for Factored MDPs. *Journal of Artificial Intelligence Research 19*(c), 399–468.

Gurney, K., N. Lepora, A. Shah, A. Koene, and P. Redgrave (2013). Action discovery and intrinsic motivation: a biologically constrained formalisation. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pp. 151–181. Springer.

Kaplan, F. and P. Oudeyer (2007). Intrinsically motivated machines. In *50 years of artificial intelligence*, pp. 303–314. Springer-Verlag.

Kowalczuk, Z. and M. Czubenko (2011, December). Intelligent decision-making system for autonomous robots. *International Journal of Applied Mathematics and Computer Science 21*(4), 671–684.

Kubota, N., Y. Nojima, N. Baba, F. Kojima, and T. Fukuda (2000). Evolving pet robot with emotional model.

Li, L., T. J. Walsh, and M. L. Littman (2006). Towards a Unified Theory of State Abstraction for MDPs. *Work*, 531–539.

Lopes, M., T. Lang, M. Toussaint, and P.-Y. Oudeyer (2012). Exploration in model-based reinforcement learning by empirically estimating learning progress. In *NIPS*, pp. 206–214.

Lorentz, K. (1987). *Behind the Mirror*, Volume 12. Methuen, London.

Maes, P. and R. A. Brooks (1990). Learning to Coordinate Behaviors. In *Advanced Robotics*, Volume 10 of *AAAI'90*, pp. 796–802. AAAI Press/The MIT Press.

Mahadevan, S. and J. Connell (1992). Automatic programming of behavior-based robots using reinforcement learning. *Artificial intelligence*.

Malfaz, M. (2007). *Decision Making System for Autonomous Social Agents Based on Emotions and Self-learning.* Ph. D. thesis, Carlos III University of Madrid.

Malfaz, M. and M. A. Salichs (2009). Learning to deal with objects. In *Proccedings of the 8th International Conference on Development and Learning (ICDL 2009)*.

Malfaz, M. and M. A. Salichs (2010). Using muds as an experimental platform for testing a decision making system for self-motivated autonomous agents. *Intelligence and Simulation of Behaviour Journal (AISBJ) 2*(1).

Malfaz, M. and M. A. Salichs (2011, November). Learning To Avoid Risky Actions. *Cybernetics and Systems 42*(8), 636–658.

Martinson, E., A. Stoytchev, and R. C. Arkin (2001). Robot behavioral selection using q-learning.

Matarić, M. J. (2007, September). *The Robotics Primer*. The MIT Press.

McCallum, A. K. (1996). *Reinforcement learning with selective perception and hidden state*. Ph. D. thesis, University of Rochester.

Olds, J. and P. Milner (1954). Positive reinforcement produced by electrical stimulation of septal area and other regions of rat brain. *Journal of Comparative and Physiological Psychology; Journal of Comparative and Physiological Psychology 47*(6), 419.

Oudeyer, P.-Y., A. Baranes, and F. Kaplan (2013, February). Intrinsically Motivated Learning of Real World

Sensorimotor Skills with Developmental Constraints. *Intrinsically Motivated Learning in Natural and Artificial Systems*.

Rivas, R., A. Corrales, R. Barber, and M. A. Salichs (2007). Robot Skill Abstraction for AD Architecture. In *6th IFAC Symposium on Intelligent Autonomous Vehicles*.

Salichs, J., A. Castro-González, and M. A. Salichs (2009). Infrared Remote Control with a Social Robot. In Springer (Ed.), *FIRA RoboWorld Congress 2009*, Incheon, Korea. Springer.

Salichs, M. A., R. Barber, A. Khamis, M. Malfaz, J. Gorostiza, R. Pacheco, R. Rivas, A. Corrales, E. Delgado, and D. Garcia (2006). Maggie: A Robotic Platform for Human-Robot Social Interaction. *2006 IEEE Conference on Robotics Automation and Mechatronics*, 1–7.

Singh, S., A. G. Barto, and N. Chentanez (2005, June). Intrinsically Motivated Reinforcement Learning. *Advances in neural information processing systems 17*, 1281–1288.

Sprague, N. and D. Ballard (2003). Multiple-Goal Reinforcement Learning with Modular Sarsa ( 0 ). *Processing 18*(0), 0–2.

Starzyk, J. A. (2010). Motivated Learning for Computational Intelligence. *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives*.

Thrun, S. B. (1992). Efficient Exploration In Reinforcement Learning. Technical Report January, Carnegie-Mellon University.

Vigorito, C. M. and A. G. Barto (2010). Intrinsically Motivated Hierarchical Skill Learning in Structured Environments.