Shared control of a robot using EEG-based feedback signals

Iñaki Iturrate
I3A, DIIS
EINA, University of Zaragoza
Zaragoza, Spain
iturrate@unizar.es

Jason Omedes
I3A, DIIS
EINA, University of Zaragoza
Zaragoza, Spain
jomedes@unizar.es

Luis Montesano
I3A, DIIS
EINA, University of Zaragoza
Zaragoza, Spain
montesano@unizar.es

ABSTRACT

In the last years there has been an increasing interest on using human feedback during robot operation to incorporate non-expert human expertise while learning complex tasks. Most work has considered reinforcement learning frameworks were human feedback, provided through multiple modalities (speech, graphical interfaces, gestures) is converted into a reward. This paper explores a different communication channel: cognitive EEG brain signals related to the perception of errors by humans. In particular, we consider error potentials (ErrP), voltage deflections appearing when a user perceives an error, either committed by herself or by an external machine, thus encoding binary information about how a robot is performing a task. Based on this potential, we propose an algorithm based on policy matching for inverse reinforcement learning to infer the user goal from brain signals. We present two cases of study involving a target reaching task in a grid world and using a real mobile robot, respectively. For discrete worlds, the results show that the robot is able to infer and reach the target using only error potentials as feedback elicited from human observation. Finally, promising preliminary results were obtained for continuous states and actions in real scenarios.

Keywords

Reinforcement learning; Brain-machine interfaces

General Terms

Algorithms, Design, Experimentation, Human Factors

Categories and Subject Descriptors

H. Information Systems [**H.5 INFORMATION INTERFACES AND PRESENTATION**]: H.5.2 User Interfaces

1. INTRODUCTION

When learning complex tasks, robots are usually faced with vast action-state spaces which are difficult and expensive to explore without prior knowledge of the structure of the environment. Furthermore, there exist hazardous regions or configurations that should be avoided since they may be dangerous for the robot or people around the robot. Humans are naturally aware of the intrinsic structure and domain knowledge of a task and, consequently, they can provide feedback during robot operation for learning or control purposes. Indeed, this feedback is a very powerful way to provide supervision during robot learning for complex tasks [19]. This use of human feedback during robot operation and learning requires some kind of communication between the human and the robot, where the most common modalities include speech gestures and physical interaction [1].

Most work in this area has used a reinforcement learning framework to incorporate human feedback during the learning process [9]. Since feedback occurs through the interaction between human and robots, the human shapes the reward according to her own understanding of the task. Some authors have studied how to model binary feedback (e.g. approval or disapproval) and incorporate it to the learning process [12, 19, 20]. Another important issue is that during interaction humans do not only provide feedback but also tend to provide guidance for future actions [18].

This paper explores a different communication channel to provide feedback to robots using brain signals. Brain-machine interfaces (BMI) have been proposed in the last years as a way of communicating with virtual or real devices using only brain activity. Among the different ways of recording the brain signals, non-invasive electroencephalogram (EEG) is the most extended one despite its low signal-to-noise ratio, mainly due to its easiness of use and portability. EEG has been successfully used for the control of robotic arms, wheelchairs or mobile robots among others (see [14] for a review). However, in most cases BMIs decouple the operation of the device from the mental task used to control the robot (e.g. motor imagery of body limbs to operate a virtual cursor) and there has been little effort in terms of using brain signals that directly encode cognitive information about the task itself and, in particular, feedback information about the behavior of the robot.

A promising cognitive EEG signal are the so-called error potentials (ErrPs), signals elicited and measurable in the user's EEG after she commits or perceives an error [7]. More interestingly, this signal is also visible when the user observes a machine committing an error [5, 8]. Thus, ErrPs are a natural candidate as feedback for a robotic device directly extracted from brain activity [10]. They present advantages and disadvantages with respect to other types of feedback signals. First, they can be difficult to detect due to their low signal-to-noise ratio and, when detected, they contain little information about the nature of the error. Indeed, they can be seen as a binary signal indicating the absence or presence of an error. In addition to this, these signals have been usually studied using locked stimulus (e.g. a clear visual cue synchronized with the EEG) which makes their detection easier than during the con-

MLIS'13, August 03 - 04 2013, Beijing, China

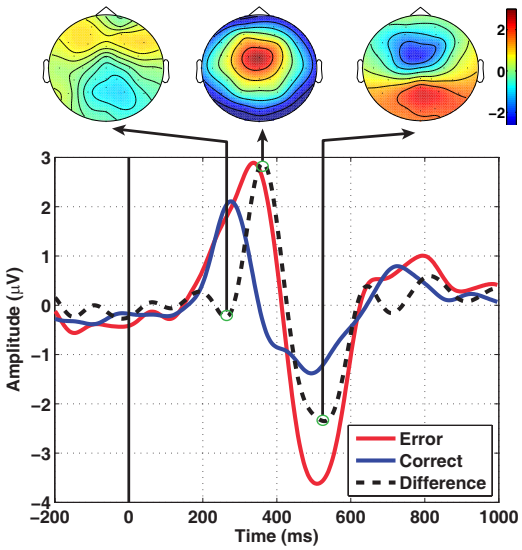

Figure 1: Error potentials elicited when assessing actions as correct or erroneous, and difference between the two assessments (time 0 ms indicates the action onset). The topographic head interpolations on the largest peak values are also shown.

tinuous operation of a robot. On the positive side, ErrPs provide a direct access to user's assessment of the robot operation without the ambiguities, connotations and conventions of other communication protocols, and in principle without possibility of distorting them.

This paper proposes the use of EEG error potentials as feedback for controlling a robot. We exemplify the main idea for a target reaching task in two different scenarios: a simple, virtual grid world; and a 2D real mobile robot navigation task. During the experiments, the role of the user was simply to evaluate the robot actions as correct or wrong, while the robot tried to learn and reach the intended user's goal. In order to cope with the limited information provided by ErrPs, we use a shared-control strategy based on the inverse reinforcement learning framework, where the robot maintains a belief over a set of possible targets updated using feedback signals extracted from brain activity during robot operation. For discrete worlds, the results show that the robot is able to reach the target using only ErrPs as feedback elicited from human observation. Finally, promising preliminary results for continuous domains and real robots are also reported using a mobile robot.

2. TRANSLATION OF EEG ERROR POTENTIALS INTO FEEDBACK SIGNALS

Error potentials belong to the family of event-related potentials (ERP) [13], voltage deflections appearing on the EEG after the occurrence of an event. In our case, they are elicited during the observation of actions (events) performed by a device. The user task is simply to assess the actions as correct or incorrect, which generates different signals for the two different conditions. Figure 1 shows an averaged example of these signals, together with their topographic head interpolations depicting the EEG activity map. In the remainder of the section, we describe the error potential detection process including EEG recording, BCI calibration and online detection during robot operation.

2.1 Data Recording

Electroencephalographic (EEG) and electrooculographic (EOG) activity were recorded using a gTec system. For the EEG, 32 electrodes were recorded, distributed according to an extended 10/20 international system (FP1, FP2, F7, F8, F3, F4, T7, T8, C3, C4, P7, P8, P3, P4, O1, O2, AF3, AF4, FC5, FC6, FC1, FC2, CP5, CP6, CP1, CP2, Fz, FCz, Cz, CPz, Pz and Oz), with the ground on FPz and the reference on the left earlobe; for the EOG, 6 monopolar electrodes were recorded (placed above and below each eye, and from the outer canthi of the left and right eyes [6]), with the ground on FPz and the reference on the left mastoid. The EEG and EOG signals were digitized with a sampling frequency of 256 Hz, powerline notch filtered, and band-pass filtered at $[1, 10]$ Hz. The EEG was also common-average-reference (CAR) filtered. Additionally, the horizontal, vertical, and radial EOG were computed as in [6] to remove the EOG from the EEG using a regression algorithm [16]. The data acquisition and online processing was developed under a self-made BCI platform.

2.2 Calibration of error potentials

Although the grand averages of Fig. 1 show a clear difference between error and non-error signals, single trial recordings present large variability due to EEG low signal-to-noise ratio and non-stationarities. Moreover, different tasks induce slight variations on the ErrPs [11]. Therefore, it is common to carry a user-specific calibration phase prior to the control phase as such.

During calibration, examples of error and non-error responses are elicited in a controlled manner and used to train a classifier. When the actions are discrete and instantaneous (e.g. moving in a grid world), the events that elicit the ErrPs are clearly defined in time as it is common in the general case of event-related potentials [13] and in particular in ErrPs [5, 8, 10]. However, in many robotic tasks, robot actions are continuous (e.g. a mobile robot moving towards a target or a manipulator trying to grasp an object) and the elicitation of the potential will occur in an undetermined point in time according to the user's assessment of the task. We next describe how we detect error potentials in each of these situations.

2.2.1 Discrete actions

Following previous studies [10], features were extracted from eight fronto-central channels (Fz, FC1, FCz, FC2, C1, Cz, C2, and CPz) within a time window of $[200, 800]$ ms (being 0 ms the action onset) downsampled to 64 Hz, forming a vector of 312 features. The features were then normalized, and its dimensionality reduced with PCA retaining 95% of the variance. A regularized linear discriminant (LDA) [3] was trained using the previous features. The classifier output has the form $y(\mathbf{x}) = \mathbf{w}'\mathbf{x} + b$, where $y(\mathbf{x}) < 0$ was classified as a correct assessment (class 0), and $y(\mathbf{x}) \geq 0$ as an error assessment (class 1). This output $y(\mathbf{x})$ was transformed into the probability that an example $\mathbf{x}$ was an error, $p(c = 1|\mathbf{x}) = \frac{1}{1+e^{-y(\mathbf{x})}}$ [2].

2.2.2 Continuous actions

For continuous actions there is not a clear trigger for the elicitation of the error potential. During calibration, this can be solved by using two trigger buttons pressed by the users according to their assessments. During the control phase, this trigger is removed, and the classification is performed using a overlapping sliding window (fixed to steps of 62.50 ms for the experiments).

In addition, the absence of a proper cue difficult the ErrPs detection, making the temporal features described above insufficient to obtain low misdetection rates. To mitigate this effect, we added an additional set of features from the frequency domain, namely the power spectral density (PSD), which are rather insensitive to time shifts. The PSD is calculated on 800 ms of EEG for each of the

eight fronto-central channels used before. The new features are the power values in the theta band ($[4, 8]$ Hz) $\pm$ 1 Hz for each channel (as previous studies suggest that the error potentials are generated within this band [4]), making a vector of 200 features. Single-trial classification was carried out using a support vector machine (SVM) with a radial basis function (RBF) kernel, whose output was the probability that an example x was an error, $p(c = 1|\mathbf{x})$.

3. SHARED-CONTROL OF A REACHING TASK VIA FEEDBACK BRAIN SIGNALS

This section describes the proposed shared-control strategy that allows the robot to simultaneously infer the user's intended goal and reach it using ErrPs. Although ErrPs provide feedback about the device actions, the amount of information conveyed by them is limited. In particular, the decoders (see previous section) do not contain any information about direction or magnitude, and have a non-negligible number of misdetections. The proposed shared control uses an inverse reinforcement learning algorithm to accumulate evidence about a set of predefined possible goals while executing a trajectory. The proposed approach consists of two phases. The first one computes offline optimal trajectories (i.e. policies) for each potential target, while the second performs an online policy matching to rank them during robot operation based on error potentials elicited for wrong actions.

We next give a general view of the method, which is then particularized in the next two sections. Let **s** and **a** denote the state of the world and a robot action. Given a set of possible targets, let $f_i(\mathbf{s}, \mathbf{a})$ be the value function [17] that describes the value of executing action **a** in state **s** for a given target i. The optimal policies can be obtained from $f_i(\mathbf{s}, \mathbf{a})$ as:

$$\pi_i^*(\mathbf{s}) = \arg\max_{\mathbf{a}} f_i(\mathbf{s}, \mathbf{a}). \quad (1)$$

In the examples of the next sections these functions can be computed exactly, although in general it may be necessary to approximate them.

During the control phase, the value functions are used to estimate the probability of each target by measuring how well non-error actions match the policies of each target. At each time step t, the device performs an action $\mathbf{a}_t$ from state $\mathbf{s}_t$. Let $\mathbf{x}_t$ denote the EEG window corresponding to time t and $p(c_t = 1|\mathbf{x}_t)$ be the probability provided by the ErrP decoder described in subsection 2.2. Let $p(\pi_i^* \mid (\mathbf{a}, \mathbf{s}, \mathbf{x})_{1\ldots t})$ be the posterior probability of policy π_i^*, that is, of target i being the one selected by the user. This posterior is computed recursively for each new action executed by the robot

$$p(\pi_i^*|(\mathbf{a}, \mathbf{s}, \mathbf{x})_{1\ldots t}) \propto p(\mathbf{a}_t|\pi_i^*, (\mathbf{s}, \mathbf{x})_t) \cdot p(\pi_i^*|(\mathbf{a}, \mathbf{s}, \mathbf{x})_{1\ldots t-1}), \quad (2)$$

where the likelihood $p(\mathbf{a}_t|\pi_i^*, (\mathbf{s}, \mathbf{x})_t)$ measures the dissimilarity (similarity) between the executed action and the policy of target i when an error (non-error) is detected from the EEG. The actual implementation depends on the protocol and is described in the next sections. The execution finishes when a probability $p(\pi_i^*)$ reaches a convergence criterion, p_c.

4. DISCRETE REACHING TASK

4.1 Experimental design

The visual protocol is shown in Fig. 2. The protocol consisted of a virtual cursor (green circle) that could perform discrete actions within a 5x5 grid, and its goal was to reach the target location

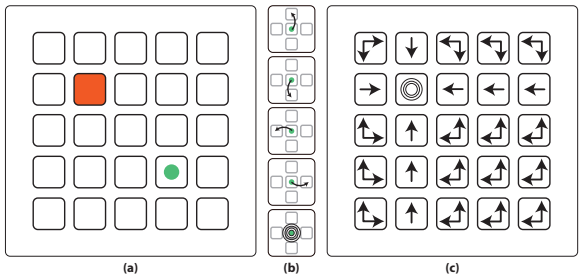

Figure 2: (a) Experimental protocol designed. The protocol showed a 5x5 grid with a virtual cursor (green circle) and a goal location (shadowed in red). (b) The cursor could perform five different actions (from top to bottom, move one position up, down, left or right, or performing a goal-reached action). (c) Optimal policy for the goal exemplified on (a).

(shadowed in red). The cursor could perform five different instantaneous actions: move one position left, right, up or down; and a goal-reached action, represented as concentric blue circumferences (see Fig. 2b). The time between two actions was random within the range $[3, 3.5]$ s. The users evaluated the actions as correct for (i) a movement towards the goal position, or (ii) a goal-reached action over the goal position; and as incorrect otherwise (see Fig. 2c). Four subjects (mean age 26 ± 2 years) performed the experiments, seated one meter away of a computer screen displaying the protocol. The users were instructed not to move their eyes during the cursor actions, and to restrain blinks only to the resting periods.

During the calibration phase, the device performed random actions with 20% of probability of performing an incorrect one. This phase lasted for 30 minutes, acquiring around 80 correct and 320 erroneous examples. During the control phase, two different groups of goal locations were tested: (i) the first group (denoted fixed goals) was shared for all the subjects, and consisted of five goals and initial cursor positions (see Figure 4); (ii) for the second group (denoted free goals), each user was asked to freely choose five different initial cursor positions and goals to reach. During this group of goals, the goal position was not shadowed in red, since it was the user who chose it.

4.2 Shared-control strategy

For this protocol, the value function $f_i(\mathbf{s}, \mathbf{a})$ was computed from the Q-values $Q_i^*(\mathbf{s}, \mathbf{a})$, which can be computed prior to the control phase using the Q-learning reinforcement learning algorithm [17]. Once calculated, the Q-values were converted into probabilities, following a soft-max normalization:

$$f_i(\mathbf{s}, \mathbf{a}) = \hat{Q}_i^*(\mathbf{s}, \mathbf{a}) = \frac{e^{Q_i^*(\mathbf{s},\mathbf{a})/\tau}}{\sum_{\mathbf{b}} e^{Q_i^*(\mathbf{s},\mathbf{b})/\tau}}, \quad (3)$$

where τ is denoted the temperature (fixed to $\tau = 0.3$). This parameter served as a degree of reliability of the observed information (classifier output).

The likelihood function was computed as follows:

$$p(\mathbf{a}_t|\pi_i^*, (\mathbf{s}, \mathbf{x})_t) = p(c_t = 0|\mathbf{x}_t) \cdot \hat{Q}_i^*(\mathbf{s}_t, \mathbf{a}_t) + p(c_t = 1|\mathbf{x}_t) \cdot (1 - \hat{Q}_i^*(\mathbf{s}_t, \mathbf{a}_t)), \quad (4)$$

Notice that the first term of the likelihood represents how we should increase the policy π_i^* if the user's assessment was correct, while the second term penalized the policy π_i^* weighted by the probability of having and incorrect user's assessment. Figure 3 shows several examples of actions and likelihoods. For the performed experiments, a new action $\mathbf{a}_{t+1}$ was chosen following an ε-

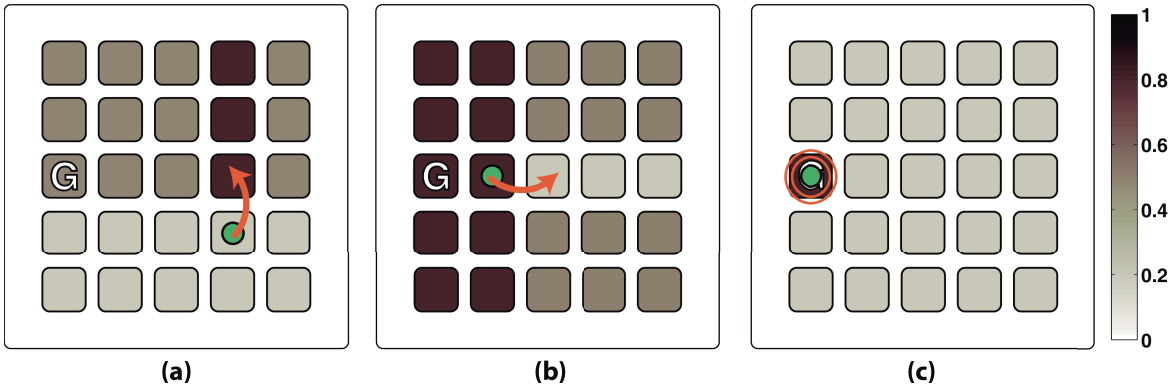

Figure 3: Likelihoods of each policy π_i after performing different actions: (a) correct movement with $p(c_t = 1|x_t) = 0.2$ (b) incorrect movement $p(c_t = 1|x_t) = 0.8$ (c) or a goal-reached action $p(c_t = 1|x_t) = 0.2$. The goal position is marked with a capital G.

greedy strategy, and the run finished when reaching a convergence criterion of $p_c = 0.9$.

4.3 Results

For each group of goals (fixed and freely-chosen), five metrics were evaluated: (i) Number of goals reached; (ii) number of actions needed to reach the goal, (iii) EEG seconds needed to reach the goal (net time); (iv) total time needed to reach the goal; and (v) classifier accuracy, measured as the percentage of detection of correct and erroneous signals. Note that the difference between the net and total times was the seconds belonging to inter-action intervals, which could be easily removed.

Table 1 shows the results for each subject and group of goals. The main result was that the device always reached the targets from any starting point, needing 25 ± 13 actions and 21 ± 8 actions (for the fixed and freely chosen goals) to reach the target. With inter-action intervals of around 3.25 s, the total time needed to reach the goals was of 80.76 ± 73.68 and 66.63 ± 26.85 seconds (fixed and free goals). Nonetheless, the net time (the seconds of EEG signal used for decoding) was of 19.88 ± 10.75 and 16.40 ± 6.61 seconds. The mean classifier accuracy was of 74.38 ± 4.66 and 77.67 ± 5.02. As expected, there was a significant negative correlation between the classifier mean accuracy and the time needed to reach the task ($r = -0.47, p = 0.038$ and $r = -0.79, p = 3 \cdot 10^{-5}$ for fixed and free goals). An interesting result was that not all the states were visited to reach the goal (see Figure 4). For instance, during run 3, mostly all the central states were visited, whereas the peripheral states were not. This could allow for a better scalability of the system (e.g. as the state space was increased, the percentage of visited states would decrease).

5. CONTINUOUS REACHING TASK

5.1 Experimental design

The second experiment consisted in reaching a target location with a low cost mobile robot (ePuck, [15]). The experimental protocol is shown in Figure 5a. The arena was a 200×200 cm^2 map, that was discretized into a 5×5 of possible goal positions. To ease the assessment of the robot actions by the user and for visualization purposes, each target was depicted as an icon of a different city. The robot moved in the following way. First, it executed a pure rotation motion to orientate the robot towards a desired direction (i.e. towards a goal). Then, it followed a straight line to the desired position. Despite the goal positions were discrete, the possible states and actions of the robot were continuous. In order to obtain a robust measure of the robot position, the robot was visually tracked in real time with a camera located on the ceiling.

The main difference with the previous protocol was that the robot moved continuously and the user constantly evaluated the robot actions. As long as the decoder did not detect an error, the robot continued its motion to the selected goal. The robot stopped for a second after reaching a goal or detecting an error. Then, it moved towards a new goal selected based on the probabilities of each target. The user was asked to look over the robot actions, evaluating them as correct when the robot advanced or turned towards the goal, and when the robot stopped over the desired goal position. On the contrary, the user had to evaluate as incorrect those motions that were not oriented towards the goal, when the robot stopped on a wrong spot, or when the robot overpassed the desired position or orientation. Currently, one subject (age 28) has performed the experiment. The user was seated one meter away from the map (see Figure 5a), and was instructed not to move his eyes or blink during the robot movements.

For this experiment, the calibration phase required two steps, one to acquire error and another for non-error responses. In each step, the user had to evaluate the robot actions towards five predefined goals and he had to push a button when an error occurred (step one) or when the robot was executing a correct action (step two), always with separations of at least one second between two trigger events. These runs were repeated until acquiring around 70 examples of each class. The calibration phase lasted a total of 30 minutes. During the control phase, the user freely chose the initial and goal locations (see Figure 5b-c).

5.2 Shared-control strategy

Let us encode each possible state as the position and orientation of the robot, $\mathbf{s} = (u, v, \theta)$, and each action as the combination of a turn and a linear movement $\mathbf{a} = (\theta_a, \rho)$ represented by the angle θ and distance ρ. In this case, we use a potential $U_i(u, v)$ to define the optimal policy for target i, ignoring the non-holonomic constraints of the robot. We used the symmetric 2D quadratic function:

$$U_i(u, v) = [u, v]' A_i [u, v] + b_i' [u, v] + c_i, \quad (5)$$

where A_i, b_i and c_i depend on the position of target i and the size of the map.

The likelihood function was computed differently depending on the action step (rotation or linear movement). While turning, the

Table 1: Results of the reaching task for the fixed and free goals

	S1		S2		S3		S4		$\mu \pm \sigma$	
	FIXED	FREE	FIXED	FREE	FIXED	FREE	FIXED	FREE	FIXED	FREE
♯ TARGETS REACHED (OUT OF 5)	5	5	5	5	5	5	5	5	5 ± 0	5 ± 0
♯ ACTIONS	16 ± 2	23 ± 9	43 ± 9	21 ± 7	23 ± 12	16 ± 6	17 ± 5	23 ± 11	25 ± 13	21 ± 8
NET TIME (S)	12.96 ± 1.91	18.08 ± 7.37	34.56 ± 7.10	16.48 ± 5.84	18.40 ± 9.73	12.96 ± 5.10	13.60 ± 4.38	18.08 ± 8.44	19.88 ± 10.75	16.40 ± 6.61
TOTAL TIME (S)	52.65 ± 7.76	73.45 ± 29.93	140.40 ± 28.83	66.95 ± 23.73	74.75 ± 39.54	52.65 ± 20.73	55.25 ± 17.80	73.45 ± 34.29	80.76 ± 73.68	66.63 ± 26.85
MEAN ACCURACY (%)	83.14 ± 15.15	78.75 ± 12.62	69.49 ± 4.13	82.48 ± 10.66	72.49 ± 3.74	88.18 ± 10.51	76.35 ± 8.89	78.66 ± 14.44	74.38 ± 4.66	77.67 ± 5.02

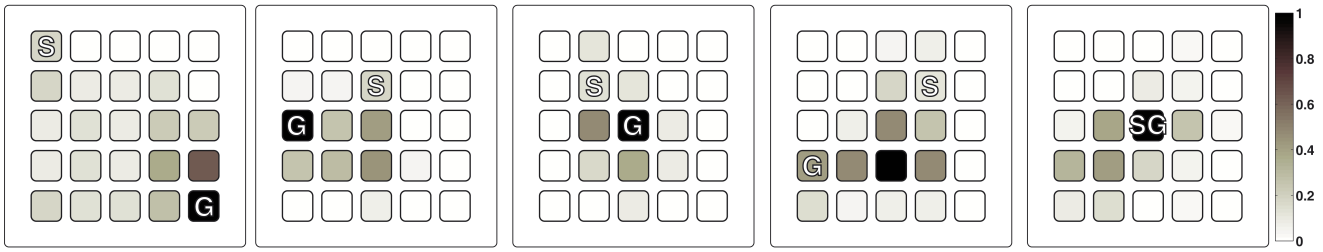

Figure 4: States visited by all the subjects, for each of the five runs executed with the fixed goals (from left to right, runs 1 to 5). Darker colors indicate more visited states. The range was normalized from 0 to 1 according to the most visited state for each run. The initial and goal positions are marked with an S and a G respectively.

likelihood was computed as a piecewise function:

$$p(\mathbf{a}_t|\pi_i^*,(\mathbf{s},\mathbf{x})_t) = \begin{cases} k_n & \text{if } (p(c_t=1|\mathbf{x}_t) \geq T_e) \wedge (\theta_t - \theta_{t-1} > 0) \wedge (\theta_i - \theta_t \in (\ \ 0, \pi\ ]), \\ k_n & \text{if } (p(c_t=1|\mathbf{x}_t) \geq T_e) \wedge (\theta_t - \theta_{t-1} < 0) \wedge (\theta_i - \theta_t \in (-\pi,\ 0\ ]), \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

$k_n < 1$ is a penalization constant, fixed to 0.2 for the performed experiments; and $(\theta_i - \theta_t)$ is the relative angle between goal i and the robot state $\mathbf{s}_t$. The three boolean conditions of the first two pieces of the likelihood describe: (i) the output of the classifier was considered an error based on a threshold $T_e \in [0, 1]$. Since we wanted to minimize the number of false positives (correct assessments detected as errors), we fixed this threshold to a high value, $T_e = 0.8$; (ii) the robot is turning clockwise or anti-clockwise; and (iii) the goal is located left or right relative to the current robot position and orientation. Intuitively, if an error was detected, this likelihood simply penalized those targets where the robot was turning to; on the contrary, no changes were made on the policies when the user's assessments were detected as correct.

For the linear movement step, the likelihood was computed as follows:

$$p(\mathbf{a}_t|\pi_i^*,(\mathbf{s},\mathbf{x})_t) = \begin{cases} 1 + k_p \cdot \mathcal{N}(\theta_t - \theta_i\ ;\ 0, \sigma) & \text{if } (p(c_t=1|\mathbf{x}_t) < T_e), \\ 1 - k_n \cdot \mathcal{N}(\theta_t - \theta_i\ ;\ 0, \sigma) & \text{if } (p(c_t=1|\mathbf{x}_t) \geq T_e) \end{cases} \tag{7}$$

The first piece corresponds to a correct user's assessment and assigns a higher likelihood to goals in front of the robot ($k_p = 0.01$). The second one is applied when an error is detected and assigns a lower likelihood to targets in front of the robot ($k_n = 0.7$). We modeled the uncertainty in the user's perception of directions with a normal probability distribution with zero mean and standard deviation σ, fixed to have a field of view of $\pm$ 20 degrees. The difference between k_p and k_n reflects the fact that number of detected errors should be lower than the number of correct actions.

The next action was selected greedily as the optimal policy according to the potential function $U_i(u, v)$ of the target with the higher probability at that point in time. This basically rotated the robot to align it with the direction of the gradient of $U_i(u, v)$ and then moved forward to the target. The run finished when reaching a convergence criterion of $p_c = 0.4$.

5.3 Preliminary results

Figure 5(b-c) shows the two trajectories resulting from controlling the mobile robot. The time elapsed from the start of the movement until goal reaching of each trajectory was 60 and 121 seconds respectively, counting up to 11 error events in the first run, and 26 in the second. The performed trajectories revealed some of the properties of the proposed protocol: (i) most of the errors were concentrated during turns. This allowed the robot to perform mostly long straight paths towards the believed goal location; (ii) as no errors are detected, the robot maintains a fixed trajectory, as can be seen on the subpath from Las Vegas to Pisa (see Figure 5b); and (iii) the system can recover from false positives. For instance, during the second run the robot chose to go from Beijing to Tokyo (see Figure 5c) but an error was detected. This made the robot deviate towards other goals (Cairo and Berlin), but in the end it reached the desired position.

6. CONCLUSIONS

In this paper, we have presented an alternative way of giving human feedback to virtual and real devices, extracted directly from the user's brain signals. In order to cope with the limited information provided by these signals, a shared control strategy based on an inverse reinforcement learning framework was used to maintain a belief over possible targets, updating them according to the user's assessments. The use of this shared control allowed reaching the target in a 5x5 grid world after 23 actions on average (less than one minute of EEG). Furthermore, the preliminary results obtained in real environments with a mobile robot were very promising, suggesting that it is possible to constantly determine the user's assessments while learning a task. The proposed shared-control BCI might scale to more complex scenarios because: (i) it is not necessary to explore every single trajectory or potential goal; and (ii) the user only has to monitor the device actions and evaluate if they are right or wrong.

The promising results on real robots require more work to understand and characterize error potentials that appear in the absence of a clear cue and prevent the detector to use time-locked signals. Also, additional experiments have to be conducted with

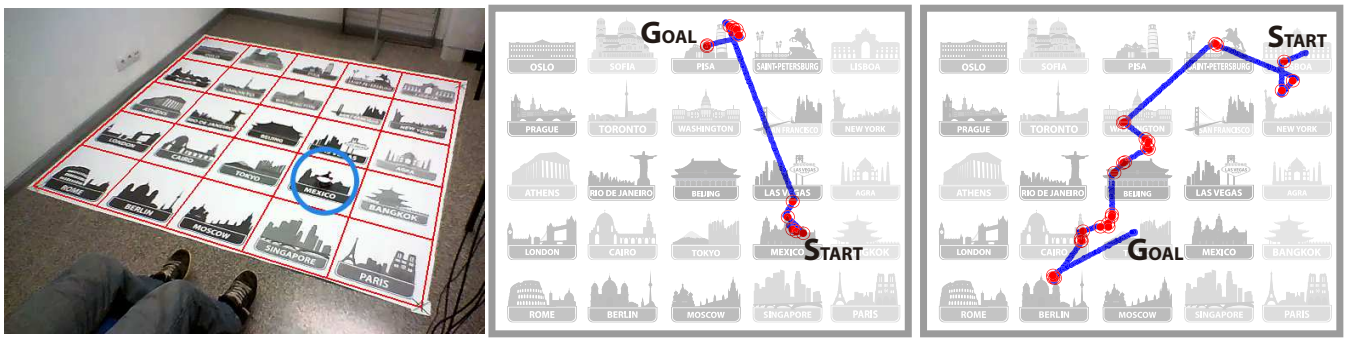

Figure 5: (a) Snapshot of the experiment performed, together with the grid superimposed to the image. The mobile robot location is marked with a circle. (b-c) Trajectories performed by the robot (marked in blue) during the two online runs. The initial and goal positions were from (b) Mexico to Pisa and (c) Lisbon to Tokyo. Each red mark indicates the moment when an error was detected from the EEG signal.

more subjects in order to confirm the results presented here. Nevertheless, there are some interesting research directions. For instance, it is possible to use more intelligent exploration strategies than the greedy one to infer the users' intended target. Also, we believe that this type of feedback will be very useful in application related to neurorehabilitation or neuroprosthetics, since the device can use this feedback to adapt its trajectories to the user preferences in a transparent way.

7. REFERENCES

[1] A. Austermann and S. Yamada. Learning to understand multimodal rewards for human-robot-interaction using hidden markov models and classical conditioning. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 4096–4103, 2008.

[2] C. Bishop. *Pattern recognition and machine learning*. Springer, 2011.

[3] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K. Müller. Single-Trial Analysis and Classification of ERP Components-a Tutorial. *NeuroImage*, 2010.

[4] J. Cavanagh, M. Cohen, and J. Allen. Prelude to and resolution of an error: EEG phase synchrony reveals cognitive control dynamics during action monitoring. *Journal of Neuroscience*, 29(1):98–105, Jan. 2009.

[5] R. Chavarriaga and J. Millán. Learning from EEG error-related potentials in noninvasive brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(4):381–388, 2010.

[6] R. Croft and R. Barry. EOG correction of blinks with saccade coefficients: a test and revision of the aligned-artefact average solution. *Clinical neurophysiology*, 111(3):444–51, Mar. 2000.

[7] M. Falkenstein, J. Hoormann, S. Christ, and J. Hohnsbein. ERP components on reaction errors and their functional significance: A tutorial. *Biological Psychology*, 51:87–107, 2000.

[8] P. Ferrez and J. Millán. Error-related EEG potentials generated during simulated Brain-Computer interaction. *IEEE Transactions on Biomedical Engineering*, 55(3):923–929, March 2008.

[9] C. Isbell, C. Shelton, M. Kearns, and a. P. S. S. Singh. Cobot: A social reinforcement learning agent. In *Proceedings of the 5th Intern. Conf. on Autonomous Agents*, 2001.

[10] I. Iturrate, L. Montesano, and J. Minguez. Single trial recognition of error-related potentials during observation of robot operation. In *Conf Proc IEEE Eng Med Biol Soc (EMBC), Boston, USA*, 2010.

[11] I. Iturrate, L. Montesano, and J. Minguez. Task-dependent signal variations in eeg error-related potentials for brain-computer interfaces. *Journal of Neural Engineering*, 10(2):026024, 2013.

[12] W. B. Knox and P. Stone. Reinforcement learning from simultaneous human and MDP reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, June 2012.

[13] S. Luck. *An introduction to the event-related potential technique*. The MIT Press, 2005.

[14] J. Millán et al. Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges. *Frontiers in Neuroscience*, 4, 2010.

[15] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65, 2009.

[16] A. Schlögl et al. A fully automated correction method of EOG artifacts in EEG recordings. *Clinical neurophysiology*, 118(1):98–104, Jan. 2007.

[17] R. Sutton and A. Barto. *Reinforcement learning: An introduction*. MIT Press, 1998.

[18] A. L. Thomaz and C. Breazeal. Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, AAAI'06, pages 1000–1005. AAAI Press, 2006.

[19] P. S. WB Knox. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge*, 2009.

[20] P. S. WB Knox. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multi-Agent Systems*, 2010.