Online adaptation to human engagement perturbations in simulated human-robot interaction using hybrid reinforcement learning

Theodore Tsitsimis*, George Velentzas*, Mehdi Khamassi†*, Costas Tzafestas*
*School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece
†Institute of Intelligent Systems and Robotics, Sorbonne Universités, UPMC Univ Paris 06, CNRS F-75005 Paris, France
Email: {thodotsi, geovelentzas}@gmail.com, mehdi.khamassi@upmc.fr, ktzaf@cs.ntua.gr

***Abstract*—Dynamic uncontrolled human-robot interaction requires robots to be able to adapt to changes in the human's behavior and intentions. Among relevant signals, non-verbal cues such as the human's gaze can provide the robot with important information about the human's current engagement in the task, and whether the robot should continue its current behavior or not. In a previous work [1] we proposed an active exploration algorithm for reinforcement learning where the reward function is the weighted sum of the human's current engagement and variations of this engagement (so that a low but increasing engagement is rewarding). We used a structured (parameterized) continuous action space where a meta-learning algorithm is applied to simultaneously tune the exploration in discrete and continuous action space, enabling the robot to learn which discrete action is expected by the human (e.g. moving an object) and with which velocity of movement. In this paper we want to show the performance of the algorithm to a simulated human-robot interaction task where a practical approach is followed to estimate human engagement through visual cues of the head pose. We then measure the adaptation of the algorithm to engagement perturbations simulated as changes in the optimal action parameter and we quantify its performance for variations in perturbation duration and measurement noise.**

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 [2], [3]. As a recent review for reinforcement learning applications to Robotics highlights [4], many algorithms have been developed for different tasks. For stationary environments, human prior knowledge has also been used in order to determine the balance between exploration and exploitation, but such an approach denotes weak performance for non-stationary environments.

For continuous action spaces, there has been important contribution with promising real world applications in the field of Robotics. Very recently though, the combination of discrete actions $A_d = \{a_1, a_2, ..., a_k\}$ where each action $a \in A_d$ is described by a set of m_a continuous parameters $\{\theta_1^a, ..., \theta_{m_a}^a\} \in \mathbb{R}^{m_a}$ was proposed for RL algorithms in structured Parameterized Action Space Markov Dedision Processes (PAMDP) [5], [6]. This approach was applied to Robocup 2D soccer simulations, where an agent learned the optimal action (kick the ball, run, turn, etc) as also the optimal parameter value of each action (power, speed, angle, etc). Similar work has been done in [6], where the optimal action and parameters are learned in parallel, instead of altering between learning phases, in order to ensure convergence. However, these methods use a fixed exploration-exploitation trade-off which results in issues discussed in [4].

Exploration in parameterized action space described in [6] uses ϵ-greedy exploration by picking a random discrete action $a \in A_d$ with probability ϵ and then sample the action's parameters θ_i^a out of a uniform distribution. The value of ϵ is decreased over time steps, but the decrease rate is set by hand, thus requiring human prior knowledge. In [7] a Gaussian distribution is used instead, but with a fixed value σ. In our work, we use a hybrid algorithm in the sense that we combine different learning processes in parallel that rely on different types of representations: discrete action values at the highest level, and continuous action parameters at the lowest level. We apply the meta-learning algorithm proposed in [8], and use short-term and long-term reward running averages to both adaptively tune the inverse temperature parameter β of Boltzmann softmax exploration function for action selection, as well as the width of the Gaussian distribution from which each action parameter is sampled around its current value. We applied our proposed algorithm to a simple simulated human-robot interaction task where the objective was to maximize human engagement (modeled with a function unknown to the robot). The results we obtained outperformed continuous parameterized RL both without active exploration and with active exploration when using Kalman RL-algorithm [9] based on uncertainty variations.

In [1] we considered human engagement to be known to the robot for the calculation of the reward. However, in real human-robot interactions the engagement is not directly accessible. To this purpose, social signals expressed through non-verbal gestures and gaze behaviors have to be used in order to automatically evaluate the engagement, as described in [11]. Here, we consider the human engagement to be unknown and further assume that an estimation process is in place based on visual features measured during human-robot interactions. In order to have more realistic results we show how engagement perturbations, modeled as temporary changes of the optimal action parameter, affect the learning process. We also examine how the algorithm behaves in the presence of noise that

introduces uncertainties in the engagement estimation process. The goal is to conduct an initial evaluation as to how scalable and generalizable the proposed learning algorithms are in more close to real-life scenarios and how the presence of an uncertainty on human engagement estimation may affect the performance of the system.

II. Active exploration algorithm

This section briefly describes the mathematical formulation underlying the proposed active exploration method that is extensively analyzed in [1]. The meta-learning algorithm is summarised in Algorithm 1. It first employs Q-Learning [10] to learn the value of a 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 $\widetilde{\theta}^{a_j}_{i,t}$ are selected from a Gaussian exploration function centered on the current values $\theta^{a_j}_{i,t}(s_t)$ in state s_t of the parameters of this action [7]:

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

where the width σ_t of the Gaussian is a meta-parameter which will be tuned through meta-learning (see below) and action parameters $\theta^a_{i,t}(s_t)$ are learned with a continuous actor-critic algorithm [7]. A reward prediction error $\delta_t = r_t + \gamma V_t(s_{t+1}) - V_t(s_t)$ is computed from the critic and is used to update the parameter vectors ω^C_t and ω^A_t of the neural network function approximations in the critic and the actor:

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

$$\omega^A_{i,t+1} = \omega^A_{i,t} + \alpha_A \delta_t (\widetilde{\theta}^a_{i,t} - \theta^a_{i,t}(s_t)) \frac{\delta \theta^a_{i,t}(s_t)}{\delta \omega^A_{i,t}} \tag{6}$$

where α_C and α_A are learning rates and $V_t(s_t)$ is the output of the function approximation at time t with state s_t as input. 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. Thus here, following the proposition of [8], 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 [8] 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:

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

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.

Algorithm 1 Active exploration with meta-learning

1: Initialize $\omega^A_{i,0}$, $\omega^C_{i,0}$, $Q_{i,0}$, β_0 and σ_0
2: **for** $t = 0, 1, 2, ...$ **do**
3: Select discrete action a_t with $softmax(s_t, \beta_t)$ (Eq. 3)
4: Select action parameters $\widetilde{\theta}^a_{i,t}$ with $GaussianExploration(s_t, a_t, \theta^a_{i,t}, \sigma_t)$ (Eq. 4)
5: Observe new state and reward $\{s_{t+1}, r_{t+1}\} \leftarrow Transition(s_t, a_t, \widetilde{\theta}^a_{i,t})$
6: Update $Q_{t+1}(s_t, a_t)$ in the discrete Q-Learning (Eq. 2)
7: Update function approx. $\omega^C_{i,t+1}$ and $\omega^A_{i,t+1}$ in continuous actor-critic (Eq. 5, Eq. 6)
8: **if** meta-learning **then**
9: Update reward running averages $\bar{r}_t$ and $\bar{\bar{r}}_t$ (Eq. 7)
10: Update β_{t+1} and σ_{t+1} (Eq. 8)
11: **end if**
12: **end for**

III. Experiments

A. Simple HRI simulation

We tested 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. Each action corresponds to a pointing gesture of the robot to one of the 6 objects in front of it and the action parameter represents the movement intensity. 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 $\sigma^\star$. Every n timesteps, $\mu^\star$ changes so that the task is non-stationary and requires constant re-exploration and learning by the robot.

Since during interaction tasks the actions performed by a robot can have delayed effects on the human's behavior and engagement [11], we chose the reward to be given by

a dynamical system. This is based on the virtual engagement $e(t)$ of the human in the task which represents the attention that the human pays to the robot.

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

where η_1 is the increasing rate, η_2 is the decreasing rate, $\mathcal{H}(x) = 2\left(exp\left(-\frac{(x-\mu^\star)^2}{2\sigma^{\star 2}}\right) - 0.5\right)$ is the reengagement function and $a^\star$, $\mu^\star$ and $\sigma^\star$ are respectively the optimal action, action parameter and variance around $a^\star$. We modeled this attention to be between 0 (no attention) to 10 (maximum attention) and the reward function was computed as $r(t+1) = (1-\lambda)e(t+1) + \lambda\Delta e(t+1)$ where $\lambda = 0.7$ is a weight. This formulation has great meaning, since the reward is modeled by using both the engagements current value as also its rate of change.

We run the simulation and compared the results for different models: active exploration with meta-learning, without active exploration (we used fixed values for σ) and active exploration with Kalman Q-Learning algorithm. These were tested in a task where the optimal action tuple $(a^\star, \mu^\star)$ alternated between $(a_2, -50)$ and $(a_6, 50)$ every 1000 time steps. We repeated this task running 10 simulations for each algorithm, and monitored the average and standard deviation of the engagement (Fig.1 top). The algorithm without active exploration never exceeded an engagement over 6 for every interval and the Kalman Q-learning adapted fast but progressively decreased its achieved engagement. Our active exploration with meta-learning algorithm achieved the best performance, reaching high values of engagement (optimum at some intervals). More particularly, when the engagement dropped, the inverse temperature parameter β_t also dropped and the action parameter variance σ_t of the gaussian exploration function increased. This resulted on re-engaging exploration when needed, without loosing the exploitation convergence on large stationary intervals.

We also created a virtual simulation environment using V-REP robot simulator, in order to have a more realistic representation before any application to the real world (Fig.1 bottom). In this simulation environment, we considered a preliminary scenario where 6 cubes were present in front of the robot. We assumed that the optimal action at this state was the pointing gesture and the task was to find the optimal action intensity which maximized the human attention to the pointed cube. In this preliminary work, the engagement value was returned to the robot at every timestep, however more realistic scenarios are also being considered and tested in simulation, as described in the following subsection.

B. Engagement Estimation Process

In order to make the simulated HRI scenario more realistic and obtain a more reliable assessment on the applicability of the developed learning algorithms in real use-case scenarios, we consider an estimation process that evaluates human engagement through visually extracted metrics.

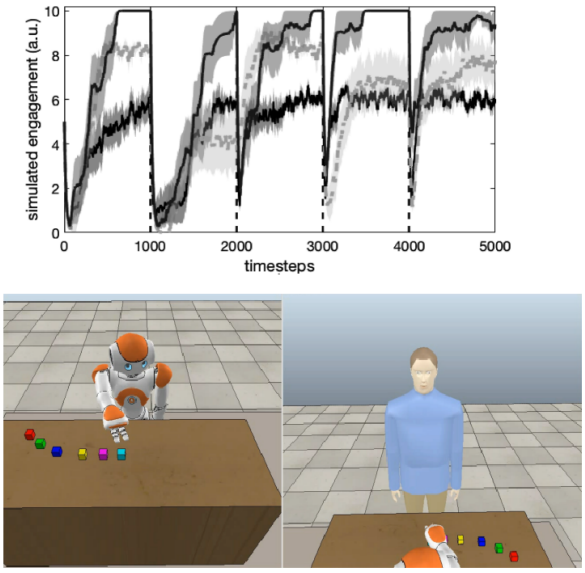

Fig. 1. Top: Comparison of engagement in 10 simulations of the meta-learning model (dark gray), the model without active exploration (black), and the Kalman-QL (light gray). Bottom: V-REP simulations of the NAO robot interacting with a human.

As mentioned in [11], [12], head pose data is proved to be highly correlated with human engagement. In the current implementation of the simulated HRI scenario, the human head pose is generated by sampling a normal distribution centered around the position of the object corresponding to the action (i.e. pointing gesture) currently performed by the robot. The standard deviation of this probability distribution is assumed to be proportional to the absolute difference between the action parameter executed by the robot and its optimal value. Thus, when the action parameter deviates from its optimal value, the human engagement drops and the head pose variance increases, meaning that the human is disengaged from the task and essentially starts looking around. Accordingly, when the action parameter is near to its optimal value, the head pose variance decreases meaning that the human pays attention to the action performed by the robot.

The engagement estimation is achieved by measuring the mean standard deviation (MSD) of the human's head yaw angle with respect to the cube pointed by the robot in a specified time window. In particular, at each trial the robot collects and processes n observations of the human head pose before selecting and executing a new action. The head pose measurement error is taken into account and modeled as an additive Gaussian noise with standard deviation σ that depends on the accuracy of the visual head pose estimation. It is thus evident that the higher the head pose MSD, the lower the human engagement. Our estimator evaluates the current human engagement based on this MSD value and provides it to the robot as a reward. The reward function now considers the estimated engagement $\hat{e}$ and is computed as $r(t+1) = (1-\lambda)\hat{e}(t+1) + \lambda\Delta\hat{e}(t+1)$.

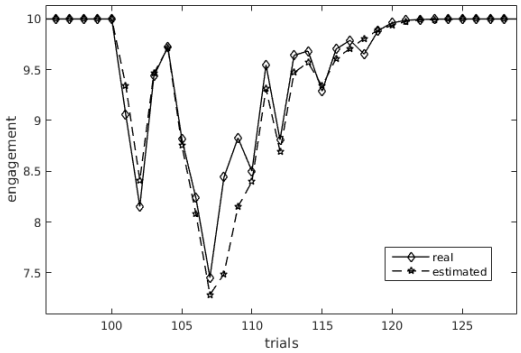

Fig. 2. Estimated vs. real (simulated) engagement in one run involving a step change (of 100%) after 100 trials, based on $n = 5$ head pose observations per trial and Gaussian measurement noise with $\sigma = 1$.

The first series of numerical experiments that we conducted involves step changes in the optimal (continuous) action parameter performed every 100 trials. Figure 2 depicts the results of a single run for the estimated vs. the real (simulated) engagement, when the robot is assumed to collect $n = 5$ head pose observations per trial (for the human engagement estimation process) with a Gaussian measurement noise of $\sigma = 1$. In this run the action parameter undergoes a step change of 100% after 100 trials. It is found that in such a situation the real (simulated) engagement does not drop below a value of 7 (i.e. 70% from the optimal engagement) and consistently converges rapidly to a value above 90% after approximately $10 - 15$ trials. A series of 50 runs for the same step-change scenario has also been conducted and the results are shown in Figure 3. It is assumed that the optimal action parameter undergoes a change of 100% (i.e. doubles from a value of 5 to a value of 10) and after 100 trials goes back to its initial value. Figure 3 (bottom) shows the actual executed action parameter (mean and variance after 50 runs). These results show that although the optimal action parameter suddenly doubled, the adaptation was fast enough to keep the engagement above a 70% value and consistently make it converge to a value above 90% after approximately 15 trials (Figure 3, top).

During a human-robot interaction task, it is natural for a person and much more for a child to be distracted by an external event (loud noise, presence of other people, etc). We simulate such a perturbation as an abrupt and short in time (impulse-type) change of the optimal action parameter. The behavior of the algorithm (mean and variance of 50 sample runs) is depicted in Figure 4 for various durations of the perturbation impulse. Here, the optimal parameter has a value of 5 which is changed to 10 during the perturbation. We observe that when the perturbation lasts for only 1 trial, the executed action parameter is almost unaffected and the human engagement does not drop lower than a value of 9.

In order to further quantify the performance of the learning algorithm, we calculate the mean absolute deviation (MAD) of the real (simulated) engagement and of the executed action

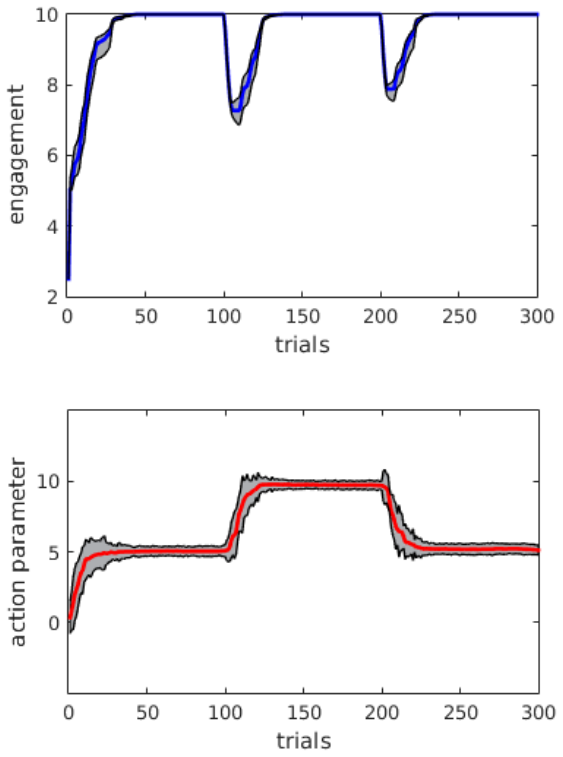

Fig. 3. Top: Real (simulated) human engagement (mean and standard deviation after 50 runs) when the optimal action parameter undergoes step changes every 100 trials as shown in left figure (engagement does not drop below a 70% value). Bottom: Executed action parameter (mean and standard deviation after 50 runs) involving step changes.

parameter from their optimal values, for perturbation durations in the range of 1 to 10 trials. The results are depicted in Figure 5, which indicates that longer perturbations lead to slower adaptation and result in larger MAD values. The same results are also numerically shown in Tables I to IV (end of the paper), including the maximum engagement deviation as well as the number of trials needed for the engagement to recover to 90% of its maximal value after the end of the perturbation. It should also be highlighted, though, that as illustrated by the obtained results, no matter how long the perturbation, the algorithm will always reconverge to the optimal value.

In a similar way, Figure 6 shows the engagement and action parameter deviation for an increasing value of σ representing the head pose measurement noise. In this particular numerical experiment, the perturbation duration is assumed to last for a single trial. It is clear from this figure that larger σ values (i.e. amplitude of noise in the head pose data and consequently in the human engagement estimation process) lead to larger deviations for the engagement and for the action parameter from their optimal values. However, it is also apparent that there is a range of values in σ (corresponding to a range of uncertainty in the human engagement estimation process) that results in a quite robust system performance (numerically, in this case, up to a value of $\sigma = 2$). Evaluating the robustness of

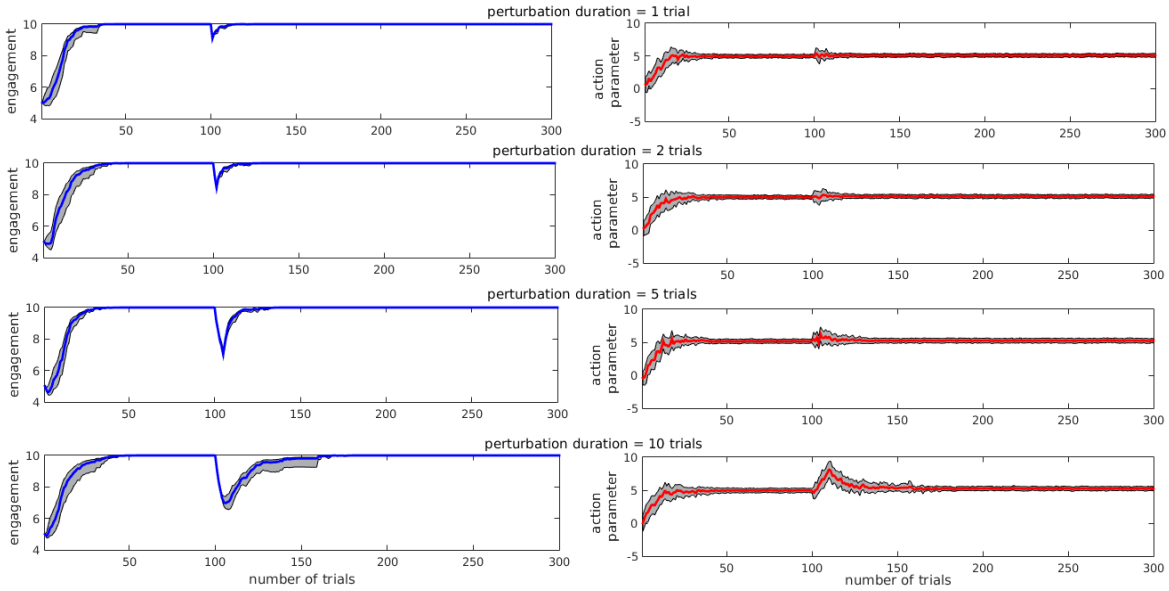

Fig. 4. Action parameter perturbations with durations of 1, 2, 5 and 10 trials, respectively.

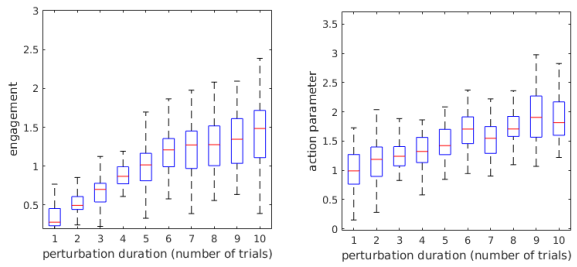

Fig. 5. Performance for increasing perturbation duration. Left: Engagement mean absolute deviation from its maximum value (10). Right: Action parameter mean absolute deviation from its optimal value (5). The measurement noise has a $\sigma = 1$.

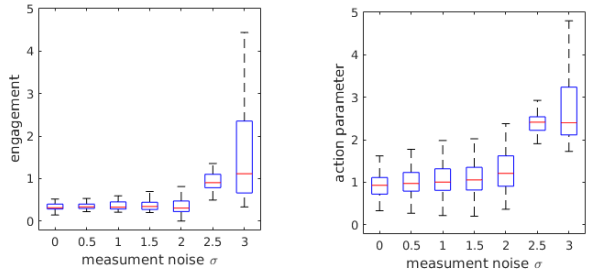

Fig. 6. Performance for increasing noise σ. Left: Engagement mean absolute deviation from its maximum value (10). Right: Action parameter mean absolute deviation from its optimal value (5). The measurement noise has $\sigma = 1$.

the learning mechanisms under assumed measurement noise and proposing counter-measures in the presence of large estimation uncertainties is a work-in-progress and will be further explored in future work.

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 then applied the proposed meta-learning algorithm to a simple simulated human-robot interaction task and 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. The robustness of the algorithm was tested in situations where the human is distracted by external events and we showed that no matter the length of the perturbation, the algorithm would always come back to optimal behavior afterwards. In fact, the algorithm succeeded to keep human engagement above 90% of its optimal value when engagement perturbations were short.

Then, we showed how engagement is affected by the presence of measurement noise during engagement estimation. Although the algorithm is not significantly affected by small noise amplitudes, it fails when uncertainties in human engagement are high. To improve this, the robot could reset its engagement estimation when the human looks at a discrete object whose location is known to the robot. The robot could even ask the human to look at the object in order to recalibrate its estimation. In future work, we will address these issues and 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 in combination with the described engagement estimation process could be a promising solution for applications related to human-robot interaction tasks in dynamic environments.

Engagement MAD values

Perturbation duration	Mean	STD	25% percentile	75% percentile
1	0.34939	0.17679	0.23192	0.45413
2	0.52831	0.14063	0.44253	0.60892
3	0.68629	0.22575	0.53915	0.77973
4	0.89065	0.17484	0.77258	0.99155
5	1.0118	0.31078	0.81102	1.1657
6	1.2232	0.34815	0.99221	1.3546
7	1.2957	0.61759	0.96951	1.4515
8	1.249	0.38641	1.0064	1.5177
9	1.3374	0.39675	1.036	1.6112
10	1.4415	0.44524	1.1093	1.7176

TABLE I
ENGAGEMENT MAD VALUES

Action parameter MAD values

Perturbation duration	Mean	STD	25% percentile	75% percentile
1	1.0337	0.42043	0.7643	1.2693
2	1.1584	0.35795	0.89604	1.3987
3	1.2774	0.31868	1.0767	1.4055
4	1.3416	0.29543	1.1319	1.5626
5	1.4847	0.30732	1.2685	1.7003
6	1.6898	0.35503	1.4562	1.9139
7	1.5712	0.44293	1.2896	1.7477
8	1.7163	0.3111	1.5788	1.9216
9	1.9387	0.48988	1.5668	2.2712
10	1.8471	0.37307	1.5986	2.1704

TABLE II
ACTION PARAMETER MAD VALUES

Engagement max deviation

Perturbation duration	Mean	STD	25% percentile	75% percentile
1	1.1558	0.48699	0.88976	1.3209
2	1.7664	0.36121	1.6266	1.8775
3	2.2081	0.56905	1.9838	2.6328
4	2.8998	0.45312	2.5507	3.2255
5	3.2186	0.75891	2.7269	3.7832
6	3.8114	0.72485	3.3217	4.3216
7	4.0132	1.1746	3.5341	4.6
8	4.0311	0.98245	3.3356	4.8395
9	4.2334	1.1234	3.3556	5.0483
10	4.5097	1.0771	3.8666	5.2737

TABLE III
ENGAGEMENT MAX DEVIATION

Number of trials to 90%

Perturbation duration	Mean	STD	25% percentile	75% percentile
1	1.4	2.8714	0	0
2	5.72	3.5516	3	8
3	6.2	4.6861	3	7
4	5.3	3.6936	3	6
5	7.66	4.7536	4	11
6	9	5.9074	5	11
7	9.42	6.2632	5	11
8	11.5	7.0226	7	14
9	11.6	8.1039	6	16
10	12.2	9.6637	6	16

TABLE IV
NUMBER OF TRIALS TO 90%

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] M. Khamassi, G. Velentzas, T. Tsitsimis, and C. Tzafestas, "Active exploration and parameterized reinforcement learning applied to a simulated human-robot interaction task," in *IEEE Robotic Computing 2017*, Taipei, Taiwan, 2017.

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

[3] 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.

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

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

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

[7] 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.

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

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

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

[11] 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.

[12] R. Ooko, R. Ishii, and Y.I.Nakano, "Estimating a user's conversational engagement based on head pose information," in *Intelligent Virtual Agents. IVA 2011. Lecture Notes in Computer Science, vol 6895.*, V. H.H., K. S., M. S., and T. K.R., Eds. Springer, Berlin, Heidelberg, 2011.

[13] 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.