Human in the Loop of Robot Learning: EEG-based Reward Signal for Target Identification and Reaching Task

Lucia Schiatti, Jacopo Tessadori, Nikhil Deshpande, Giacinto Barresi, Louis C. King and Leonardo S. Mattos

***Abstract*— Shared control and shared autonomy play an important role in assistive technologies, allowing the offloading of the cognitive burden required for control from the user to the intelligent robotic device. In this context, electrophysiological measures of error detection, directly measured from a person's brain activity as Error-related Potentials (ErrPs), can be exploited to provide passive adaptation of an external semi-autonomous system to the human. This concept was implemented in an online robot learning task, where user's evaluation of the robot's actions, in terms of detected ErrP, was exploited to update a reward function in a Reinforcement Learning (RL) framework. Results from both simulated and experimental studies show that the introduction of human evaluation in the robot learning loop allows for: (1) the acceleration of optimal policy learning in a target reaching task, (2) the introduction of a further degree of control in robot learning, namely identification of one among multiple targets, according to the user's will. Overall, presented results support the potential of human-robot co-adaptive and co-operative strategies to develop human-centered assistive technologies.**

I. INTRODUCTION

In the context of assistive technologies for severely motor-impaired people, the development of Brain-Computer-Interfaces (BCIs) and neuro-prostheses offered new channels to allow communication and control of external devices [1], inferring user intentions directly from the ongoing brain activity, without the need to exploit any muscular control. However, the continuous delivery of mental commands to a brain-actuated device is highly demanding in terms of cognitive attention and effort. Moreover, the identification of user intention, especially when based on non-invasive recording techniques such as Electroencephalography (EEG), is prone to errors and allows for a very restricted number of discriminable states, and consequently, degrees of control. A viable solution to these issues is offered by advances in autonomous robotics, which make it possible to exploit shared autonomy and shared control solutions [2]. Examples of such applications include semi-autonomous wheelchairs, prostheses, or robots for telepresence [3]–[5]. Studies involving disabled patients reported their preference for a semi-autonomous rather than fully autonomous approach, in which the intelligent system helps the human to cope with the problematic low-level aspects of planning, while the user keeps as much control as possible over decision-making [6].

Following this idea, a novel approach for BCI-based interaction between a human and an assistive system has been investigated during the last two decades. This is based on the detection of electrophysiological correlations of error recognition in the human brain, the so-called error-related EEG potentials (ErrPs). In this approach, systematized by [7] and shown in Fig. 1, the user monitors the performance of an autonomous system, acting like a critic instead of continuously generating control commands, and the user's spontaneous perception of an erroneous action is used to optimize the system's behavior. This concept is built on the ability of recognizing ErrPs on a single-trial basis. ErrPs are responses arising in the fronto-central area of the cortex, and have been observed during perception of erroneous actions performed by an interface [8] or an external robotic system [9], when interpreting the user's intention.

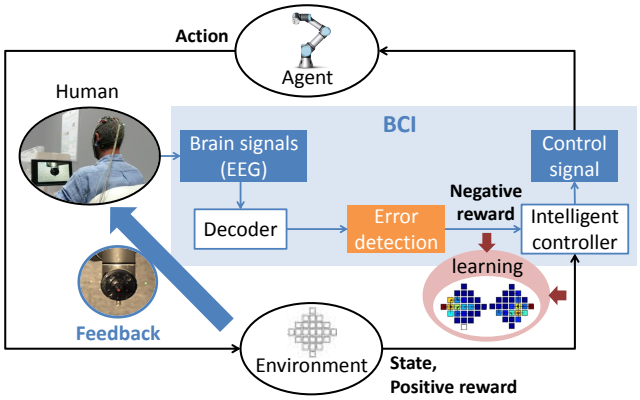

Fig. 1: System components and information flow in ErrP-based robot learning.

ErrP single-trial detection has been implemented online in BCI applications for erroneous action correction, e.g. in P300-based speller [10] or for robot movements [11], with reported classification accuracies between 70 and 80%. Some studies explored the use of ErrP signals for error-driven learning, e.g. for online re-training of a two-class motor imagery classifier [12], or, as is the focus of the present work, to improve the behavior of a semi-autonomous system. A first implementation of this concept, where ErrP is used as negative reward value in a RL scheme, was presented by [7], in a mono-dimensional control task. An online evaluation of the same approach on two subjects monitoring a simulated robot was demonstrated by [13]. In [14] a study on ErrP implementation for robot RL of optimal policy for target reaching in a 2-D discrete space was carried out. The study, conducted with 12 subjects, reported algorithm convergence towards optimal behavior, even for targets unseen during training, with a 74.3% ErrP online classification accuracy. In [15], a possible integration with shared control technique

All authors are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, 16163 Genova, Italy. Corresponding author: lucia.schiatti@iit.it

Fig. 2: Graphical User Interface (GUI) for teleoperation during: training (left), and testing phase (right).

was shown, in a 2-D cursor-reaching task. The rationale behind these works lies in the fact that ErrP detection offers a natural framework for improving the performance of an artificial intelligence system in an RL scheme (Fig. 1). Error identification through user monitoring can be exploited by the RL agent to learn an optimal behavior, by reducing the likelihood of repeating the same decision in the same context.

The aim of the present study is to improve upon the mentioned works, by exploiting the concept of the EEG-based reward for robot RL, to implement a simultaneous learning of: (1) the optimal policy for reaching a target, and (2) the identification of user-intended target location, among a pre-defined set of two positions. Our main contribution is to provide a framework for simultaneous learning of the two problems mentioned above, and to test this framework in an online setting involving physical robot movements. We validated the proposed learning algorithm in a simulation study aimed at assessing its functioning compared to a classic RL framework, where only environmental information (not related to the user's error perception) is exploited. We then tested it online in a study involving 8 subjects, during tele-observation of the movements of a robotic manipulator towards a target, in a 2-D discrete space. Since achieving an ErrP classification accuracy above chance level (50-50) is a key aspect for RL algorithm convergence, two different feedback conditions were tested. In particular, as we observed that tactile feedback can improve ErrP detection [16], these consisted of providing either visual or visuo-tactile stimulation to the subject during the robot's movements.

II. METHODS

A. Experimental Protocol

Eight subjects (6 males, 2 females, aged 27.6 $\pm$ 1.7) participated in the study. Before the experiment, all subjects agreed with the experiment guidelines, and signed an informed consent document[1]. All experimental data is available online[2]. The experiment, lasting about one hour, included a training phase aimed at collecting data to train an ErrP classifier, and a testing phase, where such classifier was exploited online to include information related to the user's evaluation of robot actions in the robot learning loop.

Training: During training, the subject was asked to observe, by means of a graphical user interface (GUI) shown on a PC screen, a robotic arm moving towards a target in a pre-programmed way, with sporadic (25%) direction errors. Movements occurred in a discrete mono-dimensional grid including 7 positions, superimposed on the robot scene (Fig. 2, left). The target position was selected to be either of the two extreme locations on this 7-position 1-D grid, and highlighted in green color on the GUI. It was randomly reset at each new iteration (the sequence of the robot's moves from the starting position, i.e. the middle of the 1-D grid, to the target). The training session included 600 trials, each one consisting of a single robot movement, i.e. movement between two adjacent positions. Time between two consecutive movements was set as 2.5 s. In order to make the ErrP classifier scalable from training (related to 1-D movements) to testing (related to 2-D movements) data, the training was divided into four sub-sessions, alternately showing horizontal and vertical 1-D grid orientation. In this way the training data included all four movement directions (left-right-up-down).

Testing: During testing, the robot was allowed to move in a 2-D diamond grid with 25 positions, as visible in Fig. 2. The grid had two possible target locations, at either extreme vertical or horizontal positions that were alternated between subjects. Testing encompassed 600 trials, i.e. robot steps. The movements of the robot were dictated by the output of the RL algorithm, explained in detail in section II-C.2. The RL algorithm took as input the result of the online ErrP classification after each robot movement, reflecting user's evaluation of a correct or erroneous robot behavior. Therefore, contrary to the training session, in the testing phase the subject's brain activity directly influenced robot movement behavior. The testing session was divided into a route learning phase, where information on the current target was provided in a supervised way, and a phase in which also autonomous target identification was active. Specifically, the first 500 trials were used to generate optimal step sequences toward each target (route learning), whereas in the last 100 trials ErrP detection was used together with the information obtained in the route-learning phase, as input to the target learning algorithm, to identify which target was currently selected and to command the robot to move towards it.

Feedback condition: Each subject tested one of two feedback conditions: the visual (V) condition or the visuo-tactile (VT) condition. In the V condition, a red cursor was superimposed on the robot position within the GUI. The cursor served the purpose of slightly anticipating the movement of the robot, i.e. it instantly moved to the next commanded position when the robot movement command was issued. In the VT feedback condition, subjects were asked to wear vibrating bands on their wrists and ankles, and received 1-s vibrations on the right wrist, left wrist, both wrists, and both ankles for movements towards right, left, up, and down direction respectively. The vibration command was delivered simultaneously with the motion of the red cursor, when the robot movement command was issued, therefore again slightly anticipating the robot motion itself.

B. Experimental Setup

The experimental setup (Fig. 1) included: a robotic arm (UR5, Universal Robots), a HD camera (Prosilica GT1910

[1]IIT ADVR TEEP01 protocol, approved by the Ethical Committee of Liguria Region on June 14th, 2016.

[2]Dataset at: http://teep-sla.eu/index.php/results/41-icra2018-dataset

GigE Vision 1080p, Allied Vision) for robot real-time video streaming and recording, a teleoperation GUI showing the robot movements on a PC screen, through a real-time Ethernet connection to the camera, and the EEG acquisition system. Two independent PCs were used in the setup, the control-side PC (GUI PC) and a robot-side PC.

During both training and testing experimental sessions, the EEG signals from the user's brain activity were recorded using 16 active g.LADYbird (g.tec) gel electrodes connected to a g.USBamp biosignal amplifier, and located at scalp positions Fz, FC3, FC1, FCz, FC2, FC4, C3, C1, Cz, C2, C4, CP3, CP1, CPz, CP2, and CP4 according to the standard International 10/20 system. Ground and reference were respectively placed on the forehead (AFz) and left ear lobe. Signals were sampled at 256 Hz. The teleoperation GUI, signal processing, and the learning algorithm were implemented in MATLAB, while EEG data acquisition occurred through a Simulink model that handled the g.USBamp amplifier.

The robotic arm was the commercially available 6 degrees-of-freedom (DOFs) Universal Robot 5 (UR5) manipulator. The robot end-effector was modified so as to be seen as a single point easily distinguishable against the background in the GUI. The robot-side PC implemented the robot position controller, by means of a custom C++ script, and the motion commands from the control-side PC were received using UDP communications over a direct LAN connection.

Four custom-made silicone rubber (ACC Silicone M230) cuffs with embedded vibration motors (Precision Microdrives 304-116) were cast in order to provide tactile stimuli during the VT feedback condition. The vibrations were synchronized with each robot movement, and commanded through serial communication over a direct USB connection with the control-side PC.

C. Algorithms

1) EEG Analysis:

Pre-processing and features extraction: Following the state-of-the art for signal processing related to ErrP detection, EEG data was spatially filtered by common-average re-referencing (CAR), and band-pass filtered between 1 and 10 Hz with a 4th order Butterworth filter, since EEG error correlates are known to be slow potentials. Time windows of 800 ms within each trial were extracted to compute time and frequency features. Time features were obtained by sub-sampling the signal in the epoched time windows by a factor of 4. Frequency features were selected from non-overlapping 2 Hz-wide spectrum bands between 2 and 10 Hz, computed using Fast Fourier Transform after epochs multiplication with a Blackman window. This procedure resulted in 45 time features and 8 frequency features for each channel.

Offline classifier training: In order to take into account possible variations in ErrP latency, reflecting differences in user's perception originated by the feedback condition and by the physical delay in the observed robot movements, feature sets were extracted as explained in the previous paragraph, considering time windows with different delays. Specifically, 10 possible delays ranging from 0 to 1 s after robot movement onset, in 100 ms steps, were considered for time window epoching. Thus, relevant features were obtained in windows ranging from 0-800 ms to 1000-1800 ms after robot movement onset. A different linear Support Vector Machine (SVM) classifier was trained on the feature set corresponding to each considered time window, without prior feature selection. The imbalance in correct (C) and error (E) classes occurrence was compensated by random downsampling of the majority (C) class. A 10-fold cross-validation was used to compute the classification performance, in terms of area under the ROC curve (AUC).

Online classification: For each subject, the time window (and therefore the time delay) that led to the higher classification accuracy in the offline analysis was selected. A linear SVM was trained on all available training data (without performing cross-validation), on data epoched according to the optimal time window. During online classification, this classifier was applied to the features extracted on ongoing EEG data, in the same time window, selected according to the chosen optimal delay. The output of such classification was sent to the learning algorithm, in terms of a label $\in$ [0,1] indicating whether the last robot movement (EEG trial) belonged to correct (0) or error (1) class.

2) Robot Learning: In order to implement the robot learning, a reinforcement learning architecture was chosen. In an RL framework, an agent learns through interaction with environment, based on a reward signal (r), which recompenses or penalizes its actions. The analytical framework of an RL problem is a Markov Decision Process defined by the tuple $\{S, A, P, r, \gamma\}$, where S is the state-space (in the reaching problem described here, it corresponds to the set of all discrete positions that the robot can assume), A is the action-space (in this case movements in four directions), and $P : S \times A \rightarrow S$ are the transitions probabilities from one state to the next one, when executing a particular action a. The function $r : S \times A \rightarrow R$ defines the reward obtained by the agent when executing an action a at state s, and $\gamma \in [0, ..1]$ is a discount factor on the total accumulated reward. The goal of RL is to find a policy $\pi : S \rightarrow A$ to map each state to the action that maximizes the accumulated reward $R_i = \sum_{i=0}^{n} \gamma^t r_{i+1}$. For discrete tasks, as the one considered here, the standard Q-Learning algorithm is the most suitable to compute the optimal policy [17]. The so-called Q-function is used to compute the state-action map:

$$\begin{aligned} Q^*(s,a) =& r(s,a) \\ &+ \gamma \sum_{s' \in S} P(s,a,s') max_{a' \in A} Q^*(s',a') \end{aligned} \quad (1)$$

The optimal Q^* function is estimated iteratively from empirical data. At each time step, the agent in state s_i executes an action a_i, that produces the agent being in a new state s_{i+1}, and receiving a reward $r_{i+1}(s_i, a_i)$, associated to this transition. The Q-function is updated according to:

$$\begin{aligned} Q_{i+1}(s_i,a_i) =& (1-\alpha_i) Q_i(s_i,a_i) + \alpha_i [r_{i+1}(s_i,a_i) \\ &+ \gamma max_{a' \in A} Q_i(s_{i+1},a')] \end{aligned} \quad (2)$$

Algorithm 1 Route Learning Strategy

1: $s_0 \leftarrow s_{start}$
2: **for** $i = 1$ to $Ntrials$ **do**
3: **if** $s_i \neq s_{final}$ **then**
4: Choose a_i from $Q(s_i, a)$ (ϵ-greedy)
5: Take action a_i leading to state s_{i+1}
6: **if** $ErrP = true$ **then** ▷ Error is detected
7: $r_{i+1}(s_i, s_{i+1}) \leftarrow r_i(s_i, s_{i+1}) * 0.9 - 1$ ▷ Update reward for last transition (converging to -10)
8: Update $Q(s_i, a_i)$ according to Eq.(2)
9: $r_i \leftarrow r_{i+1}$
10: $s_i \leftarrow s_{i+1}$
11: **else** ▷ Target is reached
12: $r(s_i, s_{i+1}) \leftarrow 100$ ▷ Set environmental reward
13: $s_i \leftarrow s_{start}$ ▷ Start a new iteration

Algorithm 2 Target Learning Strategy

1: Initialise $Q_{target}(s_t, a_t)$ as zeros matrix
2: Initialise $r_{target}(s_t, a_t)$ as zeros matrix
3: Randomly choose initial guessed target s_{t_j}, $t \in \{1, ..nTargets\}$
4: **for** $j = 1$ to $TargetLearningNtrials$ **do**
5: **if** $s_i \neq s_{final}$ **then**
6: Select route Q matrix corresponding to current guessed target, $Q[s_{t_j}]$
7: Choose a_i from $Q[s_{t_j}](s_i, a)$ (ϵ-greedy)
8: Take action a_i leading to state s_{i+1}
9: **if** $ErrP = true$ **then** ▷ Error is detected
10: $r_{target_{j+1}}(:, s_{t_j}) = r_{target_j}(:, s_{t_j}) - [Q[s_{t_j}](s_i, s_{i+1}) - mean(Q[s_{t_j}](s_i, \{s_k : k \neq i+1\}))]$ ▷ Penalize actions bringing to current guessed target
11: **for** $t_w \neq t_j$ **do**
12: $r_{target_{j+1}}(:, s_{t_w}) = r_{target_j}(:, s_{t_w}) + Q[s_{t_w}](s_i, s_{i+1}) - mean(Q[s_{t_w}](s_i, \{s_k : k \neq i+1\}))$ ▷ Reward actions bringing to different guessed target
13: Update $Q_{target}(s_{tk1}, a_{tk2})$ using Eq.(2), for $k1, k2 = 1..nTargets$
14: Choose a_{t_j} from $Q_{target}(s_{t_j}, a_t)$ (i.e. always optimal action) ▷ Compute new guessed target
15: Take action a_{t_j} leading to target $s_{t_{j+1}}$
16: **if** $s_{t_{j+1}}$=s_{t_j} **then** ▷ Guessed target does not change
17: Update route reward and state-action matrices corresponding to current guessed target $r[s_{t_j}]$, $Q[s_{t_j}]$, as in lines 6:10 of Algorithm 1
18: $r_{target_j} \leftarrow r_{target_{j+1}}$
19: $s_{t_j} \leftarrow s_{t_{j+1}}$
20: **else** ▷ Target is reached
21: Execute lines 12:13 of Algorithm 1
22: Execute lines 1:3 of Algorithm 2 ▷ Reset target learning data and start a new iteration

where Q_i is the current estimate, at step i, of the Q-function, α is the learning rate, and γ is a coefficient $\in$ [0, 1], accounting for the importance of long-term rewards. During learning, it is necessary to choose a policy a, i.e. which action to perform, from a particular state s. In this work, an ϵ-greedy policy was adopted: the best action, obtained from the current policy (current Q-map values), is chosen (100-ϵ)% of the times, while an exploration strategy (random action) is selected ϵ% of times. When the agent behavior approaches the desired one, ϵ is set to zero.

The Q-learning algorithm was used to implement both route learning, i.e. learning the optimal sequence of steps to reach each one among two available target positions, and target learning, i.e. once the routes towards the targets were sufficiently learned, the robot was allowed to choose which one of the two learned routes to follow, so as to match the will of the human observer. The learning strategies for the two tasks are explained in detail below.

Route Learning: The route learning strategy consisted of a slightly modified version of the basic Q-learning algorithm, and it is reported in Algorithm 1. For the discrete space of 25 positions considered for the task, the state-action Q-function consisted of a sparse 25x25 matrix representing the transition probabilities from each one of the possible 25 states to the sub set of 1 to 4 linked (reachable) states. The reward function r represented a 25x25 matrix of reward values assigned to each transition. Both matrices were initialized to zeros, and the initial state set to the central position of the grid, three steps away from each possible target.

Both targets were alternately presented during subsequent iterations, until 250 trials (robot steps) were performed towards each target. At each robot step, the Q-matrix was updated using (2). The output of the online ErrP classifier provided information about user detection of an error in the last observed robot step, and was used to update the corresponding transition value in the reward matrix r, according to line 7 in Algorithm 1, with the effect of discouraging that action. This allowed the implementation of a short-term negative reward. The r matrix was also updated with a positive value when the target was reached, thus allowing the forward term, multiplied by γ in (2), to start propagating information on the long-term environmental reward.

In this phase of the task, two Q matrices were learned, one for each target, and information on the current target was provided in a supervised way, by sending to the learning algorithm the true label for the current target. Learning rate was fixed to 0.7, while ϵ was initially set to zero (first iteration), to allow a faster learning of a possible route to the target, exploiting information provided by EEG ErrPs. After the fisrt iteration, it was increased to 0.4 to allow exploration of a route map robust to an ErrP misclassification, and for fast route correction in case of a sub-optimal learned route. When the best (3 steps-wide) route was learned, ϵ was set to 0.2, and then to 0 during target learning. Similarly, γ was initially set to 0.5 to avoid the strong learning of a non-optimal route, and then increased to 0.9 when the best route was found, in order to rapidly consolidate optimal information.

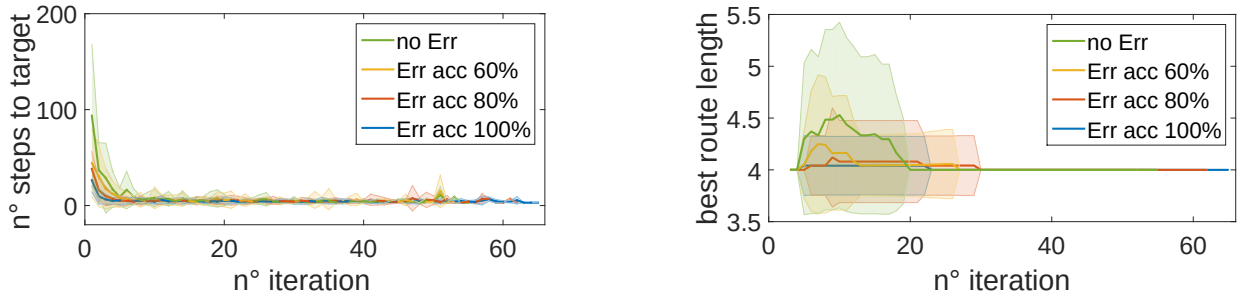

Fig. 3: Simulation results for route learning. Number of moves to the target (left), and length (number of crossed states) of optimal route (right).

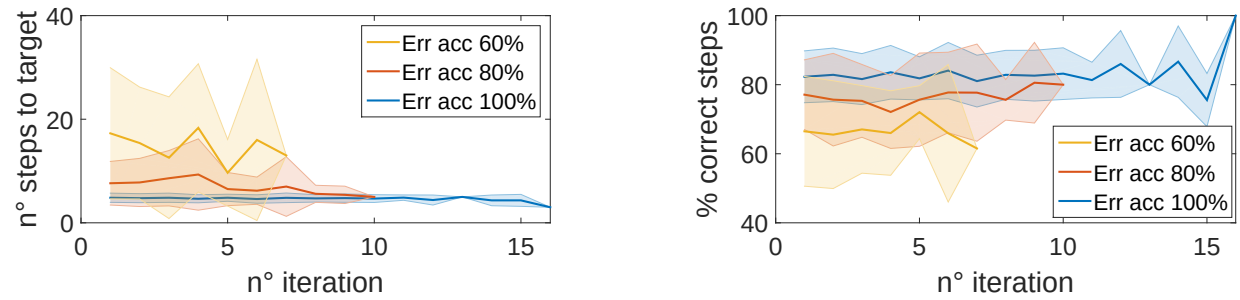

Fig. 4: Simulation results for target learning. Number of moves to the target (left), and % of moves in which correct target is identified (right).

Target Learning: In this phase, the robot had to learn which one of the two available maps to choose, in order to reach the current target. To this aim, the information on ErrP detection after each step, together with the one provided by the learned Q-maps, was used. As explained in Algorithm 2, the learning algorithm for target identification was restarted at each iteration: target learning, differently from route learning, should desirably be fast, without the possibility of exploiting accumulated information. For this reason, γ was set to zero.

Similarly to the route learning case, a Q and r matrix were defined: Q was a 2x2 fully linked matrix with possible actions corresponding to conserving the guessed target or switching to the other one. The strategy used to make such a decision was based on the q-values accumulated in the Q matrices corresponding to the routes towards each target. Specifically, as explained in lines 9-10 of Algorithm 2, after a robot step, if an error was detected, a negative reward was assigned to all actions keeping the same guessed target, proportional to the q-value for that transition in the route Q matrix of the current guessed target. A positive reward was simultaneously assigned to all actions leading to changing the guessed target, proportional to the q-value for the same transition in the route Q matrix of the other target (lines 11-12). After reward updating, the target Q matrix was also updated using Eq. (2), with $\alpha = 0.7$, and the optimal action was always chosen ($\epsilon = 0$). If the selected action did not bring to change the current guessed target, ErrP detection was used to continue the corresponding route learning (Algorithm 1).

III. SIMULATION STUDY

Before practical implementation, the route and target learning algorithms described in Section II-C.2 were tested in a simulation study, comparing their performance for different simulated levels of ErrP detection accuracy. Results are reported in Fig. 3 and 4.

For route learning, Fig. 3 reports average values and standard deviations (over 50 simulations) for the number of steps taken from the starting position to the target (left panel), and the length (number of states crossed) of current best route (right panel), against number of iterations. Results are compared for different rates $(100\%, 80\%, 60\%)$ of ErrP correct detection, and for simple Q-learning (without using ErrP information for step-by-step reward updating). With 100% ErrP detection accuracy, the inclusion of the negative step-by-step rewards reduces the number of steps required during the first iteration from 80 to 20, and considerably reduces the variability among task repetitions. Also, ErrP inclusion in Q-learning reduces the length of non-optimal learned routes during early iterations, thus hastening route correction. These results are consistent also when a misclassification rate in ErrP detection is simulated, even if with a progressive decrease of performance. For 80% of ErrP detection accuracy, which corresponds to the state-of-the-art standard for ErrP detection in simulated tasks, the algorithm performance is very close to the 100% accuracy case. Even for lower accuracy in ErrP detection, the algorithm allows for a more efficient performance than a simple Q-learning.

For target learning, the number of steps taken from the starting position to the target, and the percentage of steps with correctly guessed target during each iteration, are shown in Fig. 4. In this case, simple Q-learning cannot be implemented, and in fact, one of the contributions of the proposed use of ErrP is the possibility of having an additional degree of freedom in the learning. Even in this case, an increase in ErrP classification accuracy reduces the number of moves necessary to reach the target, and increases the correct steps

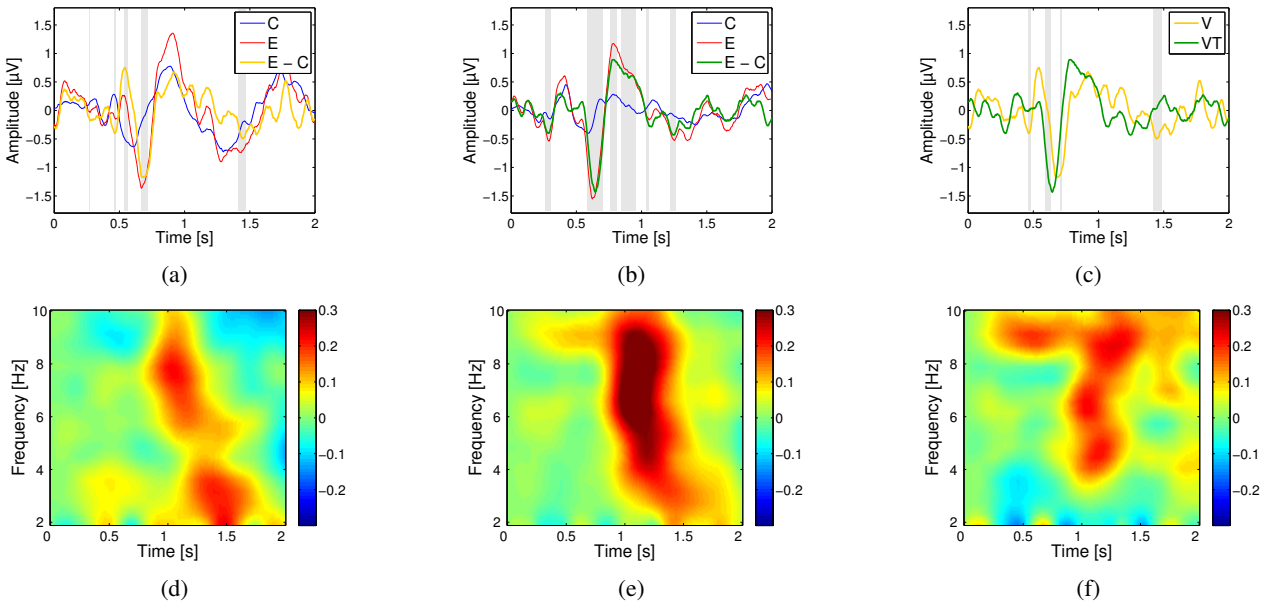

Fig. 5: Grand average of ErrPs (E-C) across all subjects and training trials for channel FCz. Panels (a) and (b) display average C, E, and ErrP for V and VT conditions. Panel (c) provides a comparison between V and VT ErrPs. Panels (d-f) provide a graphical representation of ErrP in frequency domain, for V (d), VT (e), and VT-V (f).

percentage, i.e., steps with correctly guessed target.

In the optimal case (100% ErrP accuracy), the number of steps to reach the target corresponds to the optimum value for the considered task: 3 or 5 steps, respectively when the algorithm correctly guesses the target at the first step, or when it initially guesses the wrong target, and it needs one erroneous step to correct the identification. For 80% accuracy, the mean number of steps to reach the target is 7, corresponding to 2 erroneous steps necessary to correct the target guess. For lower accuracy (60%), the correct steps ratio decreases to 70%, and the number of steps to the target increases to 15 on average.

IV. RESULTS AND DISCUSSION

A. EEG Analysis

1) ErrP Detection: Fig. 5 shows the grand averages of E, C, and ErrPs (E-C) signals, across subjects and trials, for V (Fig. 5a) and VT (Fig. 5b) conditions. The difference E-C is also reported in the frequency domain for V (Fig. 5d) and VT (Fig. 5e). Right-column panels highlight the difference between ErrPs in the two feedback modalities, both in time (Fig. 5c) and frequency domain (Fig. 5f). Shaded areas in time graphs represent significant differences between C and E responses (5% significance level), according to t-tests.

The difference between classes present a similar pattern for both considered feedbacks (Fig. 5a and 5b): a first significant negative peak centered around 270 ms after stimulus onset, and a larger significant negative peak, with center around 650 ms after stimulus presentation. All other significant features differ between conditions, with two relevant peaks in the V condition right before the 650 ms negative peak, while the VT condition shows significant differences in the whole time window between 700 and 1200 ms. In general, recorded responses are very similar in shape to what is reported in the literature [18], but with a significant delay (around 200 ms). This reflects observations in previous works, reporting that the latency of ErrP waveform varies among different tasks, typically with increasing delay when coping with observation of physical movements and complex tasks [19].

On frequency domain, the most evident feature, clearly visible both in V (Fig. 5d) and VT (Fig. 5e) responses, is an increase of up to 30% in the spectral density power following feedback stimuli. The timing of this increase is almost matching for both tested conditions and occurs between 1 s and 1.5 s after stimulus onset, while the peak frequencies involved are between 5-10 Hz. Despite the matching position in both time and frequency, this increase is about 50% more intense in the case of VT feedback. A marked difference in response, on the other hand, occurs earlier in time (around 0.5 s after stimulus) at a frequency of 9 Hz: while the VT feedback causes an increase of this response component, the visual-only response results weakened compared to baseline.

2) ErrP Classification: Classification results, in terms of AUC, are reported in Tables I and II for V and VT feedback conditions, respectively. It is possible to note how VT feedback generally leads to more consistent responses (smaller standard deviations), and classification accuracy tends to peak earlier for VT than V condition (0.3 s delay against 0.53 s). Classification accuracy obtained from training data varies significantly between subjects, ranging from very good (highest AUC = 0.85), to discrete (lowest AUC = 0.61), with average values around 0.7 (0.73 and 0.69 for V and VT feedback). These values drop by almost 10% during testing phase, to 0.60 and 0.59 on average for V and VT conditions respectively. This is likely due to subjects' attention lapses, given the length of the experiment. Also the differences, when moving from training to testing, in the observed task (1-D to 2-D) and in errors occurrence (decreasing according to

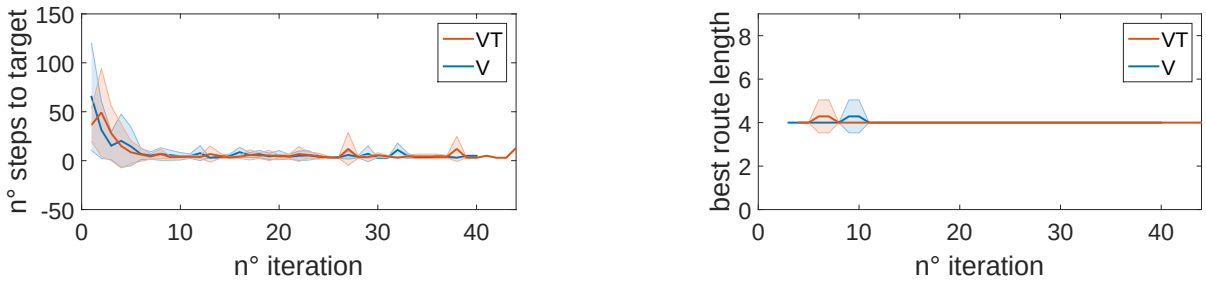

Fig. 6: Experimental results for route learning. Number of robot moves to the target (left), and length (number of crossed states) of optimal route (right).

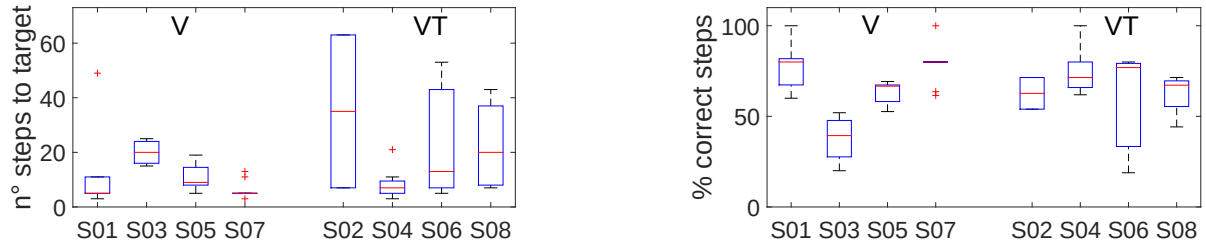

Fig. 7: Experimental results for target learning. Number of robot moves to the target (left), and % of moves in which correct target is identified (right).

TABLE I: ErrP classification results during training and testing, for subjects with V feedback.

Subj.	Delay[s]	Training			Testing		
		C	E	AUC	C	E	AUC
S01-V	0.50	0.82	0.74	0.85	0.61	0.64	0.68
S03-V	0.60	0.59	0.61	0.62	0.52	0.49	0.52
S05-V	0.40	0.65	0.60	0.68	0.63	0.51	0.59
S07-V	0.60	0.67	0.71	0.75	0.6	0.59	0.62
Mean	**0.53**	**0.68**	**0.67**	**0.73**	**0.59**	**0.56**	**0.60**
Sd	**0.08**	**0.08**	**0.06**	**0.09**	**0.04**	**0.06**	**0.06**

TABLE II: ErrP classification results during training and testing, for subjects with VT feedback.

Subj.	Delay[s]	Training			Testing		
		C	E	AUC	C	E	AUC
S02-VT	0.50	0.70	0.69	0.74	0.63	0.42	0.55
S04-VT	0.20	0.55	0.62	0.61	0.51	0.62	0.57
S06-VT	0.20	0.73	0.57	0.72	0.62	0.64	0.66
S08-VT	0.30	0.65	0.67	0.69	0.58	0.51	0.59
Mean	**0.30**	**0.66**	**0.64**	**0.69**	**0.59**	**0.55**	**0.59**
Sd	**0.12**	**0.07**	**0.05**	**0.05**	**0.05**	**0.09**	**0.04**

learning), could play a significant role in such a decrease in performance. To statistically evaluate the observed results, a 2x2 mixed model ANOVA (assumptions of the test checked) was performed, with AUC as the dependent variable, V/VT as the between-group independent variable, and training/testing as the within-group independent variable. No global effect of the interaction between the two variables was found ($F_{1,6} = 0.437$, $p = 0.533$). Overall, no effect of the V/VT variable was found ($F_{1,6} = 0.240$, $p = 0.642$) while an effect of the training/testing variable was found ($F_{1,6} = 33.866$, $p = 0.001$). This latter is mainly caused by the V condition, since an effect of the training/testing variable was found only for V condition ($F_{1,3} = 46.47$, $p = 0.006$).

B. Robot Learning

Results collected during online robot learning are reported in Fig. 6, for route learning trials, and Fig. 7, for target identification (last 100 trials)[3].

1) Route Learning: Left and right panel of Fig. 6 show average results across subjects in terms of number of steps to the target (left panel), and length of best route, i.e. number of crossed states (right panel), against iterations, considering V and VT feedback conditions separately.

While ErrP classification accuracy, in terms of mean AUC (see Table I and II), was not varying among V (0.60) and VT (0.59) conditions, the average number of steps and the variability across subjects results lower for VT condition than for V (36.5 $\pm$16.9 against 65.3 $\pm$ 54.8), in the first iteration. Here no exploration was allowed and only information from ErrP signal was used to update the reward and choose the best action. The observed values match respectively an accuracy of around 80% and 60% in the simulated data (see Fig. 3 left panel). This observation suggests that ErrP classification accuracy could be not uniformly distributed over all experiment duration, especially for the VT condition.

In both conditions, the number of steps converge to the minimum at around nine iterations, as in the case of simulated 60% ErrP detection accuracy. On the other hand, results related to best route length show that for the majority of subjects in VT condition the route correction happens between the 5th and the 8th iteration, while for V condition this is observed between 8th and 11th iteration. On simulated data (Fig.3 right), it appears that an earlier route correction happens for lower ErrP detection accuracy, since more states

[3]A video of the online learning algorithm functioning is available at the link: https://goo.gl/SnDwe9

are explored in the first iterations compared to cases with high ErrP detection accuracy. In contrast with this observation, on experimental data route correction for VT condition happens earlier but without requiring a higher number of steps.

2) Target Learning: Fig. 7 reports boxplots for distributions of number of steps to target, and % of correct steps across iterations, for each subject during the phase of robot target learning, when route exploration was set to zero.

Comparing results from experiments with V and VT feedback, the VT condition results in a much bigger subjective variability in terms of moves to the target across iterations. At the same time, the % of correct steps is on average between 60 and 80%, and more uniform across subjects undergoing VT condition compared to those testing V feedback. Also in this case the experimental results are coherent to what expected from simulations for an ErrP detection accuracy of around 60%, as the one computed on testing data.

V. CONCLUSIONS

This work presented the evaluation of an ErrP-based reward signal to include human in the loop of robot learning, in order to: 1) improve a classic RL algorithm for a robot target reaching task, and 2) introduce the possibility of identifying the target selected by the user among a predefined set of possible positions, in a human-robot co-operative scenario.

The learning algorithm was evaluated offline simulating different levels of ErrP detection accuracy, and then validated through online experiments involving 8 subjects, using two different feedback conditions superimposed on the robot movements: visual and visuo-tactile. Results showed that, even with a low accuracy in ErrP online detection (60%), the proposed framework allowed for a faster convergence of the learning algorithm compared to classic Q-learning, and for successful target identification with a mean accuracy of 70%.

The big decrease in ErrP correct detection from training to testing (70% to 60%) pointed out some difficulties related to the use of ErrP in a realistic human-robot co-operative task. To cope with such issue there is a need for exploring techniques able to improve ErrP online classification accuracy, e.g. exploiting unsupervised adaptive learning methods [20], as well as using generalization methods to enable knowledge transfer between different training and testing tasks.

The addition of tactile feedback did not result in a significant variation of ErrP detection accuracy, although it produced some encouraging differences in the learning behavior. In future work, customized and more sophisticated designs of feedback will be taken into account.

Overall, these results confirmed the high potential of introducing human psychophysiological signals into the learning process of an intelligent agent, towards an improved human-robot interaction aimed at developing comfortable and user-centered assistive solutions.

ACKNOWLEDGMENT

The research leading to these results has been partially funded by Fondazione Roma under the grant agreement supporting the TEEP-SLA project.

REFERENCES

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[2] L. Tonin, R. Leeb, M. Tavella, S. Perdikis, and J. d. R. Millán, "The role of shared-control in bci-based telepresence," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1462–1466.

[3] D. Vanhooydonck, E. Demeester, M. Nuttin, and H. Van Brussel, "Shared control for intelligent wheelchairs: an implicit estimation of the user intention," in *Proceedings of the 1st international workshop on advances in service robotics (ASER03)*, 2003, pp. 176–182.

[4] H. K. Kim, J. Biggs, W. Schloerb, M. Carmena, M. A. Lebedev, M. A. Nicolelis, and M. A. Srinivasan, "Continuous shared control for stabilizing reaching and grasping with brain-machine interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1164–1173, 2006.

[5] J. R. Millan, F. Renkens, J. Mourino, and W. Gerstner, "Noninvasive brain-actuated control of a mobile robot by human eeg," *IEEE Transactions on biomedical Engineering*, vol. 51, no. 6, pp. 1026–1033, 2004.

[6] K. A. Tahboub, "A semi-autonomous reactive control architecture," *Journal of Intelligent & Robotic Systems*, vol. 32, no. 4, pp. 445–459, 2001.

[7] R. Chavarriaga and J. d. R. Millán, "Learning from eeg error-related potentials in noninvasive brain-computer interfaces," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 18, no. 4, pp. 381–388, 2010.

[8] P. W. Ferrez and J. d. R. Millán, "You are wrong!—automatic detection of interaction errors from brain waves," in *Proceedings of the 19th international joint conference on Artificial intelligence*, no. EPFL-CONF-83269, 2005.

[9] I. Iturrate, L. Montesano, and J. Minguez, "Single trial recognition of error-related potentials during observation of robot operation," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 4181–4184.

[10] P. Margaux, M. Emmanuel, D. Sébastien, B. Olivier, and M. Jérémie, "Objective and subjective evaluation of online error correction during p300-based spelling," *Advances in Human-Computer Interaction*, vol. 2012, p. 4, 2012.

[11] A. F. Salazar-Gomez, J. DelPreto, S. Gil, F. H. Guenther, and D. Rus, "Correcting robot mistakes in real time using eeg signals," *ICRA. IEEE*, 2017.

[12] A. Llera, V. Gómez, and H. J. Kappen, "Adaptive classification on brain-computer interfaces using reinforcement signals," *Neural Computation*, vol. 24, no. 11, pp. 2900–2923, 2012.

[13] I. Iturrate, L. Montesano, and J. Minguez, "Robot reinforcement learning using eeg-based reward signals," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4822–4829.

[14] I. Iturrate, R. Chavarriaga, L. Montesano, J. Minguez, and J. d. R. Millán, "Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control," *Scientific reports*, vol. 5, 2015.

[15] I. Iturrate, J. Omedes, and L. Montesano, "Shared control of a robot using eeg-based feedback signals," in *Proceedings of the 2nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication*. ACM, 2013, pp. 45–50.

[16] J. Tessadori, L. Schiatti, G. Barresi, and L. S. Mattos, "Does tactile feedback enhance single-trial detection of error-related eeg potentials?" in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 1417–1422.

[17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[18] R. Chavarriaga, A. Sobolewski, and J. del R Millán, "Errare machinale est: The use of error-related potentials in brain-machine interfaces." *Frontiers in Neuroscience*, vol. 8, 2014.

[19] I. Iturrate, R. Chavarriaga, L. Montesano, J. Minguez, and J. R. del Millan, "Latency correction of error potentials between different experiments reduces calibration time for single-trial classification," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 2012, pp. 3288–3291.

[20] Q. Huang, D. Yang, L. Jiang, H. Zhang, H. Liu, and K. Kotani, "A novel unsupervised adaptive learning method for long-term electromyography (emg) pattern recognition," *Sensors*, vol. 17, no. 6, p. 1370, 2017.