IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 20, 2021, accepted June 22, 2021, date of publication July 7, 2021, date of current version July 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095392

Speeding-Up Action Learning in a Social Robot With Dyna-Q+: A Bioinspired Probabilistic Model Approach

MARCOS MAROTO-GÓMEZ, RODRIGO GONZÁLEZ, ÁLVARO CASTRO-GONZÁLEZ, MARÍA MALFAZ, AND MIGUEL ÁNGEL SALICHS, (Senior Member, IEEE)
Department of Systems Engineering and Automation, University Carlos III of Madrid, 28903 Madrid, Spain

Corresponding author: Marcos Maroto-Gómez (marmarot@ing.uc3m.es)

This work was supported in part by the Robots Sociales para Estimulación Física, Cognitiva y Afectiva de Mayores (ROSES), in part by the Innovación y Universidades and RoboCity2030-DIH-CM through the Ministerio de Ciencia, in part by the Madrid Robotics Digital Innovation Hub under Grant S2018/NMT-4331, in part by the Programas de Actividades I+D en la Comunidad de Madrid, and in part by the Structural Funds of the European Union (EU).

ABSTRACT Robotic systems that are developed for social and dynamic environments require adaptive mechanisms to successfully operate. Consequently, learning from rewards has provided meaningful results in applications involving human-robot interaction. In those cases where the robot's state space and the number of actions is extensive, dimensionality becomes intractable and this drastically slows down the learning process. This effect is specially notorious in one-step temporal difference methods because just one update is performed per robot-environment interaction. In this paper, we prove how the action-based learning of a social robot can be improved by combining classical temporal difference reinforcement learning methods, such as Q-learning or Q(λ), with a probabilistic model of the environment. This architecture, which we have called Dyna, allows the robot to simultaneously act and plan using the experience obtained during real human-robot interactions. Principally, Dyna improves classical algorithms in terms of convergence speed and stability, which strengthens the learning process. Hence, in this work we have embedded a Dyna architecture in our social robot, Mini, to endow it with the ability to autonomously maintain an optimal internal state while living in a dynamic environment.

INDEX TERMS Action learning, decision-making, human–robot interaction, probabilistic model, reinforcement learning, social robots.

I. INTRODUCTION

Adaptive learning in changing environments is crucial for living beings to survive. Accordingly, if robots are to be deployed in dynamic environments to assist humans in real-time complex tasks, then endowing them with learning capabilities is essential [1]. Thrun and Mitchell [2] claims that if a robot lacks initial knowledge about itself and its environment, then learning becomes inevitable because otherwise the robot would be a simple unresponsive automaton. In this context, multiple works describe how social robots learn to behave naturally by learning from demonstration, as surveyed in [3]. Many studies support the use of social robots to help older adults in many different tasks, which suggests that the opinion of the aging population will become very important for robot acceptance and usability [4]–[6]. Consequently, many authors have concentrated their efforts in designing robots with natural interaction capabilities [7], [8] and biologically inspired motivated affective behaviours [9]–[11].

This work is a continuation of our previous studies about endowing our social robot, Mini, with action-based learning capabilities according to its needs and the stimuli that it perceives from the environment [12]. Mini is a social robot that is principally devoted to assisting caregivers in cognitive stimulation therapies with older adults, and is also able to offer different entertainment and educative applications. Thus, when the robot is not focused on developing an specific task, it autonomously performs complementary behaviours (e.g. sleeping, dancing, or talking), which allows it to behave in a more natural fashion. Consequently, autonomous

The associate editor coordinating the review of this manuscript and approving it for publication was Tomasz Trzcinski.

decision-making architectures have been designed to provide social robots with action-based reinforcement learning capabilities according to its internal needs and the perceived resources in the environment; initially by Malfaz *et al.* [13], Malfaz and Salichs malfaz2011biologically, and later in Castro-González [14], [15].

In a recent contribution, we demonstrated how Mini learns, using Q-learning [16], to autonomously behave according to its motivational urges and the resources that it perceives in the environment. In these previous works, the goal of the robot is to behave following a policy that allows it to maintain the best possible internal welfare. Despite the positive results obtained in our previous studies regarding the policy of behaviour learnt by the robot, classical reinforcement learning methods, such as Q-learning, require extensive training periods as the number of state-action combinations increase. Consequently, to overcome this issue, this work aims to present a new reinforcement learning approach that is based on a Dyna architecture, which allows the robot to speed-up the action learning to a large extent. Dyna [17] can be defined as a probabilistic model, which improves the convergence speed and stability of classical algorithms, such as one-step Q-learning [18] or multi-step Q-(λ) [19]. This is based on the idea of combining real experiences, which are gathered from classical algorithms during real human-robot interactions, with simulated trials supported by previous real experiences lived by the robot. The probabilistic model that we propose in this work replicates the rewards and state transitions that were previously experienced by the robot in simulation domains. This allows it to plan how it should behave, while at the same time it gains real experience from interacting with the real world. The possibility to improve the model of the environment from real interactions is the main contribution of this work because the learning system is able to learn faster, while reducing any possible errors generated by the model.

To assess how Dyna-Q+ (i.e. a Dyna architecture presenting online adaption mechanisms) improves the performance of Q-learning and Q(λ), we compare their efficiency in a real human-robot interaction scenario where the robot learns which action produces the best effects on its internal state in each situation where the robot is involved. Reaffirming our previous hypothesis, Dyna-Q+ produces better results in terms of learning speed and stability, but at the cost of requiring more computational power per time step. In addition, Dyna-Q+ yields an online adaptive behaviour to dynamic environments because it promotes exploratory behaviours, which is something that classical algorithms do not contemplate unless training is repeated (leading to a loss of the robot's previous knowledge).

The rest of this paper is structured as follows. In Section II, we survey the current state of the reinforcement learning techniques that are used for robotic applications. Section III theoretically describes the reinforcement learning algorithms that will be used in this work, Q-learning, Q(λ) and Dyna-Q+. It will describe the modifications that have been made to improve their performance on our specific application. A detailed enumeration of their advantages and disadvantages is also provided. Section IV describes how the social robot, Mini, motivationally learns to behave according to its needs and the state of external cues perceived from the environment. In Section V, we describe the experimental set-up and how the performance of the learning system has been evaluated, which demonstrates how the combination of a probabilistic model with classical techniques improves the convergence speed and stability. Section VI contains the results obtained from the comparison of classical reinforcement learning techniques and Dyna-Q+ (model version). In the next section VII, we discuss the outcomes produced by each of the algorithms. Finally, we conclude in Section VIII, including a description of some new ideas to enhance robot biological modelling and learning by reinforced actions.

II. RELATED WORK

Assistive social robots in elderly care are now capable of performing many different tasks, such as companionship [20], assisting in mild cognitive impairment therapies [12], [21], [22], entertainment [23], [24], or education [25]. Consequently, endowing robots with proper human-robot interaction (HRI) mechanisms is essential to accomplish these tasks. A natural interaction between humans and robots will improve the user experience, which will allow the robot to correctly execute its actions. Consequently, robot behaviour plays an important role in the attainment of these tasks.

In most cases the behaviour of a robot is normally predefined. However, in the last few years robots have started to autonomously learn how to behave through social interaction using machine learning techniques (e.g. [26], [27]). Reinforcement learning has been proven to allow robots to learn how to behave in a changing world, while interacting with their environment. Remarkable examples in this field can be found in the navigation systems of both social and mobile robots, where behaviour adaptation is essential in not disturbing the user's intentions while moving around. Furthermore, the navigation system must consider more information than just path planning between two points, such as the personal distance with the users, collision avoidance, and being receptive for interaction; as Takayama and Pantofaru [28] posit. Deep reinforcement learning has recently been applied in a social robot's navigation system. For example, in [29] the navigation system that is embedded in the robot uses the user's feedback reward and the human's prior knowledge to autonomously wander in the environment. Chen *et al.* presented [30] a fully autonomous learning process, which they called 'Socially Aware Collision Avoidance with deep Reinforcement learning' (SA-CADRL). SA-CADRL allows a robot to learn how to autonomously navigate while respecting social norms in a dynamic environment with the presence of pedestrians. Collision avoidance with deep reinforcement learning has evolved in recent works, such as [31], [32]. In particular, its performance in environments where big

groups of people are involved and the robot does not have any knowledge of people's dynamics and behaviours has been improved.

Reinforcement learning techniques are not just used in social navigation applications. Robot behaviour adaptation can be successfully obtained in real-time in many different ways, such as using speech recognition or estimating user engagement by body posture and face orientation [33]. For example, the speech velocity and vocal content of a robot has been adapted in post-stroke rehabilitation therapies, depending on the personality of the patient [34]. Similar studies presented in [27] show how the robot's social behaviour has been adapted, depending on the task it is performing. In a recent contribution, Park *et al.* [35] demonstrate how their robot selects optimal stories according to the child's educational level. In this educational line, assistive robots with adaptation capabilities have been studied for use with children with autism disorders [36].

Depending on the field of sociable robots, learning by rewarded actions has gained huge attention in the last few years. For example, Matarić studied how groups of mobile social robots learn to yield and share information in a foraging task [37]. In addition, Qureshi *et al.* [38] study how a robot learns social skills from trial and error. They found that the robot's behaviour adaptation, while learning from social interaction, lets the agent perceive the effects of its actions on the environment and on other peers. By using action rewards, the robot is able to modify its behaviour if it detects that the human peer is not engaged, or maintain its behaviour if feedback is positive. Following similar studies, Ritschel [39] shaped the social behaviour of a robot by dynamically adapting its extroversion using reinforcement learning techniques. By gathering audio and visual information from the user, the robot was also able to adapt its affective expressiveness. In a similar approach, Weber *et al.* used social information to adapt the humour of a robot using reinforcement learning and social rewards [40]. In conclusion, these previous studies are representative examples of how social robots can successfully learn after interacting with their environment by reinforced signals.

Continuing with the reinforcement learning paradigm, many algorithms have been proven to converge to an optimal solution in different robotic applications (Q-Learning [18] and Q(λ) [19] are two outstanding examples). In social robotics, Q-learning has been successfully applied to learning the optimal policy of behaviour in motivated agents [14], [16], or in user personalized robots during social interaction [35]. Departing from the remarkable results provided by Q-learning, Q(λ) arose as a temporal difference method that notably improves the convergence speed using the well-known multi-step update, which is called eligibility traces. Q(λ) has been used in the context of mobile robots, which allows the robot to learn how to navigate in the environment, in works such as [41]. In the field of healthcare and therapy, users have socially interacted with the Zeno robot [42]. Other applications can be found in the well-known robot soccer competition [43]. However, Q-learning and Q(λ) can often have slow training processes because each update performed on the algorithm requires long interactions and many states may not be visited regularly. To overcome this drawback, a learning architecture based on modelling the environment, called Dyna, has developed been [44]. The Dyna architecture is grounded in the idea of combining a classical reinforcement learning temporal-difference method, such as Q-learning or Q(λ), with a model of the environment. The use of an environmental model results in an increase of the learning speed. This has been demonstrated in robotic navigation systems, where Dyna outperforms previous algorithms, especially speeding up learning and providing online adaptation by promoting exploration [45]. Dyna has also been used with deep reinforcement learning algorithms, such as the one described in [46], where a task-completion dialogue agent is trained in a more efficient way using less real user interactions. More recently, Hayamizu *et al.* [47] have proven how a Guided Dyna-Q architecture (GDQ) allows a mobile robot to successfully navigate while reducing the exploration speed of the environment. Similarly, Lee and Jeong [48] highlight the benefits provided by a Dyna-Q architecture in comparison with classical Q-learning during path planning tasks for mobile robots. In their results, Lee at al. demonstrate how paths generated by Dyna-Q are much shorter than those learnt by Q-learning, at the cost of an increase in time consumption. In mobile robotics, Deep RL has been used to optimize autonomous exploration tasks (e.g. [49]), providing remarkable results. In relation to social robotics, recent works have used Dyna-Q architecture and Deep RL to improve the learning speed of classical algorithms in tasks related to getting a pedestrian's attention without causing them discomfort [50], in learning proper approach behaviour [51], and in autonomous speech volume control for noisy environments [52].

In addition to its broad application in robotics, Dyna-Q and Deep RL has also been applied to different fields, notably improving the learning speed and convergence stability of a system. For example, in underwater scenarios, both acoustic communications [53] and acoustic sensor networks [54] have been enhanced using Deep RL algorithms. In line with the previous work, Deep RL in combination with Dyna-Q has also been applied to support green computing in Internet of Things (IoT) scenarios. Min *et al.* [55] propose a Dyna model that allows IoT devices to reduce computational latency and energy consumption. In electrical vehicles, Wang *et al.* [56] proposed an autonomous scheduling system for charging the vehicles' battery. Finally, in an interesting study presented in [57], both Deep RL and model-based Dyna architectures are combined to enhance the learning process. Using a new framework, these authors developed an intelligent trainer that is able to autonomously deal with common algorithm problems, such as hyperparameter tuning.

Considering the previous works presented in this section, in this contribution we propose an action-based learning system for a social robot, Mini, that relies on a probabilistic

Dyna architecture. This architecture allows classical temporal difference algorithms, such as Q-learning or Q(λ), to improve both convergence speed and stability. This enhancement is produced because the learning process occurs simultaneously during real and simulated human-robot interactions. Thus, this architecture is embedded in our robot to learn how to adapt its behaviour to maintain its physiological and social state in the best possible conditions. Even though many of the previous works use Deep RL to approximate the learning function, this approach is mostly useful when the space-action state is too large to be handled by tabular methods. In our environment, the representation of our state-action space can be afforded by tabular methods, so we decided not to use Deep RL because we believe that it will not improve the results but will instead increase the complexity of the learning system. This work extends our previous research in the field [13]–[16], not only by speeding-up the learning process but also by endowing it with a greater convergence stability (in spite of presenting a stochastic reward distribution).

III. LEARNING BY REWARDED ACTIONS

This section formalizes the reinforcement learning problem, which will form the basis of the principal algorithms that will be used in this work and how they have been combined with a probabilistic model to speed-up the learning process. Reinforcement learning methods have commonly been grouped into dynamic programming, Monte Carlo, and Temporal-Difference (TD); as Sutton and Barto propose in [44]. Whereas the Monte Carlo and TD methods are model-free, and dynamic programming is model-based, the former are used for episodic tasks. Consequently, in this work we decided to use TD methods because they can be used in continuous applications and do not require an explicit model of the environment.

A. FORMALIZATION

Before delving more deeply in the TD algorithms, it is important to formalize the problem that we are facing in this work. A reinforcement learning problem can be formulated as a Markov Decision Process (MDP), which fulfils the Markov property [58]. This property states that the values of future variables, such as the optimality of executing action a in state s, only depend on present values, if and only if Equation 1 equals Equation 2.

$$Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0\} \tag{1}$$

$$Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t\} \tag{2}$$

MDPs are represented by a tuple of the type $\{S, A, T, R, \gamma\}$, where S denotes a set of states, A denotes a set of actions, $T(s, a, s') = P(s`|s, a)$ denotes the transition probability function, R denotes the reward function, and $\gamma \in (0, 1]$ denotes the discount factor. Considering this, in a reinforcement learning problem, the agent, who is in a certain state at a particular time (s_t), after performing an action (a_t), selected following a greedy policy, ends in a new state (s_{t+1}). In response to this transition, the agent obtains a reward r from the reward function R, which represents the suitability of executing action a_t in state s_t. Having this formulation in mind, the goal of the agent is to find the policy $(\pi: A \times S \rightarrow [0, 1], \pi(a, s) = Pr(a_t = a | s = s_t)$ that maximizes the reward obtained during its lifespan, which is normally defined as cumulative reward.

B. ONE-STEP Q-LEARNING

Watkins proposed Q-learning [18] as an off-policy TD algorithm, which evaluates how beneficial an action turned out to be in a certain state and assigns an optimality value (denoted as Q-value) to each state-action pair of the robot, as in the following Equation 3.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\delta_t \tag{3}$$

where:

$$\delta_t \leftarrow r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a) - Q(s_t, a_t) \tag{4}$$

The Q-value assigned to the pair (s, a) is only updated when the robot is in state s and performs action a. The new Q-value depends on the discount factor (γ), which regulates the importance given to the last reward obtained, and on the learning rate (α), which defines the importance of the new information gathered by the agent opposed to the previous Q-value assigned to the state-action pair. According to [44], the computational complexity of Q-learning is $O(d)$ because Q-learning is in nature an off-policy TD(0) method.

C. MULTI-STEP Q(λ)

TD(λ) methods appeared as an evolution of early TDs, being supported by the concept of eligibility traces [19]. Eligibility traces influence the Q-value update by considering how recently an action has been executed. Then, as the agent performs new actions, traces of the different (s, a) pairs decay considering the last time that these actions were executed. TD(λ) algorithms allow us to take into account actions that occurred in the past. Thus, the update rule provides that not only the Q-value assigned to the actual (s, a) pair is updated but all (s, a) pairs with an eligibility trace different from 0 are updated also per iteration. Consequently, the learning speed of the agent is notably increased at the cost of requiring more computational time per step, as learning back spreads to previous executed actions. Thus, the computational complexity of Q(λ) is $x \cdot O(d)$, where x represents the robot's state space number. Equation 5 shows the update rule considered in Q(λ), making use of the eligibility trace $e(s_t, a_t)$ assigned to each state-action pair in Equation 6. Every time that an action a is selected in state s, its corresponding trace $e(s_t, a_t)$ is increased by 1 unit. Meanwhile, traces of remaining state-action pairs decay according to parameters discount factor γ and decay rate λ, which controls the decay speed of traces.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\delta_t e(s_t, a_t) \tag{5}$$

where:

$$e(s_t, a_t) \leftarrow \begin{cases} e_{t-1}(s,a)+1 & \text{if } a \text{ is executed in } s \\ \gamma\lambda e_{t-1}(s,a) & \text{if } Q(s_t,a_t) = \max_{a'} Q(s_{t+1},a) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

D. DYNA-Q+

Temporal difference methods learn from real interaction with the environment, which makes them especially slow, as the number of the (s, a) pairs increases. This issue becomes more remarkable in robotics, mainly because real HRI actions last too long and some states may not be visited regularly. In contrast, model-based methods can overcome this problem by simulating these interactions by replicating a model of the environment.

The Dyna-Q+ architecture [17] offers the simplicity of model-free methods with the faster convergence speed of model-based ones by using a TD algorithm, such as Q-Learning or Q(λ), in combination with a model of the environment which plans in the background. In the Dyna-Q+ architecture, the model of how the agent interacts with the real world is used to speed-up its action learning. In the architecture, the agent makes use of the information obtained from interacting with the environment in two different ways. First, experience is used in model improvement, being an important process since the model should accurately represent the real world dynamics, which are not static. Second, it handles the learning process itself, combining both real and simulated experiences. Once the agent has gathered enough real experience, it plans its optimal behaviour using the model while gaining more real experience. As the agent continues interacting with the environment, planned information generated by the model is used, in combination with real experiences to speed-up learning, because the agent is more aware of the effects produced by its actions. Considering the operation of Dyna-Q+, its main advantages are bounded to the planning process because it can be repeated more than once per real interaction. In addition, states that are rarely visited in real interactions can be elicited more often in simulation domains, which produces positive effects in the exploration of the environment. However, it is clear that the computational complexity of Dyna-Q+ is increased with respect to Q-learning and Q(λ). Thereby, according to [59], the computational complexity of Dyna-Q+ can be expressed as $n \cdot x \cdot O(d)$, where n represents the number of planning steps and x the number of possible states of the agent. Hence, Dyna-Q+ requires $n \cdot x$ times more computational power that Q-learning and n times more than Q(λ).

E. OUR APPROACH

The learning system that we propose in this work combines Q(λ) with a model based on a probabilistic transition and reward prediction. The model gets feedback from real interactions that occur between the agent and the environment,

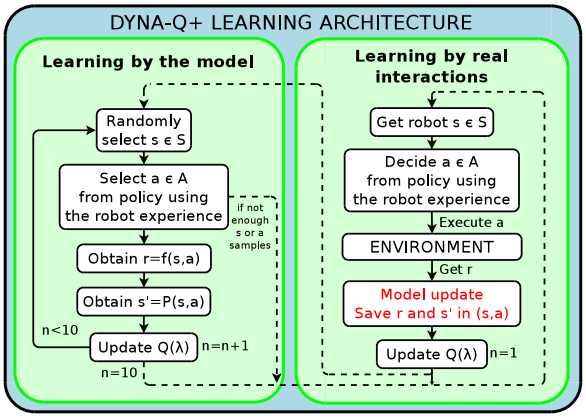


FIGURE 1. Learning process using the Dyna-Q+ architecture: Planning in simulation (left-hand) and acting in the real world (right-hand).

and stores this information aiming to simulate (plan) the effects produced by each action. The experiences lived by the agent are saved in its memory so that it can preview (while acting) the optimal behaviour that it has to execute in real domains while maintain a good internal state. As Figure 1 depicts, every time that a real action ends and updates its Q-value using Q(λ)'s update rule, the reward r and the next state s' derived from executing action a in state s are saved. The model then randomly selects a (s, a) pair. In case the agent's memory contains at least 5 values of real rewards and 5 previous state transition from s to potential s' after executing action a, the model starts working in the background while the agent continues interacting with the real world. Both the number of real rewards and real transitions between states necessary to start the simulation domain have been set empirically. Otherwise, the planning process will not run until enough samples are obtained from real interactions. Whether the Dyna-Q+ model has gained enough information for (s, a) pair, it generates a simulated reward r and probabilistically decides the following virtual s' to update the Q(λ) algorithm.

The simulated reward r is obtained by randomly selecting a value after defining a normal distribution from the previous obtained rewards in (s, a) pair. Following a similar approach, s' is selected by assigning a probability to each of the previous states to which the agent transits from state s after executing action a. The probability assigned to each s' depends on the number of times that the agent has transited from s to s', divided by the total number of transitions starting in s. The planning process is repeated n times per real acting, unless not enough experience had been acquired for any of the (s, a) pairs involved in the agent's planning. It is important to remark that the action selection process, both in real and simulated interactions, is performed using the well-known Boltzmann equation [60], which is a method that is widely used in reinforcement learning domains, and which controls the agent's environmental exploration and exploitation of learnt actions. This equation allows us to regulate the degree of

exploration and exploitation by means of the Temperature (T) parameter. High T values foster action exploration, while low values promote exploitation of the robot knowledge. Section IV-D describes how the learning system online regulates the Temperature parameter by decreasing it as the number of actions performed by the robot in each action-state pair increases. Finally, the model is updated because it exists a real interaction between the agent and the environment. As a result of this interaction, a reward and a transition between states is obtained. These data are then added to the previously stored samples.

IV. AUTONOMOUS DECISION-MAKING IN THE SOCIAL ROBOT MINI

The social robot Mini [12], as represented in Figure 2, is an autonomous platform that was developed to assist older adults with mild cognitive impairment. Mini was devised to daily coexist with people at home, aiding them in common tasks while providing them companionship and entertainment. To provide these services, the robot is able to execute a wide range of diverse activities that allow it to interact with its environment in many different ways. Its software architecture is supported on a decision-making system that controls the behaviour execution using the actuation system of the robot depending on the inputs received from the perception system of the robot.

FIGURE 2. Mini, a social robot used to assist older adults with mild cognitive impairment in cognitive stimulation therapies.

A. PERCEPTION AND ACTUATION SYSTEMS

Mini can perceive different stimuli from the environment, due to its wide range of sensors. It contains four touch sensors placed on its head, one on each shoulder, and another in the belly, a 3D stereo camera, and a microphone. The information given by these sensors is received by the perception manager, which is a module that translates raw data into understandable information that is processed by the rest of modules of the robot.

Considering its actuation system, Mini is able to move its upper body using five servomotors that are located at its hip, one on each arm, one on the neck, and another on head. Additionally, two animated screens simulate two expressive eyes and a RGB led emulates the heartbeat. It contains a stereo speaker to play nonverbal sounds and generate speech, and a tablet device to display multiple multimedia content. Actuators are controlled by the expression manager, which is a module of the robotic architecture that receives commands by the decision making system and handles them to produce suitable expressions.

In this experiment, Mini principally uses its 3D camera to perceive whether the user is present in the scene or not, so that the interaction can be accomplished. It uses a microphone and a speaker to verbally communicate with the user, and a tablet device to play a quiz game. In addition, it can perceive if a virtually created music player is able to play music.

B. ROBOT STATE

In this work, the state of the robot is defined as a combination of its internal (S_{inner}) and external state (S_{ext}), being mathematically expressed as $S_{robot} = S_{inner} \times S_{\text{ext}}$. On the one hand, the internal state is ruled by its dominant motivation m_{dom}, which is the motivation with the highest level of intensity among all active motivations ($S_{inner} = m_{dom}$). On the other hand, the external state is represented as the state of the robot in relation to the agents and objects of the environment that can interact with it. In this work, the robot's external state is defined by the state of a user ($S_{\text{ext user}}$) and by the state of a music player ($S_{\text{ext music player}}$), being mathematically expressed as $S_{ext} = S_{\text{ext user}} \times S_{\text{ext music player}}$. The following sections will give a definition of both the internal and external state of the robot. It is worth noting here that, as will be detailed later on, the robot's internal state can be defined by four different dominant motivations, the state of the user can be present or absent, and the music player is on or off. Thus, the robot's state space dimension is $4 \times 2 \times 2 = 16$, which is denoted in this manuscript as $S_{robot} = \left(m_{dom}, S_{\text{ext user}}, S_{\text{ext music player}}\right)$.

1) ROBOT'S INTERNAL STATE

The behaviour exhibited by the robot arises from the temporal evolution of 4 biological processes modelled as homeostatically controlled variables, which represent the robot's *Tiredness*, *Boredom*, *Social interaction*, and *Knowledge*. The evolution of each biological process ranges from 0 to 100, presenting an ideal value on one of both extremes; as shown in Table 1. As times goes on, each biological variable starts deviating from its ideal value by a variation rate (applied every time step of 0.2 seconds) starting from an initial value, as represented in Table 1). The deviation of a biological variable from its ideal value produces a deficit d_i on the biological process with effect on the agent, which drives the robot's internal system to reduce it. The deficit derived from each biological variable influences a motivational state, driving motivated behaviour. In this experiment, the internal state S_{inner} is defined by the dominant motivation m_{dom}; that is, the motivation with the highest value. Thus, the internal state of the robot can be *Rest, Play, Socialise, and Learn*. The higher the needs of its related biological process, the higher the intensity of its

TABLE 1. Definition of the motivational states of the robot Mini, its related biological process, its temporal evolution parameters and the stimuli with influence on the motivational state. Each motivation is linked to a specific biological variable, which is defined by a set of features. The initial value represents the starting point of the biological variable evolution. The lower and upper limits delimit the biological variable's evolution range. The ideal value represents the optimal state of the biological variable. Finally, the variation rate represents how fast the deficit related to each biological grows in absence of regulatory behaviours.

Motivation	Biological Variable	Initial value	Lower limit	Upper limit	Ideal value	Variation rate	Stimuli
Rest	Tiredness	0	0	100	0	0.1	None
Play	Boredom	0	0	100	0	0.3	User present Music player on
Socialise	Social	100	0	100	100	-0.2	User present
Learn	Knowledge	100	0	100	100	-0.3	None

corresponding motivational state. Moreover, the motivational state's intensities m_i can be increased if the robot perceives certain stimuli in the environment, as Equation 7 represents. Note that the biological processes and motivations of the robot pretend to emulate animal biology. Thus, the internal state does not have any physical correspondence with any processes occurring in the robot because they are fictional.

$$m_i = d_i + d_i \frac{1}{N} \sum_{k=0}^{N} s_k \tag{7}$$

Perceived stimuli are associated with an intensity value s_k ranging [0, 100]. The level of intensity s_k increases 2 units per time step if the stimulus is perceived, and is reduced in 1 unit if the stimulus is not perceived by the sensors of the robot. Thus, in Equation 7, m_i represents the intensity of each motivation, d_i is the deficit of the biological variable associated with the motivation, s_k is the intensity associated to the stimuli with influence over the motivation, and N is the number of stimuli with effect on the motivation. Equation 7 presents the novelty of considering more than one potential stimuli with effect on the motivation intensity, which our previous studies [14]–[16] did not consider.

In this work, the stimuli that the robot can perceive and which can affect its motivational states are the presence of the user, that increases its motivation to Play and Socialise, and, if the music player is turned on, music increases the motivation to play. Table 1 shows the parameters that denote the temporal evolution of each biological variable in terms of its initial value, lower limit, upper limit, ideal value and variation rate. It also contains the relationship of each biological process to each motivational state, and the related stimulus which boost motivational intensities.

In the last step, the decision-making system of the robot selects, following a winner-take-all approach, the motivation with the highest intensity among all, denoted as dominant motivation m_{dom}, only if at least one motivational state presents an intensity m_i level above 20 units. Thus, the internal state of the robot can be expressed as $S_{inner} = max(Rest, Play, Socialise, Learn)$. Note that the idea is that the learning system implemented in Mini allows it to know which action is the best for each motivational state and the current external situation (perception of stimuli). So once the policy of behaviour has been learnt, the robot will autonomously reduce its deficits maintaining an optimal well-being.

Mini has been endowed with the behaviours of sleep, wait, dance, play a quiz game, talk alone, talk with a user, search for information, and turn on the music player. Each one of these actions produces different effects on the needs derived from the biological processes of the robot. Therefore, what is required is that our learning system acknowledges these effects and links each action with a specific robot state. Note that not all behaviours can be executed under the same situation because some of them require the perception of a certain stimulus. For example, the robot should learn that it cannot dance if the music player is off, or cannot play the game if a person is not perceived next to it. To overcome these situations, Mini has a couple of behaviours, which are normally defined as appetitive, that do not produce any direct effect over the biological variables of the agent but drives it to perceive/obtain specific environmental resources that enable the execution of a new behaviour, which is normally denoted as consummatory [61]. For example, if Mini wants to dance to reduce its boredom, then it will have to learn that first it has to turn on the music player (an appetitive behaviour). It is important to remark that the robot does not know a priori which behaviours are appetitive and which ones are consummatory, having to acknowledge this distinction during the learning process. Thus, our approach seeks to maximize the improvement in the biological state of the robot after executing an action. Consequently, the reward function (defined in Equation 9, which is explained in detail in Section IV-D), that is used in this experiment accounts for the variation of the deficits of the biological processes, giving more importance to the deficit related to the dominant motivation d_{mot}. If the action is not correctly executed, then it will fail and provide a bad reward to the robot. Later, in Section IV-C, the list of behaviours of the robot and the effects they produce over the biological variables are presented.

Using the previous biologically inspired set-up, the internal state of the robot, denoted as S_{inner}, allows it to behave by fostering the interaction with people without leaving aside the rest of its artificially created needs. This means that the

deficits of the biological variables bound to the motivations related to the interaction of the user (boredom and social interaction) increase faster than the deficits of the rest of the biological variables. Thus, the values set in Table 1 have been set by empirically denoting the evolution of the biological internal processes of the robot foster the interaction with the user. In addition, these values have been empirically set by considering the application of our robot and our previous experience in this type of systems.

2) ROBOT'S EXTERNAL STATE

The perception of certain stimuli affects the state of the robot and therefore influences which behaviour it will execute. As previously stated, Mini's external state, denoted as S_{ext}, is defined by the state of the agents and the objects that the robot interacts with. In this work, the robot's external state is influenced the user's state $S_{\text{ext user}}$ and the state of the music player $S_{\text{ext music player}}$. Using its perception system, the robot is able to perceive if the user is in front of it (present or absent, $S_{\text{ext user}} = \{present, absent\}$) and if the music player is turned on or not ($S_{\text{ext music player}} = \{on, off\}$). Thus, the external state of the robot can be defined as the state of these objects with respect to the robot, which is modelled as user present/absent and music player turned on/off. Note that in some situations the perception of the user and the music player influences the action selection of the robot because certain behaviours can only be activated if the user is present and the music player is turned on. Considering the music player, Mini is able to change its state by turning it on or off, but it cannot control the users' behaviour, and therefore their state. In contrast from the internal state of the robot, the stimuli that affect its external state have a correspondence with physical agents and objects in real the world.

C. ROBOT's BEHAVIOURS

The decision-making system [16] of Mini controls the execution of behaviours in each situation, as Figure 3 shows. As was presented in in the previous sections, the goal of this framework is to endow Mini with a learning system that allows it to know how to execute optimal actions according to its state towards maintaining a good welfare state. In this sense, the robot's state can be represented as a combination of its internal state (i.e. needs of the robot derived from its biological processes) and its external state, as a consequence of the state of environmental stimuli considered by the robot. In this learning scenario, Mini can deploy a set of behaviours, which are listed below and represented in Figure 4, that allow it to reduce its needs depending on the availability of certain resources. These actions present two possible outcomes: success if the action is completely satisfactory executed, and failed otherwise (e.g. if a necessary resource is not available). The outcome of each action determines the reward obtained by the robot once the action has finished, as will be detailed later in Section IV-D. It is important to remark that the set of behaviours that are listed below represent the action space

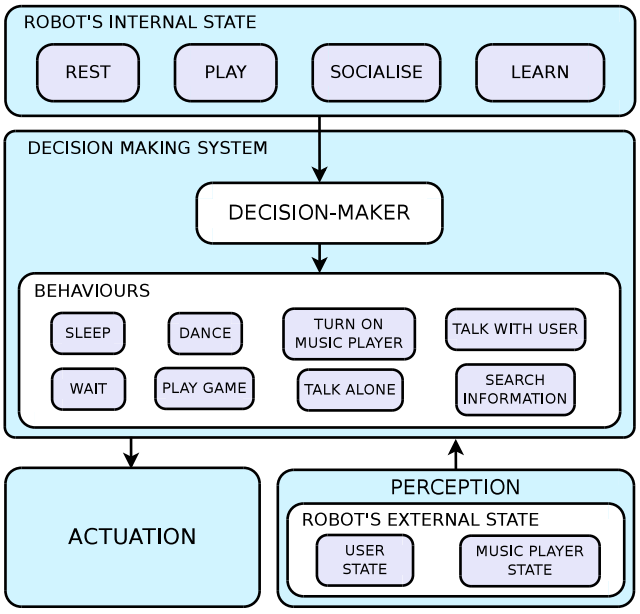


FIGURE 3. Decision-making architecture supported on the perception system and the internal motivations of the robot for controlling its behaviour execution.

considered in the RL system that learns how the robot can reduce its deficits.

Sleep: The robot simulates that it is sleeping by closing its eyes, and performing yawns and similar gestures.

Wait: The robot relaxes for a while without doing any explicit activity.

Dance: The robot plays a song on the music player and starts dancing following the rhythm of the music. This action fails if the music player has not been turned off previously.

Play game: The robot and the users play a quiz game together in which the user has to guess the answer to three different questions after selecting a category to play from sports, art and entertainment, history, science and technology, and geography. The game fails if the user is not perceived by the user.

Talk alone: The robot talks alone by saying out loud some phrases from a wide repertoire.

Talk with user: The robot asks the user general questions to maintain a short conversation with them.

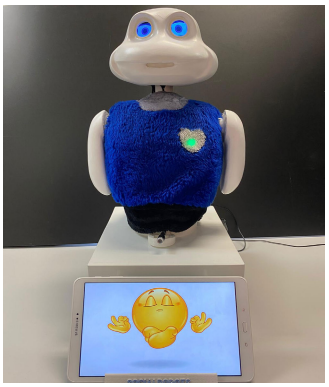
Search information: The robot autonomously surfs the Internet looking for the last news and reads the news stories out loud.

Turn on the music player: The robot turns on the virtual music player to play songs. Once on, the music player turns off by itself after two minutes.

D. LEARNING SYSTEM

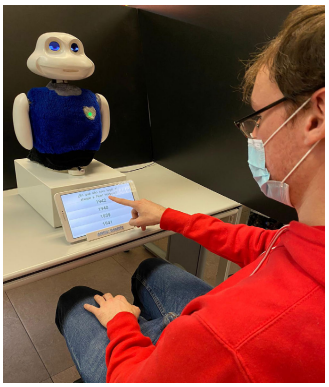
The action-based learning system that is presented in this paper is included in the decision-making module of the robot, which endows it with the ability to autonomously learn the most suitable action for each robot state. In this experiment, three algorithms are simultaneously tested while running at

(a) Mini sleeping.



(b) Mini waiting.

(c) Mini dancing.

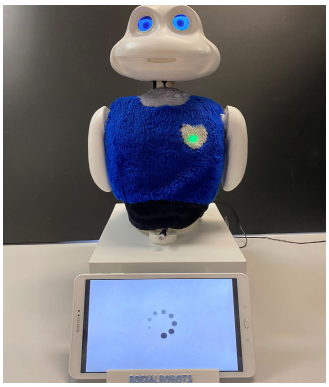


(d) Mini playing a quiz game with the experimenter.

(e) Mini talking alone.

(f) Mini talking with the experimenter.

(g) Mini searching for information.



(h) Mini turning on the music player.

FIGURE 4. **Graphical view of the social robot Mini executing its available behaviours to reduce its needs to maintain a good biological state.**

the same time in the robot, Q-learning, Q(λ), and Dyna-Q+ (which is supported on a probabilistic model and Q(λ)). Thus, during the interaction, Mini selected its actions using Boltzmann's probability distribution, which assigns a selection likelihood to each of the actions of the robot depending on its Q-value and the Temperature parameter. In this experiment, the robot initially randomly decides its actions exploring the environment (i.e. setting high Temperature values) but, as Mini gains experience from these interactions, it exploits its optimal behaviour by selecting those actions that reduce its salient needs the most (dynamically reducing the Temperature). Note that from the comparison of the three algorithms, classical Q-learning and Q(λ) are supposedly slower than Dyna-Q+ because it does not contain the benefits provided by the planned experiences. Additionally, Dyna-Q+ presents the advantage of promoting subtle exploratory periods after the learning has been completed by perceiving new changes in the environment. Nonetheless, the stopping criteria in our learning setting was produced once one algorithm converged to the optimal solution because one of the main goals of this work is to demonstrate the faster learning procedure of Dyna-Q+ architecture with respect to classical RL algorithms.

In this work, the discount factor γ of the three algorithms has been empirically set to a constant value of 0.8 units, the learning rate α was initially set to 1, decaying inversely proportional to the number of times n each action a_t has been performed in state s_t, $\alpha_{s,a} = \frac{1}{n}$. For those algorithms supported on eligibility traces, the decay rate λ was empirically set to 0.8 units. Following this configuration, the convergence of all Q-values is guaranteed by the law of large numbers [62]. Additionally, the Temperature T of the Boltzmann equation was initially set to a high value $t_0 = 100$, promoting action exploration. As the experiment moves forward, the Temperature decays following the expression defined in Equation 8, where t_A represents the sum of the number of times the agent has performed an action in a particular state s and $dr = -0.002$ defines the decay rate, which was calculated to value the low limit of 0.1 when $t_A = 1000$. Thus, as t_A increases, T decreases. This drives the agent to exploit its learnt policy by maintaining a good state of well-being.

$$T = t_0 \cdot e^{dr \cdot t_A} \tag{8}$$

Finally, it is important to clarify that once the robot has enough real experience, the Dyna-Q+ architecture performs $n = 10$ planning steps per real acting. This value has been

TABLE 2. List of behaviours of the social robot Mini and the value of their effects over each biological internal variable.

Behaviour	Effects
Sleep	Tiredness: -0.5 Boredom: -0.1 Social: +0.1 Knowledge: +0.1
Wait	Tiredness: -0.25 Boredom: None Social: None Knowledge: None
Dance	Tiredness: +0.3 Boredom: -0-5 Social: None Knowledge: None
Play game	Tiredness: +0.2 Boredom: -0.5 Social: +0.2 Knowledge: None
Talk alone	Tiredness: +0.2 Boredom: None Social: +0.5 Knowledge: None
Talk with user	Tiredness: +0.2 Boredom: -0.2 Social: +0.5 Knowledge: None
Search information	Tiredness: +0.2 Boredom: None Social: None Knowledge: +0.5
Turn on the music player	Tiredness: None Boredom: None Social: None Knowledge: None

empirically set to balance the learning speed and computational resources. As pointed out both in [44] and [59], the learning speed increases with the number of planning steps but the computational time is also negatively affected. Moreover, note that high n values may lead to the model not representing the real world because it does not gather enough real experience.

In every reinforcement learning setting, rewards are obtained from feedback received after the agent executes its actions making use of a reward function. This function indicates to the agent how adequately it is performing each action in a certain state. Previously, we stated that the goal of this work is to endow our social robot with action-based learning capabilities to maximize the improvement on the internal state. Consequently, the reward obtained by the robot after executing an action in a certain state depends on the variation of the deficits d_i tied to each biological variable. Thus, if the deficits of the robot are reduced after executing the action, then the internal state of the robot has improved leading to a positive reward. Additionally, in this setting, if the deficit related to the dominant motivation m_{dom} is satiated, then the reward obtained by the robot is even higher because it has reduced its most urgent need. Equation 9 represents the reward function proposed in this contribution and used by the action-based learning system, which defines the reward received by the robot after executing a particular action. Action rewards are calculated as the variation of the deficits of the robot per unit of time (in seconds) during the execution of the action weighting the deficit related to the dominant motivation d_{mot} by 0.5 and the rest of the deficits d_i by 0.5, where M is the total number of biological processes representing the internal state of the robot and t is the duration of the action in seconds.

$$r = \begin{cases} \frac{1}{t}\left(0.5 \cdot \Delta d_{mot} + 0.5 \cdot \sum_{i=0}^{M-1} \Delta d_i\right) & \text{if } a \text{ succeeded} \\ -1 & \text{if } a \text{ failed} \end{cases} \tag{9}$$

V. EVALUATION

This section defines the experimental set-up in which our social robot Mini demonstrates how it learns the optimal behaviour by focusing on maintaining the deficits related to its biological variables satisfied.

A. EXPERIMENTAL SET-UP

In this learning scenario, Mini was placed in a room of a house where the two participants, aged 27 and 24, lived during the COVID-19 lockdown in September 2020. None of the participants' personal information was stored. The participants interacted with the robot without following a predefined pattern, appearing at the scene of interaction at will. During the experiment, which lasted for five consecutive days, the robot was placed on a desktop in a bedroom of the house, exhibiting an autonomous behaviour. Initially, the robot did not have any kind of information about what behaviour was the most suitable to execute according to its state. Consequently, at the beginning of the experiment, the robot randomly selected the behaviour to execute. Nonetheless, as the experiment moved forward, the robot started to explore the effects yielded by each action on its deficits, and under which situations each action can be executed depending on the availability of resources (e.g. the state of the user and the state of the music player). If the resources needed to execute a particular behaviour were not accessible, then the robot had to learn the correct sequence of actions to reach them. Verbal expression used by the robot when talking alone or when playing the

game were set to motivate the user to approach the robot and start the interaction. Taking this into account, if any experimenter was close to the robot when it needed to play or talk, then Mini could perform behaviours that did not require the presence of the user. It is important to remark that when the participants were interacting with the robot, they acted at will. Consequently, the answers provided while playing the game or having a conversation with the robot were not predefined and depended on the participant's own intentions.

B. METRICS

The results obtained in this work are presented by focusing on two main streams. First, we compare the learning results of Q-learning, Q(λ), and Dyna-Q+ in terms of convergence speed and stability, by contrasting the evolution of each Q-value signal (for each algorithm) during the learning process. Second, we evaluate the learnt policy in terms of the robot's well-being. Therefore, according to this evaluation system, we formally define the metrics used in this experiment as follows.

Convergence speed and stability are described by the temporal evolution of the Q-values obtained for each of the algorithms we have appraised. The convergence speed is represented by the number of steps that each algorithm needs to converge to an optimal solution, while the convergence stability is represented by the Mean Squared Error (MSE) (assuming that the optimal Q-value is the one where the algorithm converges). Thus, in the comparison, we consider that the algorithm that needs fewer real steps to converge is the faster one and the algorithm with the minimum MSE is the more stable.

Optimal policy is the behaviour exhibited by the robot which maximizes the sum of rewards returning during the lifespan of the agent, which is represented by the maximum Q-value in each robot state.

Wellbeing is a representation of the situation of the needs of the robot. In related literature, this term has been previously denoted as well-being [15] or comfort [63]. Mathematically, it is calculated following Equation 10, where a value of 100 means that the internal well-being of the robot is at its ideal value (i.e. all of the robot's needs are satiated), while a value of 0 means that all of the robot's needs are at their peak, for a number M of biological needs d_i. This metric represents the goal of this contribution because once the learning process has finished, the robot's well-being should have improved noticeably with respect to the initial stages of the learning process.

$$Wellbeing = 100 - \frac{1}{M}\sum_{i=0}^{M} d_i \tag{10}$$

VI. RESULTS

This section contains the results obtained by comparing the performance of Q-learning, Q(λ), and Dyna-Q+ Reinforcement Learning algorithms in terms of their convergence speed and stability. It will also present the optimal policy learnt by the winner algorithm and the benefits yielded on the robot's internal state once the learning process has been completed.

A. CONVERGENCE SPEED AND STABILITY

The experiment previously described in Section V was performed to compare the performance of the three algorithms. Due to impossibility of graphically representing the 160 Q-values derived from the state-action combinations, we opted to simply show the graphs corresponding to the optimal actions (the action that produce a bigger reduction of the robot's needs) for each robot state (see Figure 6 for the optimal policy Q-values representation and visit our repository[1] to check the graphs of all Q-values obtained in the experiment). It is important to remark that the x axis of each Q-value graph represents the number of times that each action has been executed for each robot state. Note that in case of Dyna-Q+, these real interactions also contain simulated experiences. Considering this idea, if the Q-value signal converges in less action executions, then the robot is learning faster.

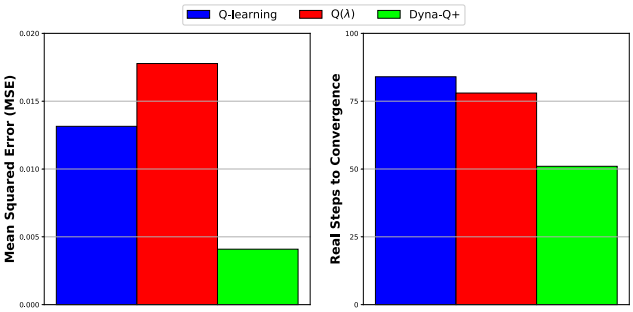


FIGURE 5. Averaged Mean Squared Error (MSE) (left-hand) and average number of real steps to convergence (right-hand) for each of the algorithms.

Looking at Figure 6 and with strong support from Figure 5, it can be concluded that the curves corresponding to the Dyna-Q+ architecture (green) converge in fewer interactions than Q(λ) (red) and Q-learning (blue) for all state-action pairs. In addition, Dyna-Q+ model provides a more stable convergence curve, whereas, in some of the graphs, especially Q-learning but also Q(λ) curves are more dependent on the stability of the rewards received. This effect is more notable in two events: first, for those robot states where the presence of the user and the state of the music player suppose a big influence, the perception system can sometimes provide wrong measures, leading to an incorrect reward; and second, when there are not enough updates. For example, Figure 6m represent a notable instability for both Q-learning and Q(λ) algorithms as a consequence of a low number of real updates (note that Dyna-Q+ Q-value evolution combines real and simulated updates). A second example of convergence instability is represented in Figure 6l, where some

[1] Link to repository.

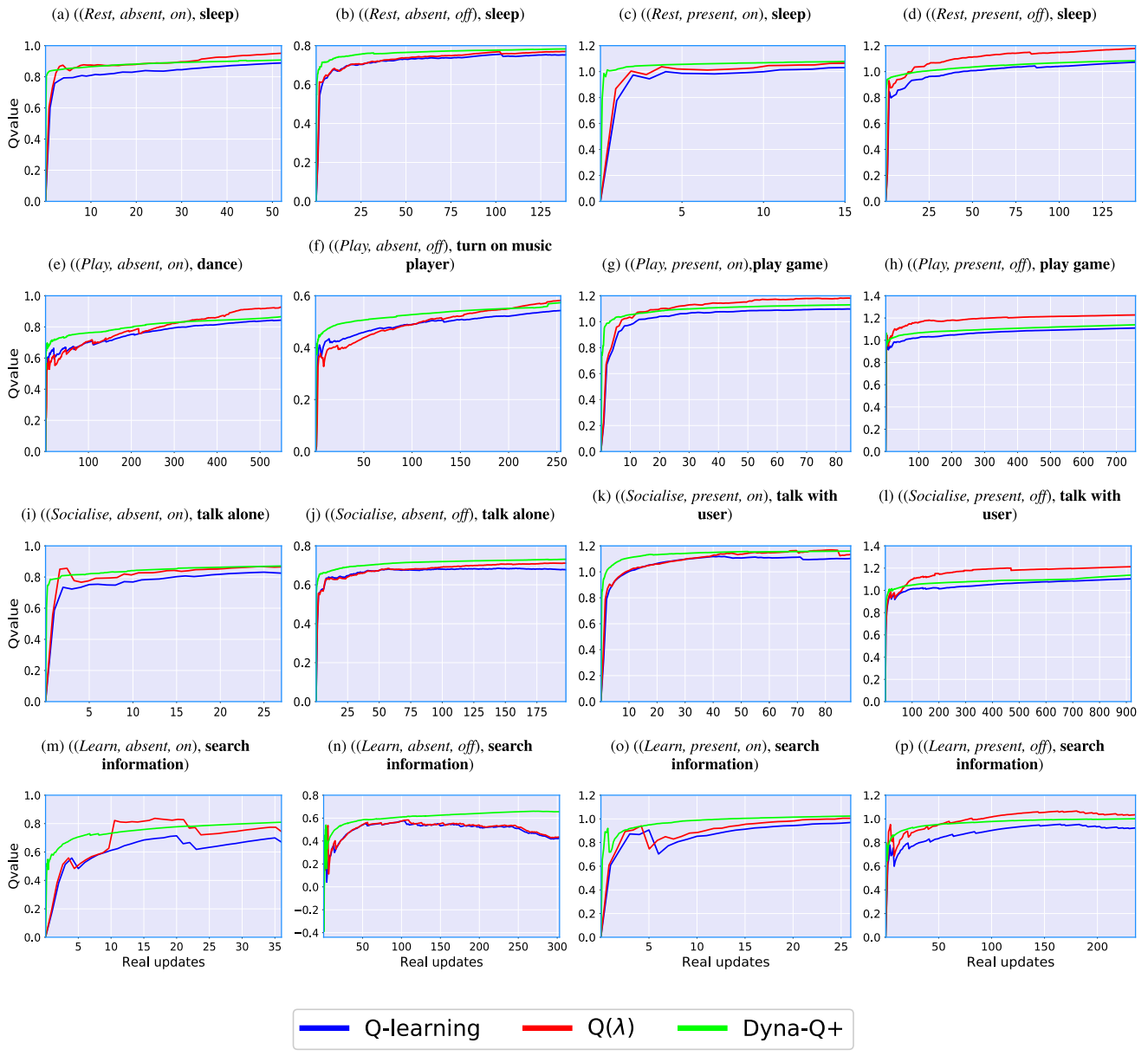


FIGURE 6. Temporal evolution for the best Q-value obtained for each robot state considering the reinforcement learning algorithms Q-learning, Q(λ) and Dyna-Q+. Note that the key defining each graph follows the format (s, a) (e.g. ((*Learn, present, off*), search information)).

incorrect perceptions of the user presence make the signals corresponding to Q-learning and Q(λ) algorithms to dither at the beginning of its evolution. Hence, as we hypothesized, the simulated experiences provided by Dyna-Q+ allow the robot to learn on average in 46 real steps, by the 75 of Q-learning and 68 of Q(λ) (Figure 5) (right-hand). Considering the MSE, Dyna-Q+ presents a value that is clearly below (0.01), while Q-learning is clearly above 0.015 and Q(λ) considerably above 0.02 (Figure 5) (left-hand). In addition, notable improvements are provided by Dyna-Q+ with respect to classical Q-learning and Q(λ) in terms of learning speed and stability. However, we observed from the outcomes obtained in this comparison that the optimal policy learnt by both classical algorithms is not correct because in some situations (in some robot states) the robot is unable to reduce its salient needs correctly.

B. LEARNT POLICY

It is worth noting here that none of the graphs in Figure 6 pretend to compare the final Q-values learnt by each of the algorithms. Instead, they provide a graphical proof that the convergence of Dyna-Q+ is faster and more stable than the convergence of both Q-learning and Q(λ). Recall that, as was explained in the previous section, the principal goal of each algorithm is to learn the optimal policy of behaviour that

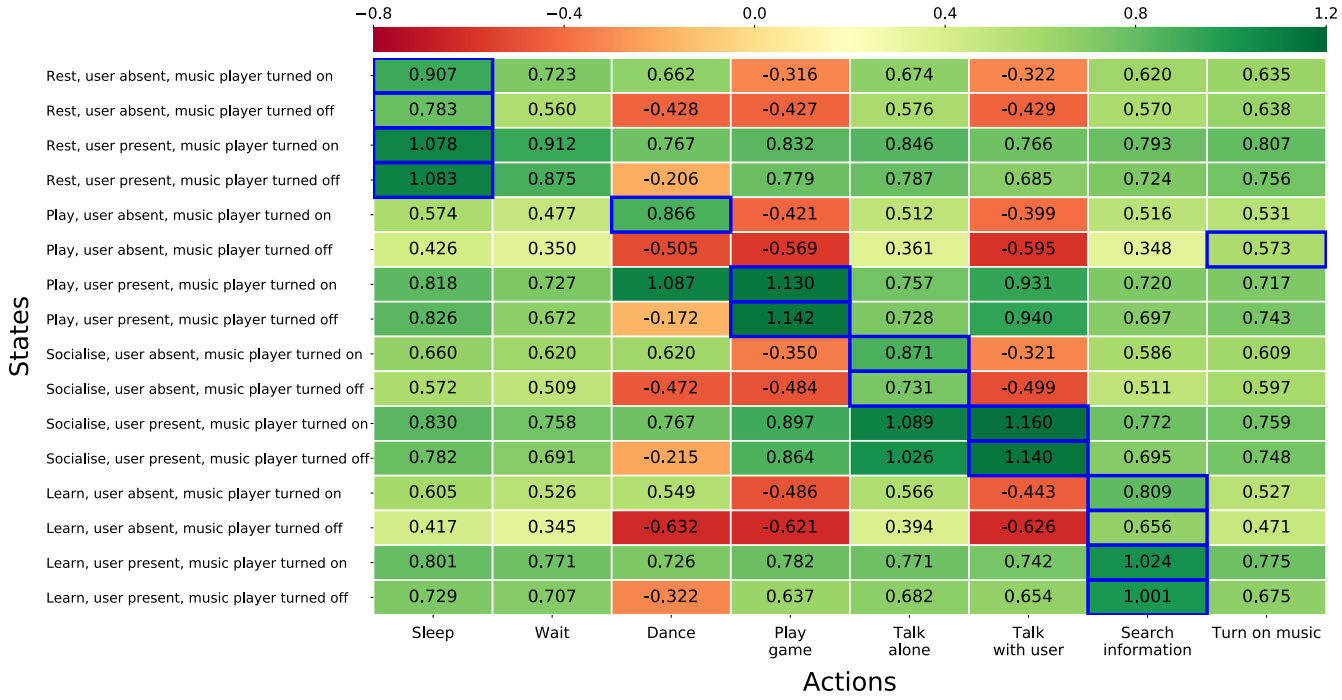


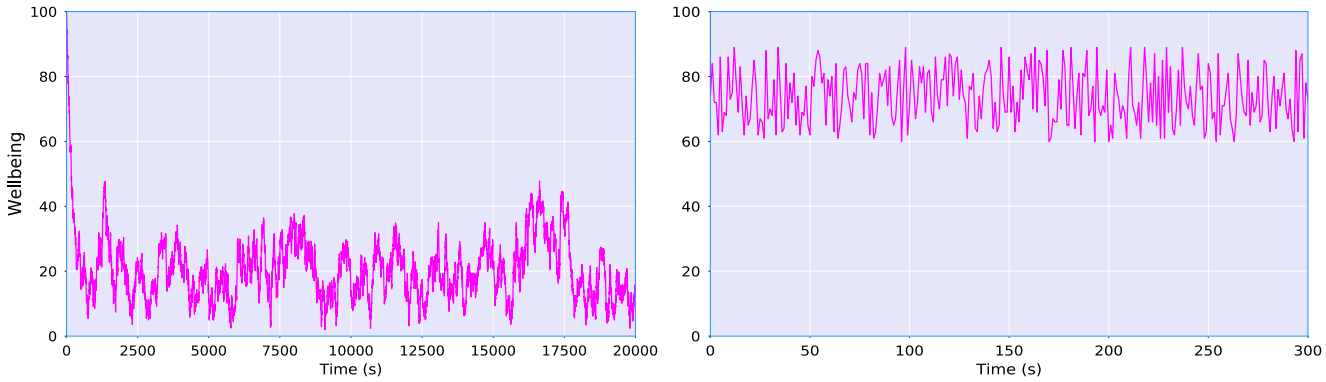
FIGURE 7. Q-values learnt by the Dyna-Q+ algorithm for each state-action pair. The optimal Q-values for each state-action pair are framed in blue. Actions with positive effects on the biological processes of the robot for each state are highlighted in green.

leads the robot to satisfy its deficits. Considering this, it is important to remark that the Q-values learnt by each algorithm must be compared among them, and not with the other algorithms. Thus, in some of the graphs, the final Q-value learnt by Q-learning and Q(λ) is above the Q-value learnt by Dyna-Q+. This may be a clue that neither of the classical algorithms has yet learnt an optimal policy of behaviour.

The optimal policy learnt by the robot is defined by the actions that present the higher Q-values for each state of the robot. Thus, the robot has learnt properly when it greedily selects the best actions. Figure 7 represents the final Q-values learnt by the robot using the Dyna-Q+ architecture, which has resulted in the one with the best performance in this experiment. Q-values are highlighted in a green to red colour scale, indicating the optimality of a certain Q-value depending on the state of the agent. In this representation, shining green indicates that the action is very optimal, producing beneficial effects in the robot's well-being when it is executed in that particular state. In contrast, dull reds represent the non-optimality of the action for a certain robot state, not reducing properly the salient deficits of the robot. Additionally, the best action of each state has been framed in blue, marking the optimal policy that the robot will follow when exploiting its learnt behaviour.

Following the blue frames highlighted in the table, it is possible to describe the behaviour learnt by Mini. For example, if the robot is motivated to Rest, independently if the user is present or if the music player is on, the robot learns that the best option is to Sleep to recover from Tiredness. Looking at those states where the dominant motivation is Play, the robot will play the quiz game if the user is present independently of the music player's state. However, if the user is absent and the music player is off, then Mini learns that to reduce its Entertainment need, it has to learn first that it is necessary to turn on the music player to be able dance. Regarding the motivation to Socialise, the robot will talk alone if any user is present because talking with the user is only successful if someone is perceived to be close to the robot. Finally, every time that Mini is motivated to Learn, the best action is to search for information because it is the action that reduces the most the robot's hunger of Knowledge. While interacting with the real world, an optimal internal state of the robot is obtained if it exploits this policy of behaviour. By doing so, it is able to rapidly reduce its needs as soon as they appear, maintaining its artificially-created internal state inside a comfortable range, as represented in the following section. In addition, the motivated behaviour exhibited by Mini could be perceived by many people as more "natural" because it expresses its needs and intentions according to a grounded reason, instead of just performing autonomous random behaviours.

In case of the algorithms Q-learning and Q(λ), the learning process does not attain an optimal solution. Considering Q-learning, the policy learnt by the robot is almost correct, excepting the robot state (Learn, user absent, music player turned off), where instead of learning that the best action is to search information on the Internet, the robot learns that it has to turn on the music player. Meanwhile, in case of Q(λ), the policy is incorrect for the state (Play, user absent, music player turned off), where the robot learns it has to



(a) Temporal evolution of the robot's well-being when the robot is exploring the environment executing a random behaviour.

(b) Temporal evolution of the robot's well-being of the robot when it is behaving following the optimal policy.

FIGURE 8. Well-being of the robot at the beginning and at the end of learning process.

sleep instead of turning on the music player. The full view of the policy learnt by the robot for Q-learning and Q(λ) can be found in the repository (referenced in footnote of Section VI-A), together with the rest of the results.

C. ROBOT WELL-BEING

The artificially created internal state of our robot seeks to define a biological ground to supply it with natural mechanisms of motivated behaviour. During the initial steps of the learning process, actions are randomly explored preventing the robot to correctly regulate its internal state. Nevertheless, as the robot gains experience with the interaction and it realizes the effect each action produces on its needs, Mini starts to select optimal actions instead of continue exploring. Thereby, once the robot had learnt how to correctly reduce its needs, the welfare state of the robot is notably improved. It is worth noting here that well-being values of 0 or 100 are unlikely to occur because it is almost impossible that all the biological process of the robot are at their highest deficit or totally satiated. Figure 8 compares the outcomes produced by the Dyna-Q+ model for evolution of the well-being of the robot when it is naive about how to regulate its internal state (Figure 8a) and once it has learnt its optimal policy (see Figure 8b).

Figure 8 represents the initial 300 minutes (the robot has not gained any experience) and Figure 8b the last 5 minutes of the learning process (once the robot has learnt the optimal policy of behaviour). As Figure 8a shows, the well-being state of the robot at the beginning of the experiment is very poor (mostly below 30 units), meaning that the needs of the robot are not correctly reduced. However, once the robot has learnt the optimal policy of behaviour, the well-being of Mini ranged between 60 and 85 units. Thus, as Figure 8b represents, the variation of the well-being signal is enclosed in a range between $60-85$ units, while in Figure 8a the well-being signal oscillates inside a wider range (values between $0-50$ units). This shows that once the robot has learnt how to behave, its internal state is more stable. Consequently, these results ground one of the main goals of this contribution because it represents how the welfare of the robot has been notably improved once the learning process has been completed. Intrinsically, this means that the deficits derived from the biological processes of the robot are correctly satisfied and controlled inside an acceptable range.

VII. DISCUSSION

According to the results presented in the previous section, and especially in the comparison shown in Section VI-A where the performance of the three algorithms is proven, we have been demonstrated how the probabilistic Dyna-Q+ model, combined with a classical multi-step Q-(λ) reinforcement learning algorithm, provides the best results in terms of convergence speed and stability. Consequently, this model was the chosen option to be embedded in the robot architecture to allow it to maintain the best possible internal state by exhibiting action-based learning capabilities.

The Dyna architecture has proven to be a promising framework to speed-up the learning process in reinforcement learning research. Among its many advantages, it allows the agent not only to speed-up the learning process but also to make classical algorithms less sensitive to changes in the reward distribution, especially in non-deterministic environments. In our application, action rewards follow a stochastic distribution, due to the problems derived from the errors produced by the perception system and to the unforeseen behaviours of the people while interacting with the robot. Consequently, this issue produces an important fluctuation in the evolution of the Q-values, as can be perceived for Q-learning and Q(λ) algorithms in the graphs represented in Figure 6, but particularly in Figures 6e, 6i, 6m, or 6n, among others.

Once the action learning has been completed, Mini is able to naturally behave by seeking to reduce its internal needs. As Figure 7 reflects, it has been able to learn the optimal policy (framed in blue) of behaviour and acknowledge that, for example, it cannot play a quiz game or talk with the user if they are not present and correctly perceived. In a similar

example, it is conscious that to dance, it is indispensable to turn on the music player first because otherwise it will not be possible to play any song. It is important to bear in mind that this learning process requires a user to be engaged with the robot because otherwise the correction of the robot's deficits cannot be tackled successfully. As previously mentioned in Section VI-A, Dyna-Q+ not only outperforms Q-learning and Q(λ) in terms of convergence speed but also in learning the optimal policy because both classical algorithms did not reach the optimal solution as Dyna-Q+ architecture did. In this approach, the dependency on the user is part of our own experimental set-up because the nature of our robot and its biological variables have been defined giving more importance to its social needs to foster human-robot interaction. As stated earlier, experimenters could behave at will during the experiment but while the robot was selecting and executing new actions, even in their absence, they were influenced many times by Mini causing them to approach it and start a new interaction. Additionally, the optimal policy leads the robot to maintain a good welfare state, as Figure 8 shows by comparing the evolution of the robot's well-being before and after learning the optimal policy of behaviour. As can be perceived by looking at Figure 8a, the well-being starts being ideal at the beginning of the experiment but rapidly decreases due to the incorrect reduction of the robot's needs. This tendency is maintained until the robot has learnt the optimal policy, driving it to correctly reduce its salient needs maintaining a good well-being and more stable value, as depicted in Figure 8b. Considering this optimal deficit reduction, we have proven how the robot correctly maintains its biological processes inside comfortable ranges, attaining a stable internal state while autonomously behaving.

At the initial stage of this experiment, the effects that each action produced on the internal state of the robot were empirically predefined using our previous experience in agent's motivational modelling, and taking into account the goal that we pursue for Mini. Despite this empirical predefinition, we realize that to attain a ''natural'' robot to interact with people, it is critical to endow it to a wide range of behaviours; as depicted in Figure 4. Moreover, it is important that the robot is able to reduce its deficits by itself and not only by depending on external resources, especially if these resources are not always reachable by the robot. For example, in the definition of this experiment, where the robot required people to reduce its Entertainment and Social interaction needs, we had to include actions (dance and talk alone) to allow it to reduce these needs independently on the presence of the user. Otherwise, in situations where the user is not engaged with the robot or is absent during long periods of time, the robot can enter in a looping situation where the other needs cannot be reduced.

Finally, to conclude this discussion, we would like to remark on the importance of behaviour adaptation in social robots. In this work, we have presented a new approach to speed-up action learning to endow Mini with the possibility to dynamically regulate its motivational behaviour. This regulation depends on two external factors (the user and the music player), but there are still many stimuli that influence people on their lives and therefore are potential factors of influence in adaptive artificial agents. In this sense, the benefits provided in learning speed by Dyna-Q+ will lead us to improve the situations and considerations of our robot, which will enhance its 'artificial' intelligence and natural behaviour.

A. LIMITATIONS

The architecture presented in this manuscript has some limitations in its operation. First, as was mentioned earlier, tabular methods are only tractable if the state-action space is not too large and can easily be represented as a table. Otherwise, function approximation methods are required, as described in Section II. In addition, the success of RL algorithms rely on a correct definition of the reward function, which is necessary to clearly specify the goal of the system during its design.

In this application, where our social robot Mini learns from trial and error how to behave to maintain a good internal state, the effects of each individual behaviour on the biological processes of the robot have to be finely and empirically set. This issue means that the designer of the robot's behaviour has to precisely define the existing relationships between the biological processes of the robot and the behaviours. If the number of behaviours and processes increases, then the designing process becomes more tedious.

VIII. CONCLUSION

In this work, we have presented a biologically inspired action learning system for our social robot Mini supported on Dyna-Q+ architecture, which combines the classical reinforcement learning's Q(λ) with a probabilistic model. This model replicates the robot state's transitions and rewards obtained after executing an action in a particular state. Thus, as was demonstrated in the results section, the learning process of the robot is much faster and more stable than classical algorithms such as Q-learning or Q(λ) because the probabilistic model of the robot plans in the background while the robot is acting in the real world. In addition, Dyna-Q+ architecture allows the robot to continue exploring the environment by recognising changes that appear in it, even after the initial learning has finished.

This study represents the next step of our previous works that are based on endowing Mini with autonomous bioinspired behaviour-based capabilities in a real human-robot interaction domain. Our future work will provide Mini with more actions and biological processes to give the robot a more natural interaction with people and other important stimulus available in the environment. Our results are necessary to explore and understand how the organism of living beings operate and how a new stimulus in the environment can be perceived and appraised to provide our robot with a wider range of capabilities. Accordingly, accomplishing natural behaviours and decision-making in Mini will allow us to deploy more social robots in homes, which will gather

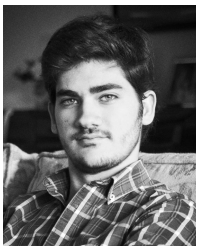
information about how people perceive the robot and whether they really engage with it.

REFERENCES

[1] J. H. Connell and S. Mahadevan, *Robot Learning*, vol. 233. USA: Springer, 2012.
[2] S. Thrun and T. M. Mitchell, "Lifelong robot learning," *Robot. Auton. Syst.*, vol. 15, nos. 1–2, pp. 25–46, Jul. 1995.
[3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.
[4] S. Frennert and B. Östlund, "Seven matters of concern of social robots and older people," *Int. J. Social Robot.*, vol. 6, no. 2, pp. 299–310, 2014.
[5] I. Pedersen, S. Reid, and K. Aspevig, "Developing social robots for aging populations: A literature review of recent academic sources," *Sociol. Compass*, vol. 12, no. 6, Jun. 2018, Art. no. e12585.
[6] L. Pu, W. Moyle, C. Jones, and M. Todorovic, "The effectiveness of social robots for older adults: A systematic review and meta-analysis of randomized controlled studies," *Gerontologist*, vol. 59, no. 1, pp. e37–e51, Jan. 2019.
[7] M. Scheutz, G. Briggs, R. Cantrell, E. Krause, T. Williams, and R. Veale, "Novel mechanisms for natural human-robot interactions in the DIARC architecture," in *Proc. AAAI Workshop Intell. Robot. Syst.*, 2013, p. 66.
[8] F. Ferland, D. Létourneau, A. Aumont, J. Frémy, M.-A. Legault, M. Lauria, and F. Michaud, "Natural interaction design of a humanoid robot," *J. Hum.-Robot Interact.*, vol. 1, no. 2, pp. 118–134, Jan. 2013.
[9] D. Cañamero, "Modeling motivations and emotions as a basis for intelligent behavior," in *Proc. 1st Int. Conf. Auton. Agents (AGENTS)*, 1997, pp. 148–155.
[10] R. Arkin, "Moving up the food chain: Motivation and emotion in behavior based robots," in *Who Needs Emotions: The Brain Meets the Robot*, J. Fellous and M. Arbib, Eds. Oxford Univ. Press, Oxford, U.K., 2005, pp. 245–270, doi: 10.1093/acprof:oso/9780195166194.003.0009.
[11] M. Malfaz, Á. Castro-González, R. Barber, and M. A. Salichs, "A biologically inspired architecture for an autonomous and social robot," *IEEE Trans. Auton. Mental Develop.*, vol. 3, no. 3, pp. 232–246, Sep. 2011.
[12] M. A. Salichs, A. Castro-González, E. Salichs, E. Fernández-Rodicio, M. Maroto-Gómez, J. J. Gamboa-Montero, S. Marques-Villarroya, J. C. Castillo, F. Alonso-Martín, and M. Malfaz, "Mini: A new social robot for the elderly," *Int. J. Social Robot.*, vol. 12, pp. 1–19, Sep. 2020.
[13] M. Malfaz and M. A. Salichs, "Learning to deal with objects," in *Proc. IEEE 8th Int. Conf. Develop. Learn.*, Jun. 2009, pp. 1–6.
[14] Á. Castro-González, M. Malfaz, and M. A. Salichs, "Learning the selection of actions for an autonomous social robot by reinforcement learning based on motivations," *Int. J. Social Robot.*, vol. 3, no. 4, pp. 427–441, Nov. 2011.
[15] A. Castro-González, M. Malfaz, J. F. Gorostiza, and M. A. Salichs, "Learning behaviors by an autonomous social robot with motivations," *Cybern. Syst.*, vol. 45, no. 7, pp. 568–598, 2014.
[16] M. Maroto-Gómez, Á. Castro-González, J. Castillo, M. Malfaz, and M. Salichs, "A bio-inspired motivational decision making system for social robots based on the perception of the user," *Sensors*, vol. 18, no. 8, p. 2691, Aug. 2018.
[17] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *SIGART Bull.*, vol. 2, pp. 160–163, Jul. 1991.
[18] C. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.
[19] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Mach. Learn.*, vol. 22, nos. 1–3, pp. 123–158, Mar. 1996.
[20] K. Dautenhahn, S. Woods, C. Kaouri, M. L. Walters, K. L. Koay, and I. Werry, "What is a robot companion—Friend, assistant or butler?" in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 1192–1197.
[21] J. Broekens, M. Heerink, and H. Rosendal, "Assistive social robots in elderly care: A review," *Gerontechnology*, vol. 8, no. 2, pp. 94–103, 2009.
[22] R. D. Schraft, C. Schaeffer, and T. May, "Care-o-bot: A system for assisting elderly or disabled persons in home environments," in *Proc. 24th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, vol. 4, Aug. 1998, pp. 2476–2481.
[23] V. Gonzalez-Pacheco, A. Ramey, F. Alonso-Martín, A. Castro-Gonzalez, and M. A. Salichs, "Maggie: A social robot as a gaming platform," *Int. J. Social Robot.*, vol. 3, no. 4, pp. 371–381, Nov. 2011.
[24] I. Aaltonen, A. Arvola, P. Heikkilä, and H. Lammi, "Hello pepper, may I tickle you? Children's and adults' responses to an entertainment robot at a shopping mall," in *Proc. Companion ACM/IEEE Int. Conf. Hum.-Robot Interact.*, Mar. 2017, pp. 53–54.
[25] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, and F. Tanaka, "Social robots for education: A review," *Sci. Robot.*, vol. 3, no. 21, Aug. 2018, Art. no. eaat5954.
[26] S. Dominguez, E. Zalama, J. G. García-Bermejo, and J. Pulido, "Robot learning in a social robot," in *Proc. Int. Conf. Simulation Adapt. Behav.* Berlin, Germany: Springer, 2006, pp. 691–702.
[27] J. Hemminghaus and S. Kopp, "Towards adaptive social behavior generation for assistive robots using reinforcement learning," in *Proc. 12th ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, Mar. 2017, pp. 332–340.
[28] L. Takayama and C. Pantofaru, "Influences on proxemic behaviors in human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 5495–5502.
[29] P.-H. Ciou, Y.-T. Hsiao, Z.-Z. Wu, S.-H. Tseng, and L.-C. Fu, "Composite reinforcement learning for social robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2553–2558.
[30] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1343–1350.
[31] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3052–3059.
[32] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10357–10377, 2021.
[33] H. Ritschel, T. Baur, and E. Andre, "Adapting a robot's linguistic style based on socially-aware reinforcement learning," in *Proc. 26th IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, Aug. 2017, pp. 378–384.
[34] A. Tapus, C. Ţăpuş, and M. J. Matarić, "User—Robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy," *Intell. Service Robot.*, vol. 1, p. 169, Feb. 2008.
[35] H. W. Park, I. Grover, S. Spaulding, L. Gomez, and C. Breazeal, "A model-free affective reinforcement learning approach to personalization of an autonomous social robot companion for early literacy education," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 687–694.
[36] C. Liu, K. Conn, N. Sarkar, and W. Stone, "Online affect detection and robot behavior adaptation for intervention of children with autism," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 883–896, Aug. 2008.
[37] M. J. Matarić, "Learning social behavior," *Robot. Auton. Syst.*, vol. 20, nos. 2–4, pp. 191–204, 1997.
[38] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro, "Robot gains social intelligence through multimodal deep reinforcement learning," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2016, pp. 745–751.
[39] H. Ritschel, "Socially-aware reinforcement learning for personalized human-robot interaction," in *Proc. 17th Int. Conf. Auton. Agents Multi-Agent Syst.*, 2018, pp. 1775–1777.
[40] K. Weber, H. Ritschel, I. Aslan, F. Lingenfelser, and E. André, "How to shape the humor of a robot–social behavior adaptation based on reinforcement learning," in *Proc. 20th ACM Int. Conf. Multimodal Interact.*, Oct. 2018, pp. 154–162.
[41] H.-G. Ren, X.-G. Ruan, and X.-Y. Li, "Bionic self-learning of two-wheeled robot based on skinner's operant conditioning," in *Proc. Int. Conf. Measuring Technol. Mechatronics Automat.*, vol. 1, 2009, pp. 389–392.
[42] I. Ranatunga, J. Rajruangrabin, D. Popa, and F. Makedon, "Enhanced therapeutic interactivity using social robot Zeno," in *Proc. 4th Int. Conf. Pervasive Technol. Rel. Assistive Environ.*, May 2011, p. 57.
[43] C. Hu, M. Xu, and K.-S. Hwang, "An adaptive cooperation with reinforcement learning for robot soccer games," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 3, pp. 1–9, May 2020.
[44] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[45] S. Al Dabooni and D. Wunsch, "Heuristic dynamic programming for mobile robot path planning based on dyna approach," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 3723–3730.
[46] B. Peng, X. Li, J. Gao, J. Liu, K.-F. Wong, and S.-Y. Su, "Deep dyna-Q: Integrating planning for task-completion dialogue policy learning," 2018, *arXiv:1801.06176*. [Online]. Available: https://arxiv.org/abs/1801.06176

[47] Y. Hayamizu, S. Amiri, K. Chandan, S. Zhang, and K. Takadama, "Guided dyna-Q for mobile robot exploration and navigation," 2020, *arXiv:2004.11456*. [Online]. Available: https://arxiv.org/abs/2004.11456
[48] H. Lee and J. Jeong, "Mobile robot path optimization technique based on reinforcement learning algorithm in warehouse environment," *Appl. Sci.*, vol. 11, no. 3, p. 1209, Jan. 2021.
[49] H. Li, Z. Qichao, and D. Zhao, "Deep reinforcement learning-based automatic exploration for navigation in unknown environment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2064–2076, Jun. 2020.
[50] Y. Ozaki, T. Ishihara, N. Matsumura, and T. Nunobiki, "Can user-centered reinforcement learning allow a robot to attract passersby without causing discomfort?" 2019, *arXiv:1903.05881*. [Online]. Available: http://arxiv.org/abs/1903.05881
[51] Y. Gao, F. Yang, M. Frisk, D. Hemandez, C. Peters, and G. Castellano, "Learning socially appropriate robot approaching behavior toward groups using deep reinforcement learning," in *Proc. 28th IEEE Int. Conf. Robot Hum. Interact. Commun. (RO-MAN)*, Oct. 2019, pp. 1–8.
[52] H.-D. Bui and N. Y. Chong, "Autonomous speech volume control for social robots in a noisy environment using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 1263–1268.
[53] Q. Fu and A. Song, "Adaptive modulation for underwater acoustic communications based on reinforcement learning," in *Proc. OCEANS MTS/IEEE Charleston*, Oct. 2018, pp. 1–8.
[54] R. Su, Z. Gong, D. Zhang, C. Li, Y. Chen, and R. Venkatesan, "An adaptive asynchronous wake-up scheme for underwater acoustic sensor networks using deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1851–1865, Feb. 2021.
[55] M. Min, X. Wan, L. Xiao, Y. Chen, M. Xia, D. Wu, and H. Dai, "Learning-based privacy-aware offloading for healthcare IoT with energy harvesting," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4307–4316, Jun. 2019.
[56] F. Wang, J. Gao, M. Li, and L. Zhao, "Autonomous PEV charging scheduling using dyna-Q reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12609–12620, Nov. 2020.
[57] L. Dong, Y. Li, X. Zhou, Y. Wen, and K. Guan, "Intelligent trainer for dyna-style model-based deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2758–2771, Jun. 2021.
[58] M. Van Otterlo and M. Wiering, "Reinforcement learning and Markov decision processes," in *Reinforcement Learning*. Berlin, Germany: Springer, 2012, pp. 3–42.
[59] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.
[60] C. Cercignani, "The Boltzmann equation," in *The Boltzmann Equation and Its Applications*. New York, NY, USA: Springer, 1988, pp. 40–103.
[61] B. M. Blumberg, P. M. Todd, and P. Maes, "No bad dogs: Ethological lessons for learning in hamsterdam," in *Proc. 4th Int. Conf. Simulation Adapt. Behav. From Animals Animats*. Cambridge, MA, USA: MIT Press, 1996, pp. 295–304.
[62] P.-L. Hsu and H. Robbins, "Complete convergence and the law of large numbers," *Proc. Nat. Acad. Sci. USA*, vol. 33, no. 2, p. 25, 1947.
[63] J. Lones, M. Lewis, and L. Cañamero, "A hormone-driven epigenetic mechanism for adaptation in autonomous robots," *IEEE Trans. Cognit. Develop. Syst.*, vol. 10, no. 2, pp. 445–454, Jun. 2018.

MARCOS MAROTO-GÓMEZ received the B.Sc. degree in industrial electronics and automation engineering from the University of Castilla-La Mancha, Toledo, Spain, in 2015, and the M.Sc. degree in robotics and automation from the Carlos III University of Madrid, Madrid, Spain, in 2016. He is currently a Researcher and an Assistant Professor with the Carlos III of Madrid University. He currently researches at the Robotics Lab Research Group, related to human–robot interaction, decision-making, adaptation, autonomy, and machine learning applied to social robots.



RODRIGO GONZÁLEZ received the B.Sc. degree in industrial technologies engineering and the M.Sc. degree in industrial engineering from the Carlos III University of Madrid, Madrid, Spain, in 2018 and 2020, respectively. He is currently a part of the Robotics Lab Research Group, where he has developed his master's thesis, related to the application of reinforcement learning techniques in social robots.

ÁLVARO CASTRO GONZÁLEZ received the B.Sc. degree in computer engineering from the University of León, León, Spain, in 2005, and the M.Sc. and Ph.D. degrees in robotics and automation from the Carlos III University of Madrid, Madrid, Spain, in 2008 and 2012, respectively. He is currently an Assistant Professor with the Department of Systems Engineering and Automation, Carlos III University of Madrid. He has been involved in several national, European, and corporate sponsored research projects. His present research interests include related to human–robot interaction, social robots, expressiveness in robots, decision-making, and artificial emotions. He is a member of the Robotics Lab Research Group.

MARÍA MALFAZ received the degree in physics science from La Laguna University, in 1999, the M.Sc. degree in control systems from the Imperial College of London, in October 2001, and the Ph.D. degree in industrial engineering, in 2007, with a focus on decision making system for autonomous social agents based on emotions and self-learning. She is currently a Full Professor with the Systems Engineering and Automation Department, Carlos III University of Madrid. Her research area follows the line carried out in her thesis and, more recently, she has been working on multimodal human–robot interaction systems. She is a member of research networks, such as European Robotics Coordination Action (EURobotics) and Plataforma Tecnológica Española de Robótica (HispaRob). She belongs to several international scientific associations, such as IEEE Robotics and Automation Society (IEEE-RAS), International Association of Automatic Control (IFAC), and Comité Español de Automática (CEA).

MIGUEL ÁNGEL SALICHS (Senior Member, IEEE) received the degree in electrical engineering and the Ph.D. degree from the Polytechnic University of Madrid. He is currently a Professor of the Systems Engineering and Automation Department, Carlos III University of Madrid. His research interests include autonomous social robots, multimodal human–robot interaction, mind models, and cognitive architectures. He was a member of the Policy Committee of the International Federation of Automatic Control (IFAC), the Chairman of the Technical Committee on Intelligent Autonomous Vehicles of IFAC, a responsible of the Spanish National Research Program on Industrial Design, a Production Member of the Spanish Society on Automation and Control (CEA), and the Spanish Representative with the European Robotics Research Network (EURON). He is the Coordinator of the Secretariat of the Spanish Robotics Technology Platform (HispaRob). He is an Associate Editor of the *International Journal of Social Robotics*.