Active exploration and parameterized reinforcement learning applied to a simulated human-robot interaction task

Mehdi Khamassi
Institute of Intelligent Systems and Robotics
Sorbonne Universités, UPMC Univ Paris 06, CNRS
F-75005 Paris, France
Email: mehdi.khamassi@upmc.fr

George Velentzas, Theodore Tsitsimis, Costas Tzafestas
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
Email: ktzaf@cs.ntua.gr

***Abstract*—Online model-free reinforcement learning (RL) methods with continuous actions are playing a prominent role when dealing with real-world applications such as Robotics. However, when confronted to non-stationary environments, these methods crucially rely on an exploration-exploitation trade-off which is rarely dynamically and automatically adjusted to changes in the environment. Here we propose an active exploration algorithm for RL in structured (parameterized) continuous action space. This framework deals with a set of discrete actions, each of which is parameterized with continuous variables. Discrete exploration is controlled through a Boltzmann softmax function with an inverse temperature β parameter. In parallel, a Gaussian exploration is applied to the continuous action parameters. We apply a meta-learning algorithm based on the comparison between variations of short-term and long-term reward running averages to simultaneously tune β and the width of the Gaussian distribution from which continuous action parameters are drawn. We first show that this algorithm reaches state-of-the-art performance in the non-stationary multi-armed bandit paradigm, while also being generalizable to continuous actions and multi-step tasks. We then apply it to a simulated human-robot interaction task, and show that it outperforms continuous parameterized RL both without active exploration and with active exploration based on uncertainty variations measured by a Kalman-Q-learning algorithm.**

I. Introduction

Important progresses have been made in recent years in reinforcement learning (RL) with continuous action spaces, permitting successful real-world applications such as Robotics applications [1], [2]. Nevertheless, a recent review on reinforcement learning applied to Robotics [3] highlighted, among other points, that (i) a variety of algorithms have been developed, each being appropriate to specific tasks: model-based versus model-free (which we have previously addressed [4], [5] but do not further consider here), function approximation versus policy search, continuous versus discrete action spaces (of particular interest here); (ii) important human knowledge is injected concerning the search in the parameter space, either by reducing it through learning from demonstration, or by pre-adjusting parameters such as the exploration rate based on the prior determination of the total number of episodes in the experiment. In particular, the balance between exploration and exploitation is often pre-determined with human prior knowledge and does not extend well to tasks with non-stationary reward functions.

To address the first issue about continuous versus discrete action spaces, the recent proposal of RL algorithms in structured Parameterized Action Space Markov Decision Processes (PAMDP) [6], [7] seems to open a promising line of research. It combines a set of discrete actions $A_d = \{a_1, a_2, ..., a_k\}$, each action $a \in A_d$ featuring m_a continuous parameters $\{\theta^a_1, ..., \theta^a_{m_a}\} \in \mathbb{R}^{m_a}$. Actions are thus represented by tuples $(a, \theta^a_1, ..., \theta^a_{m_a})$ and the overall action space is defined as $A = \cup_{a \in A_d}(a, \theta^a_1, ..., \theta^a_{m_a})$. This framework has been successfully applied to simulations of a Robocup 2D soccer task where agents have to learn to timely select between discrete actions such as running, turning or kicking the ball, and should learn at the same time with which speed to run, which angle to turn or which strength to kick. To ensure algorithm convergence, [6] alternate between learning phases: (i) given a fixed policy for parameter selection, they use Q-Learning to optimize the policy discrete action selection; (ii) Next, they fix the policy for discrete action selection and use a policy search method to optimize the parameter selection. In contrast, [7] learn both in parallel by employing a parameterized actor that learns both discrete actions and parameters, and a parameterized critic that learns only the action-value function. Instead of relying on an external policy search procedure, they are thus able to directly query the critic for gradients.

Nevertheless, the exploration-exploitation trade-off is fixed in these methods, thus falling into the second issue raised by [3]'s review. Exploration in continuous action spaces being different from discrete spaces, [7] adapt ϵ-greedy exploration to parameterized action space by picking a random discrete action $a \in A_d$ with probability ϵ and sampling the action's parameters θ^a_i from a uniform random distribution. ϵ is arbitrarily annealed from 1.0 to 0.1 over the first 10,000 updates, thus requiring human prior knowledge about the duration of the task to appropriately tune exploration.

The original contribution of this paper consists in adapting existing methods, combining them together and applying them

to a simple human-robot interaction scenario in the following manner: We use the Gaussian exploration for continuous action parameters proposed by [8], which in the original formulation uses a fixed Gaussian width σ. We then apply a noiseless version of the meta-learning algorithm of [9], which tracks online variations of the agent's performance measured by short-term and long-term reward running averages. At each timestep, we use the difference between the two averages to simultaneously tune the inverse temperature β_t used for selecting between discrete actions a_j, and the width σ_t of the Gaussian distribution from which each continuous action parameter θ_i^a is sampled around its current value. We first test this algorithm in a standard non-stationary (i.e. switching) multi-armed bandit paradigm proposed by [10]. We show that it reaches similar performance to one of the state-of-the-art upper confidence bound algorithms, while also being generalizable to continuous actions and multi-step tasks (which is not the case for bandit methods). We then apply the proposed algorithm to a simple simulated human-robot interaction task, where the algorithm tries to maximise reward computed as the virtual engagement of the human in the task, this engagement representing the attention that the human pays to the robot actions. We show that the proposed algorithm outperforms continuous parameterized RL both without active exploration and with active exploration based on uncertainty variations measured by a Kalman-RL algorithm [11].

II. Active exploration algorithm

This section describes the mathematical formulation underlying the proposed active exploration method. The proposed meta-learning algorithm is then summarised at the end of the section (Algorithm 1). It first employs Q-Learning [12] to learn the value of discrete action $a_t \in A_d$ selected at timestep t in state s_t:

$$\delta_t = r_t + \gamma \max_a(Q_t(s_{t+1}, a)) - Q_t(s_t, a_t) \tag{1}$$

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha_Q \delta_t \tag{2}$$

where α_Q is a learning rate and γ is a discount factor. The probability of executing discrete action a_j at timestep t is given by a Boltzmann softmax equation:

$$P(a_j|s_t, \beta_t) = \frac{exp\left(\beta_t Q_t(s_t, a_j)\right)}{\sum_a exp\left(\beta_t Q_t(s_t, a)\right)} \tag{3}$$

where β_t is a dynamic inverse temperature meta-parameter which will be tuned through meta-learning (see below).

In parallel, continuous parameters $\widetilde{\theta}_{i,t}^{a_j}$ with which action a_j is executed at timestep t are selected from a Gaussian exploration function centered on the current values $\theta_{i,t}^{a_j}(s_t)$ in state s_t of the parameters of this action [8]:

$$P(\widetilde{\theta}_{i,t}^{a_j}|s_t, a_j, \sigma_t) = \frac{1}{\sqrt{2\pi}\sigma_t} exp\left(-(\widetilde{\theta}_{i,t}^{a_j} - \theta_{i,t}^{a_j}(s_t))^2/(2\sigma_t^2)\right) \tag{4}$$

where the width σ_t of the Gaussian is a meta-parameter which will be tuned through meta-learning (see below) and action parameters $\theta_{i,t}^a(s_t)$ are learned with a continuous actor-critic algorithm [8]. A reward prediction error is computed from the critic: $\delta_t = r_t + \gamma V_t(s_{t+1}) - V_t(s_t)$ and is used to update the parameter vectors ω_t^C and ω_t^A of the neural network function approximations in the critic and the actor:

$$\omega_{i,t+1}^C = \omega_{i,t}^C + \alpha_C \delta_t \frac{\delta V_t(s_t)}{\delta \omega_{i,t}^C} \tag{5}$$

$$\omega_{i,t+1}^A = \omega_{i,t}^A + \alpha_A \delta_t (\widetilde{\theta}_{i,t}^a - \theta_{i,t}^a(s_t)) \frac{\delta \theta_{i,t}^a(s_t)}{\delta \omega_{i,t}^A} \tag{6}$$

where α_C and α_A are learning rates. In contrast to the original version where ω_t^A updates are performed only when $\delta_t > 0$ [8] – which occasionally led to divergence in our simulations –, here we update them all the time and proportionally to δ_t as in [13].

Finally, in order to perform active exploration, we need to dynamically update β_t and σ_t through a meta-learning process based on variations of the robot's performance. The idea is that increases in the average reward obtained by the robot can be interpreted as improvement of performance which can thus result in increasing the exploitation of learned action values [9], [14]. Conversely, drops in the average reward can be interpreted as signs of a change in the task conditions and thus as a need to re-explore. Nevertheless, the average reward is not an absolute measure and should rather be considered relatively to a reference such as the estimated average value of the task [15]. For instance, in tasks where only punishments are received, the average value of the task is negative, but should not be interpreted as an indication that the robot should only explore and never exploit. Thus here, following the proposition of [9], we measure a long-term reward running average $\bar{\bar{r}}_t$ serving as reference, and a short-term one $\bar{r}_t$ serving as current measure of performance. When $\bar{r}_t > \bar{\bar{r}}_t$, this means that the current performance is above average and that exploration can be decreased. When $\bar{r}_t < \bar{\bar{r}}_t$, this means that the current performance is below average and that exploration should be increased. Contrary to the noisy version of [9] which can lead to meta-learning instability, here we implement a noiseless version of the algorithm. We compute short- and long-term reward running averages in the following manner:

$$\Delta \bar{r}_t = (r_t - \bar{r}_t)/\tau_1 \text{ and } \Delta \bar{\bar{r}}_t = (\bar{r}_t - \bar{\bar{r}}_t)/\tau_2 \tag{7}$$

where τ_1 and τ_2 are two time constants. We then update β_t and σ_t with:

$$\beta_{t+1} = (\mathcal{R} \circ \mathcal{F})(\beta_t, \mu\tau_2 \Delta \bar{\bar{r}}_t) \text{ and } \sigma_{t+1} = \mathcal{G}(\mu\tau_2 \Delta \bar{\bar{r}}_t) \tag{8}$$

where $\mathcal{R}(x)$ is a rectifier, $\mathcal{F}(x, y)$ is affine, μ is a learning rate and $0 < \mathcal{G}(x) < 0.1M$ is a sigmoid, with M denoting the parameter range.

We also compared this meta-learning algorithm with the Kalman Q-Learning proposed by [11]. We first tested the original formulation which proposes a purely exploratory agent by replacing Q-values in Equation 3 by the action-specific diagonal terms of the covariance matrix – these terms representing the current variance/uncertainty about an action's Q-value. We then tested an extended version of the algorithm were diagonal terms of the covariance matrix are treated as *exploration bonuses* b_t^a which, like in a previous computational neuroscience work [16], are multiplied by a weight η and added to Q-values in Equation 3. A particular novelty here is that we also use the covariance terms b_t^a in replacement of $\bar{\bar{r}}_t$ in Equation 8 to tune action-specific σ_t^a with function $\mathcal{G}(x)$. As the result section will show, this turns out to be much more efficient in our task than the original purely exploratory agent proposed in [11]. This nevertheless does not outperform the meta-learning algorithm proposed in this article.

Algorithm 1 Active exploration with meta-learning

```
Initialize ω^A_{i,0}, ω^C_{i,0}, Q_{i,0}, β_0 and σ_0
for t = 0, 1, 2, ... do
    Select discrete action a_t with softmax(s_t, β_t) (Eq. 3)
    Select action parameters θ̃^a_{i,t} with GaussianExploration(s_t, a_t, θ^a_{i,t}, σ_t) (Eq. 4)
    Observe new state and reward {s_{t+1}, r_{t+1}} ← Transition(s_t, a_t, θ̃^a_{i,t})
    Update Q_{t+1}(s_t, a_t) in the discrete Q-Learning (Eq. 2)
    Update function approx. ω^C_{i,t+1} and ω^A_{i,t+1} in continuous actor-critic (Eq. 5, Eq. 6)
    if meta-learning then
        Update reward running averages r̄_t and r̄̄_t (Eq. 7)
        Update β_{t+1} and σ_{t+1} (Eq. 8)
    end if
end for
```

III. Experiments

A. Non-stationary multi-armed bandit

While multi-armed bandit problems have different setups, they can be formulated as having a set of arms $\mathcal{K} = \{1, ..., K\}$, each of them attached to a gambling machine. At every episode $t \in \mathcal{T}$, with $\mathcal{T} = \{1, ..., T\}$ denoting the sequence of decision episodes, the decision maker pulls an arm $a \in \mathcal{K}$ and receives a reward $r_t(a)$ with some unknown probability $p_t(a)$, and zero otherwise.

Here we compare our algorithm to bandit methods for two reasons: (i) one can consider the multi-armed bandit problem as a generalized fundamental benchmark for the evaluation of algorithms at the lower level of reinforcement learning framework; and (ii) the simple single-state human-robot interaction task that we will use in the next section can be seen as a non-stationary multi-armed bandit task extended to continuous action parameters. Thus we first evaluated the performance of our algorithm, simplified appropriately for such a case (i.e. only considering discrete actions without continuous parameters), on a non-stationary 3-armed bandit task with binary rewards adapted from [10] (Figure 1). Despite its multi-state nature, we provide evidence for its performance, by comparing with SW-UCB [10], D-UCB [17] and UCB1 [18], with the former two constituting proven efficient algorithms on non-stationary cases, on the same *swithing* setup used by [10], where $K = 3$, $T = 10000$, the rewards are binary $r_t \in \{0, 1\}$, $p_t(1) = 0.5$, $p_t(2) = 0.3$, $p_t(3) = 0.9$ for $3000 \leq t < 5000$ and $p_t(3) = 0.3$ otherwise.

SW-UCB has previously shown to achieve better results than D-UCB for the above test case, by using a sliding window of width τ episodes, as memory of the history of rewards and actions taken. Let us write a composite form for all three, as:

$$N_t(a; \gamma, \tau) = \sum_{s=t-\tau+1}^{t} \gamma^{t-s} \mathbb{1}_{\{a_s = a\}} \tag{9}$$

where a_s denotes the action taken at episode s, $\gamma \in (0, 1]$ a discount factor and τ the width of the memory window in number of episodes. The *average discounted windowed reward* can then be written as:

$$\overline{R}_t(a; \gamma, \tau) = \frac{1}{N_t(a)} \sum_{s=t-\tau+1}^{t} \gamma^{t-s} r_s(a) \mathbb{1}_{\{a_s = a\}} \tag{10}$$

where $r_s(a)$ denotes the reward taken at episode $s \in \mathcal{T}$ from action $a \in \mathcal{K}$. The *padding function*, also called exploration bonus, can be written as:

$$c_t(a; \gamma, \tau, \xi) = B \sqrt{\frac{\xi}{N_t(a)} log(\min(\sum_{a=1}^{K} N_t(a), \tau))} \tag{11}$$

where ξ is related to the rate at which exploration decreases in time and B relates to the upper bound of rewards. The action chosen by the decision maker is determined by:

$$a_t = \arg\max_a \left(\overline{R}_t(a) + c_t(a) \right) \tag{12}$$

For SW-UCB the width of the window τ is constant, the discount factor $\gamma = 1$, meaning no discount occurs, and B is the upper bound for the rewards. When the window size changes dynamically such that $\tau = t$, it falls down to UCB1.

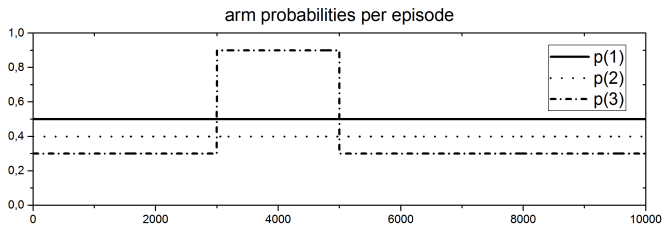

Fig. 1. Probability that an arm a will return a reward upon choice in the non-stationary multi-armed bandit task tested here. Adapted from [10].

Additionally when $\gamma \neq 1$, B is twice the reward upper bound and $\tau \geq T$, it changes to D-UCB. For the simulations, ξ, γ and τ, were chosen as proposed in [10].

For our implementation we used the Boltzmann softmax of Equation 3 for action selection, while updated Q-values according to:

$$Q_{t+1}(a_t) \leftarrow (1 - \alpha_Q)Q_t(a_t) + \alpha_Q r_t(a_t) \tag{13}$$

using $\alpha_Q = 0.4$. We then updated the inverse temperature parameter of the softmax by using a simplistic affine for $\mathcal{F}$ of Equation 8, leading to the following iterative procedure:

$$\beta_{t+1} \leftarrow \max\{0, \beta_t + \mu\tau_2\Delta\bar{\bar{r}}_t + \epsilon\} \tag{14}$$

using $\mu = 0.25$, $\tau_1 = 20$, $\tau_2 = 300$ and ϵ a very small constant to ensure increment of exploitation on long stationary intervals. For finding optimal parameters, we run the simulation on a large scale parameter grid space, observed robust areas with both low mean and variance of final cumulative regret, and rerun on smaller denser areas until no significant fluctuations of performance occurred. We then repeated our simulations for 500 sessions, computed the averaged total regret per episode, the final cumulative regret and the final cumulative reward for each session. Comparing our results with UCB1, SW-UCB and D-UCB as seen in Figure 2, we observed a performance comparable to that of SW-UCB.

In more detail, from episodes 1 to 3000, UCB1 outperforms all others as expected. After the first switch though, when the gap between the expected value of the optimal arm and the second best arm is large, meta-learning outperforms all others as observed by the flat line from 3500 to 5000, reaching "no regret" performance, while SW-UCB suffers from exploration. After the second switch, the gap becomes small again, and SW-UCB demonstrated better performance in such case.

Providing analytical mathematical details and guarantees about convergence falls out of the scope of the present work. Nevertheless, these results confirm that our algorithm is relevant for non-stationary problems and suggest a promising performance at the lower level of adaptation of the reinforcement learning framework.

B. Simple HRI simulation

We then test the algorithm described in Section 2 in a simple simulated human-robot interaction task involving a single state, 6 discrete actions, and continuous action parameters between -100 and 100. The task is similar to a non-stationary stochastic multi-armed bandit task except that rather than associating a fixed probability of reward to each discrete action, an action will yield reward only when its continuous parameters are chosen within a Gaussian distribution around the current optimal action parameter $\mu^\star$ with variance $\sigma^\star$ (which are unknown to the robot). Every n timesteps, $\mu^\star$ changes so that the task is non-stationary and requires constant re-exploration and learning by the robot.

Previous researches on human-robot interaction have shown that the human engagement can be a critical aspect of the quality of the interaction [19]. Nevertheless, during interaction tasks the actions performed by a robot can have delayed effects on the human's behavior and on his engagement. To mimic this, we chose the reward to be given by a dynamical system which is based on the virtual engagement $e(t)$ of the human in the task. This engagement is supposed to represent the attention that the human pays to the robot. It starts at 5, increases up to a maximum $e_{max} = 10$ when the robot performs the appropriate actions with the appropriate parameters, and decreases down to a minimum $e_{min} = 0$ otherwise:

$$e_{t+1} = \begin{cases} e_t + \eta_1(e_{max} - e_t)H(\theta^a_t), & \text{if } a_t = a^\star \ \&\ H(\theta^a_t) \geq 0 \\ e_t - \eta_2(e_{min} - e_t)H(\theta^a_t), & \text{if } a_t = a^\star \ \&\ H(\theta^a_t) < 0 \\ e_t + \eta_2(e_{min} - e_t), & \text{otherwise} \end{cases}$$

where $\eta_1 = 0.1$ is the increasing rate, $\eta_2 = 0.05$ is the decreasing rate, and $\mathcal{H}(x)$ is the reengagement function given by $\mathcal{H}(x) = 2\left(exp\left(-\frac{(x-\mu^\star)^2}{2\sigma^{\star 2}}\right) - 0.5\right)$ where $a^\star$, $\mu^\star$ and $\sigma^\star$ are respectively the optimal action, action parameter and variance around $a^\star$.

The reward function is then computed as $r(t+1) = (1 - \lambda)e(t+1) + \lambda\Delta e(t+1)$ where $\lambda = 0.7$ is a weight. This reward function ensures that the algorithm gets rewarded in cases where the engagement $e(t+1)$ is low but nevertheless has just been increased by the action tuple $(a(t), \theta^a(t))$ performed by the robot.

We first simulated the algorithm without active exploration (thus with a fixed $\sigma = 20$) in a task where the optimal action tuple ($a^\star$, $\mu^\star$) is $(a_6, -20)$ during 200 timesteps ($\sigma^\star = 10$ in

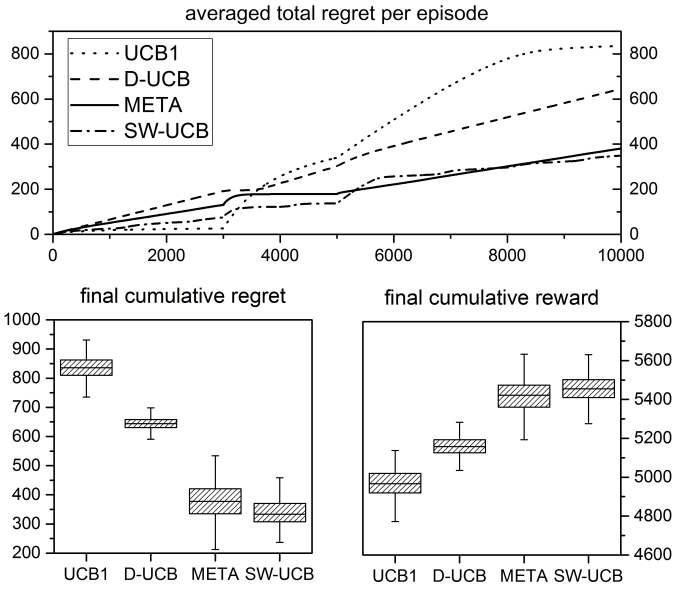

Fig. 2. Performance in the bandit task. Top: The averaged cumulative regret per episode. Bottom left: The final cumulative regret for 500 sessions. Bottom right: The final cumulative reward for 500 sessions

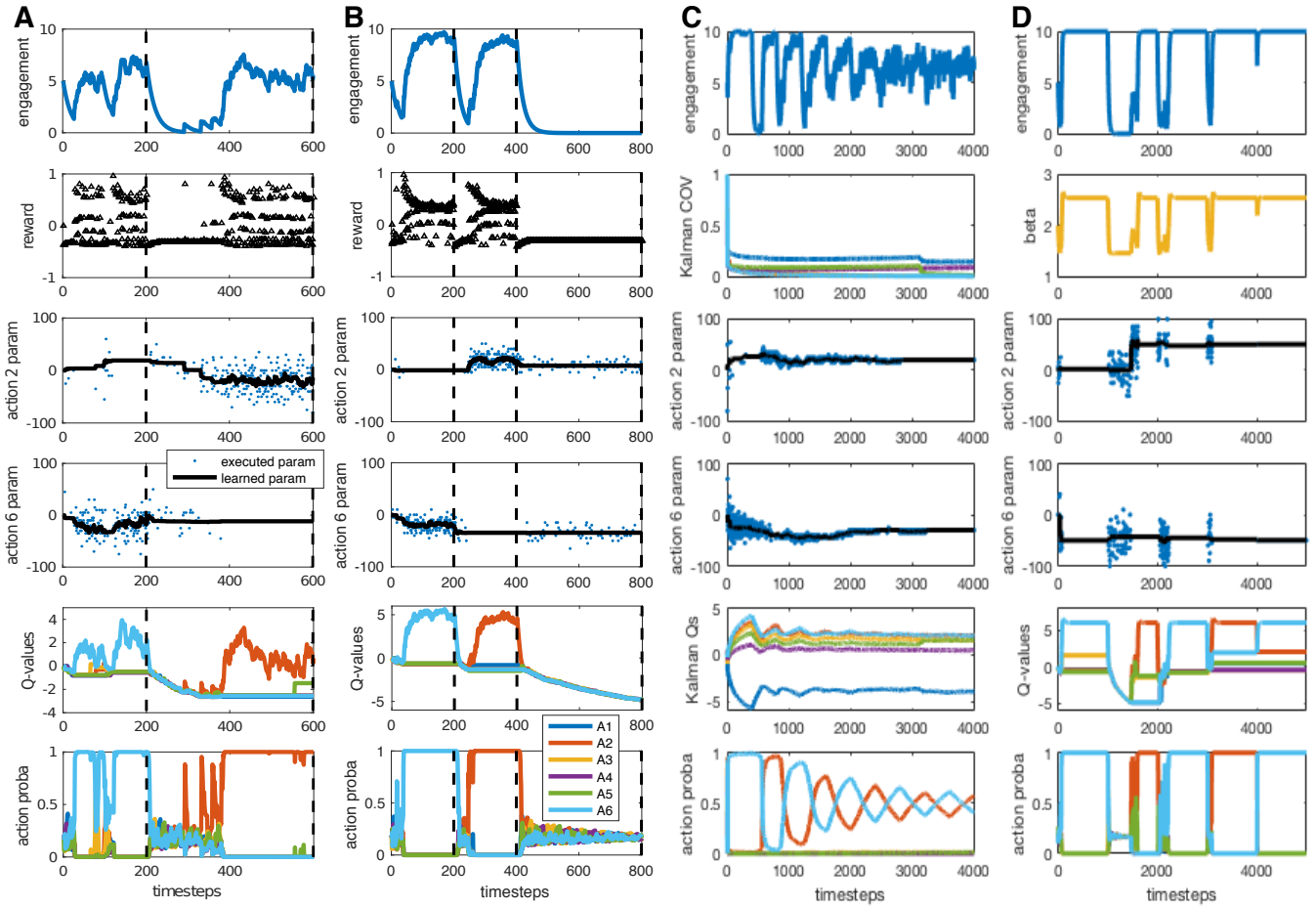

Fig. 3. Simulations of the parameterized reinforcement learning (RL) algorithm with (A) fixed $\sigma = 20$ and $\beta = 4$ (no active exploration), (B) fixed $\sigma = 10$ and $\beta = 4$ (no active exploration), (C) σ_t^a and b_t^a tuned by Kalman-RL (active exploration) or (D) σ_t and β_t tuned by meta-learning (active exploration).

all the experiments presented here), then switches to $(a_2, -20)$ until timestep 600. Figure 3A shows that the algorithm first learns the appropriate action tuple $(a_6, -20)$, then takes some time to learn the second tuple, making the engagement drop between timesteps 200 and 400 and eventually finds the second optimal tuple. Nevertheless, $\sigma = 20$ makes the robot select action parameters $\widetilde{\theta}_t^a$ with a large variance (illustrated by the clouds of blue dots around the learned action parameters θ_t^2 and θ_t^6 plotted as black curves). As a consequence, the engagement is not optimized and always remains below 7.5. In contrast, the same algorithm with a smaller fixed variance $\sigma = 10$ can make the engagement reach the optimum of 10 when the optimal action tuple is learned (Figure 3B before timestep 400), but results in too little exploration which prevents the robot from finding a new action parameter which is too far away from the previously learned one (after timestep 400, the new optimal action tuple is $(a_6, 20)$). These two examples illustrate the need to actively vary the variance σ_t as a function of changes in the robot's performance.

We next tested active exploration with the Kalman Q-Learning algorithm in a task alternating between optimal tuples $(a_2, -20)$ and $(a_6, 20)$ every 400 timesteps. Since the original purely exploratory Kalman-QL agent proposed in [11] did not manage to get an average engagement higher than 1.5 out of 10 in this task – due to the non-stationarity of the environment –, in the following we only present detailed results for the Kalman-QL with *exploration bonuses*. Figure 3C shows the results of this extended version of Kalman-QL. The diagonal terms of the covariance matrix COV in the Kalman filter nearly monotonically decrease, resulting in a large variance σ_t when action a_6 is executed until about timestep 600, and progressively decreasing the variance until the end of the experiment. Nevertheless, the algorithm quickly finds the appropriate action parameters and rapidly shifts between actions a_2 and a_6 after each change in the task condition. In the long run, the model progressively averages the statistics of the two conditions and learns to perform both actions with 50/50 probabilities (bottom part of Figure 3C) which decreases the simulated engagement (top).

We then tested active exploration with the meta-learning algorithm in a slightly more difficult task where the optimal action tuple alternate between $(a_2, -50)$ and $(a_6, 50)$ every 1000 timesteps (Figure 3D). Transient drops in the engagement result in transient decreases in the exploration parameter

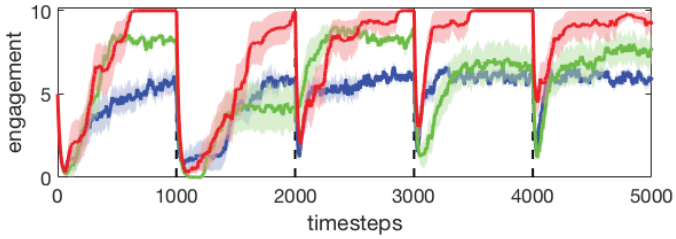

Fig. 4. Comparison of engagement in 10 simulations of the meta-learning model (red), the model without active exploration (blue), and the Kalman-QL (green).

β_t as well as transient increases in the variance σ_t. This enables the algorithm to go through quick transient but wide exploration phases and to rapidly reconverge to exploitation, thus maximizing the simulated engagement.

Finally, we performed 10 simulations of each model on the difficult version of the task described in the previous paragraph and plotted the average and standard deviation of the simulated engagement (Figure 4). The blue curve shows the performance of the algorithm without active exploration (i.e. fixed $\sigma = 19$ obtained through parameter optimization), which adapts to each new condition but never exceeds a plateau of about 6. The green curve shows the active exploration with Kalman, which adapts faster at the beginning but progressively decreases its maximal engagement. The red curve shows the active exploration with meta-learning which initially takes more time to adapt but then only performs short transient explorations and reaches the optimum engagement of 10.

C. Realistic V-REP HRI simulation

In order to have a more realistic demonstration of the proposed algorithm and to gain a better insight of its envisaged application to HRI tasks, we created and visualized a scenario using the V-REP robot simulator. In the considered scenario, a small humanoid robot, in this case a NAO, interacts with a human subject, where the envisaged goal is to collaboratively perform a task involving pointing at, picking up and placing objects in the scene in order to build a puzzle. Such a collaborative HRI scenario is in line with the objectives defined in the frames of the EU-funded project BabyRobot (H2020-ICT-24-2015-6878310), where a set of child-robot interaction use-cases will be designed and implemented to study the development of specific socio-affective, communication and collaboration skills in children. In particular, the task considered in this first simulated scenario comprises a set of 6 objects (cubes) set in front of the human and the robot (Figure 5). Each robot action at the current implementation stage corresponds to a *pointing gesture* of the robot (with its right arm) towards one of the 6 cubes. The human engagement is expressed through the gazing direction with respect to the pointed cube. In essence this means that, if the engagement is high, the attention of the human subject is directed towards the pointed cube, while if the engagement is low, the human turns his head around.

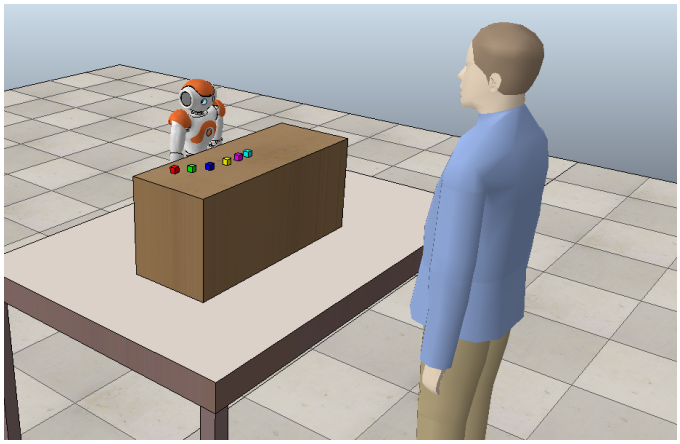

Fig. 5. V-REP simulations of the NAO robot interacting with a human.

In the current implementation, the human gaze is sampled from a normal distribution centered around the position of the object corresponding to the action (pointing gesture) currently performed by the robot, with a standard deviation that depends (inversely proportionally) on the current human engagement value. Changes of human gaze direction are sampled and executed every T_1 time-steps, while each robot action is performed and remains unchanged for T_2 time-steps ($T_2 = nT_1$); meaning that the robot is assumed to collect n observations of human gaze direction changes before selecting and executing a new action. In the current simulation scenario, a simple case is considered, where the actual simulated human engagement value is assumed to be directly known to the robot. In future work, more realistic scenarios will be implemented and tested in simulation, where the actual human engagement value will be assumed to be unknown to the robot and estimated on-line by means of a visual perception module, based on observations of the human gaze directions. Furthermore, the action parameter will also be integrated in the task and will represent a measure of the overall "intensity" of the robot's arm movements when executing a communicative action (e.g. a pointing gesture).

It is interesting in this scenario to study and visualise the performance of the proposed meta-learning active exploration algorithm when the optimal action parameter changes (while the optimal action itself remains the same). Figure 6 compares the performance of the proposed meta-learning algorithm as compared to the Kalman Q-learning mechanism, when the optimal action parameter undergoes a 50% change (from a value of -50 to -25). We can see that the meta-learning algorithm adapts much faster to the new task parameter. Specifically, the human engagement drops to no less than 70% of the maximum engagement and recovers to 85% after a few trials (in this case, after approximately 20 trials). In addition, the action parameter converges fast to the optimal value (in this example, after 26 trials). On the contrary, the Kalman Q-learning algorithm fails to adapt to the new task parameter and to raise the engagement back to its maximum value, resulting

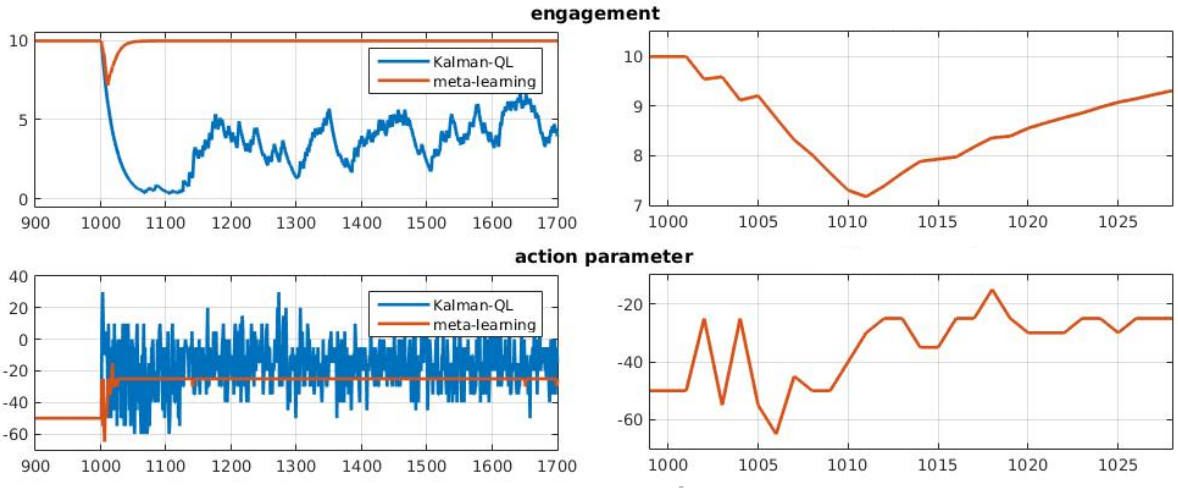

Fig. 6. Left column: Comparison of engagement (top) and action parameter (bottom) convergence between the meta-learning and the Kalman-QL algorithms. Right Column: zoomed-in plots depicting convergence performance of the proposed meta-learning algorithm.

in a sub-optimal engagement for the rest of the experiment. This behavior is also illustrated by the oscillation of the action parameter as it fails to converge to the optimal value.

Figure 7 shows indicative snapshots of the V-REP simulated HRI scenario, showing human-centric (left) and robot-centric (right) views of the scene. In the top views the human engagement is high and the human gaze direction is focused on the current robot action (pointing at the cube of cyan color). In the bottom views, the human engagement is low and the human gaze is oriented towards directions that do not focus on the current robot action (blue cube). These initial simulations provide a first understanding of practical considerations that will have to be addressed towards the implementation and deployment of more realistic HRI scenarios as already described. Initial results are promising showing the potential of the proposed meta-learning algorithm as a scheme to efficiently adapt to non-stationary conditions in challenging HRI scenarios.

IV. Discussion

In this work, we have shown that a meta-learning algorithm based on online variations of reward running averages can be used to adaptively tune two exploration parameters simultaneously used to select between both discrete actions and continuous action parameters in a parameterized action space.

We first compared the proposed algorithm with standard bandit methods in the non-stationary (switching) multi-armed bandit task proposed by [10]. We showed that it reaches a performance which is not different from one of the state-of-the-art bandit methods, namely SW-UCB. Interestingly, SW-UCB does not adapt well to some other non-stationary tasks [20]. Moreover, bandit methods work specifically in single-state tasks. The meta-learning algorithm proposed here seems promising in that it is generalized to continuous actions and multi-step tasks. In future work, we will compare it with bandit methods in a variety of non-stationary tasks and then study its performance in multi-state tasks.

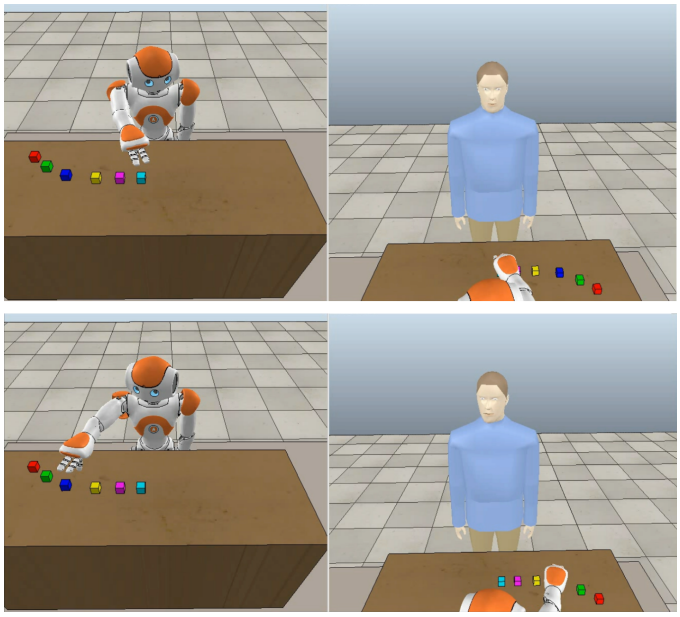

Fig. 7. Snapshots of the V-REP simulated HRI scenario, showing human-centric (left) and robot-centric (right) views of the scene. Top views: human engagement is high. Bottom views: human engagement is low.

We then applied the proposed meta-learning algorithm to a simple simulated human-robot interaction task, first in matlab simulations alone and then controlling the continuous physical simulator V-REP. We found that it outperforms continuous parameterized RL both without active exploration and with active exploration based on uncertainty variations measured by a Kalman-Q-learning algorithm. While we had previously successfully used the Kalman Q-Learning proposed by [11] to coordinate model-based and model-free reinforcement learning in a stationary task [21], it was not appropriate for the current non-stationary task. In future work, we will test the algorithm

in more complex simulated interaction tasks before applying it to real human-robot interaction.

The different results presented in this paper suggest that the proposed active exploration scheme could be a promising solution for Robotics applications of parameterized reinforcement learning, especially in non-stationary settings involving interaction with dynamic environments.

ACKNOWLEDGMENT

We would like to thank Kenji Doya, Benoît Girard, Olivier Pietquin, Bilal Piot, Inaki Rano, Olivier Sigaud and Guillaume Viejo for useful discussions. This research work has been partially supported by the EU-funded Project BabyRobot (H2020-ICT-24-2015, grant agreement no. 687831) (MK, CT), by the Agence Nationale de la Recherche (ANR-12-CORD-0030 Roboergosum Project and ANR-11-IDEX-0004-02 Sorbonne-Universités SU-15-R-PERSU-14 Robot Parallearning Project) (MK), and by Labex SMART (ANR-11- LABX-65 Online Budgeted Learning Project) (MK).

REFERENCES

[1] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Machine Learning*, vol. 84, pp. 171–203, 2011.

[2] F. Stulp and O. Sigaud, "Robot skill learning: From reinforcement learning to evolution strategies," *Paladyn Journal of Behavioral Robotics*, vol. 4, no. 1, pp. 49–61, 2013.

[3] J. Kober, J. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, pp. 1238–1274, 2013.

[4] M. Khamassi, B. Girard, A. Clodic, S. Devin, E. Renaudo, E. Pacherie, R. Alami, and R. Chatila, "Integraton of action, joint action and learning in robot cognitive architectures," *Intellectica*, vol. 2016/1, pp. 169–203, 2016.

[5] E. Renaudo, B. Girard, S. Devin, R. Alami, A. Clodic, R. Chatila, and M. Khamassi, "Can robots learn behavioral habits? coordination of model-based and model-free reinforcement learning in a robot neuro-inspired cognitive architecture," *Submitted to Frontiers in Neurorobotics*, 2016.

[6] W. Masson and G. Konidaris, "Reinforcement learning with parameterized actions," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016.

[7] M. Hausknecht and P. Stone, "Deep reinforcement learning in parameterized action space," in *International Conference on Learning Representations (ICLR 2016)*, 2016.

[8] H. van Hasselt and M. Wiering, "Reinforcement learning in continuous action spaces," in *IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 2007, pp. 272–279.

[9] N. Schweighofer and K. Doya, "Meta-learning in reinforcement learning." *Neural Networks*, vol. 16, no. 1, pp. 5–9, 2003.

[10] A. Garivier and E. Moulines, "On upper-confidence bound policies for non-stationary bandit problems," *arXiv preprint arXiv:0805.3415*, 2008.

[11] M. Geist and O. Pietquin, "Kalman temporal differences," *Journal of artificial intelligence research*, vol. 39, pp. 483–532, 2010.

[12] C. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[13] K. Caluwaerts, M. Staffa, N. S., C. Grand, L. Dollé, A. Favre-Félix, B. Girard, and M. Khamassi, "A biologically inspired meta-control navigation system for the psikharpax rat robot," *Bioinspiration and Biomimetics*, vol. 7, no. 2, p. 025009, 2012.

[14] M. Khamassi, P. Enel, P. Dominey, and E. Procyk, "Medial prefrontal cortex and the adaptive regulation of reinforcement learning parameters," *Progress in Brain Research*, vol. 202, pp. 441–464, 2013.

[15] S. Palminteri, M. Khamassi, M. Joffily, and G. Coricelli, "Medial prefrontal cortex and the adaptive regulation of reinforcement learning parameters," *Nature Communications*, vol. 6, p. 8096, 2015.

[16] N. D. Daw, J. P. O'Doherty, P. Dayan, B. Seymour, and R. J. Dolan, "Cortical substrates for exploratory decisions in humans," *Nature*, vol. 441, no. 7095, pp. 876–879, 2006.

[17] L. Kocsis and C. Szepesvári, "Discounted ucb," in *2nd PASCAL Challenges Workshop*, 2006, pp. 784–791.

[18] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[19] S. M. Anzalone, S. Boucenna, S. Ivaldi, and M. Chetouani, "Evaluating the engagement with social robots," *International Journal of Social Robotics*, vol. 7, no. 4, pp. 465–478, 2015.

[20] R. Allesiardo and R. Fraud, "Exp3 with drift detection for the switching bandit problem," in *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, Oct 2015, pp. 1–7.

[21] G. Viejo, M. Khamassi, A. Brovelli, and B. Girard, "Modeling choice and reaction time during arbitrary visuomotor learning through the coordination of adaptive working memory and reinforcement learning," *Frontiers in behavioral neuroscience*, vol. 9, 2015.