2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

BMI-based Framework for Teaching and Evaluating Robot Skills

Christian I. Penaloza[1], Yasushi Mae[1], Masaru Kojima[1] and Tatsuo Arai[1]

***Abstract*— Brain Machine Interface systems provide ways of communication and control of a variety of devices that range from domestic appliances to humanoid robots. Most BMI systems are designed exclusively to control devices using low-level commands, or high-level commands when devices with pre-programmed functionalities are available. In this paper, we build on our previous work on BMI-based Learning System in which we presented a different approach for designing BMI systems that incorporate learning capabilities that relieve the user from tedious low-level control. In this work, we extend the capabilities of our framework to allow a user to be able to teach and evaluate a robotic system by using a BMI. We provide general system architecture and demonstrate its applicability in new domains such as teaching a humanoid robot object manipulation skills and evaluating its performance. Our approach consists of 1) tele-operating robot's actions while robot's camera collects object's visual properties, 2) learning manipulation skills (i.e. push-left, lift-up, etc.) by approximating a posterior probability of commonly performed actions when observing similar properties, and 3) evaluating robot's performance by considering brain-based error perception of the human while he/she passively observes the robot performing the learned skill. This technique consists of monitoring EEG signals to detect a brain potential called error related negativity (ERN) that spontaneously occurs when the user perceives an error made by the robot. By using human error perception, we demonstrate that it is possible to evaluate robot actions and provide feedback to improve its learning performance. We present results from five human subjects who successfully used our framework to teach a humanoid robot how to manipulate diverse objects, and evaluate robot skills by visual observation.**

I. INTRODUCTION

Advances in neuroscience and robotics have made possible diverse robot control applications using a brain machine interface to assist patients with devastating motor paralysis conditions [1]. By controlling home appliances [2], a robotic wheelchair [3], [4], or even a humanoid robot [5], BMI systems allow patients to regain certain degree of controllability and independence.

Non-invasive BMI systems commonly use electroencephalogram (EEG) signals to train a classifier to detect changes of brain activity and translate them into robot commands. This is usually achieved by one of two common approaches: event related potential (ERP) detection or band power-based classification. ERP is produced when the brain is stimulated by a target of interest through flashing visual stimulus at different frequencies, such as in the case of P300 [6] for typing in a virtual keyboard. On the other hand, band power-based methods attempt to interpret user's thoughts by classifying specific motor images through the power over the frequency range [7], [8]. In both cases, patients are required to spend long periods of time learning to generate appropriate mental states or focusing on a computer screen, which may cause mental fatigue, frustration or discomfort [9].

To deal with this problem, shared control systems have been proposed in which the user selects a particular task and the robot performs the task using pre-programmed functionalities [10]. The user can benefit from using high level commands, but the robots still are lacking intelligent capabilities to adapt to new situations, remember user preferences or predict future user commands. More recently, researchers have proposed more intelligent systems that combine shared control with learning capabilities to navigate [11] or manipulate an object [12] while they are being controlled via BMI. In our most recent work [2], we presented a BMI-based domestic appliance control system that gradually becomes autonomous by learning user actions (i.e. turning on/off window, lights, etc.) under certain environmental conditions (i.e. temperature, illumination, etc.), and brain states (i.e. awake, sleepy, etc.). Our proposed approach for designing BMI framework does not require the device to have pre-programmed actions for pre-defined conditions.

In the current work our major contribution is to demonstrate how our framework can be extended to other applications such as for teaching a humanoid robot object manipulation skills. Moreover, we implement a new feature that consists of using brain-based error related negativity signals to evaluate robot learning performance. In overall, our framework can be used to teach and evaluate a robot by 1) tele-operating robot's actions while robot's camera collects object's visual properties, 2) learning manipulation skills (i.e. push-left, lift-up, etc.) by approximating a posterior probability of commonly performed actions when observing similar properties, and 3) evaluating robot's performance by considering brain-based error perception of the human while he/she passively observes the robot performing the learned skill. By using human error perception, we demonstrate that it is possible to provide feedback to the robot in order to improve its learning performance.

II. LITERATURE REVIEW

In order to relieve the user from constant BMI operation using low level commands to control a robot, researchers have proposed share control systems in which the user selects a task and the robot operates semi-autonomously with pre-programmed actions [10]. F. Lotte *et al.*, for example, devel-

*This work was supported in part by NSK Foundation for the Advancement of Mechatronics.

[1]C. Penaloza, Y. Mae, M. Kojima and T. Arai are with the Graduate School of Engineering Science, Osaka University. 1-3 Machikaneyama-cho, Toyonaka, Osaka, Japan. penaloza,mae,kojima,arai at arai-lab.sys.es.osaka-u.ac.jp

oped an application to explore a virtual museum with EEG-based BMI, in which they provide two types of navigation: Free navigation mode (forward-left-right commands) and assisted navigation based on *points of interest*, namely pre-programmed "shortcuts" that when selected provide automatic navigation [13]. While systems like these may reduce constant operability, they still lack of intelligent capabilities that make them adaptable to new situations and provide better assistance to the user.

More recently, intelligent systems where the user uses a BMI to teach a robot new skills that can later be invoked using high-level commands have been developed. M. Chung *et al.* [11] presented a hierarchical BCI (HBCI) in which users use low-level movement-by-movement commands to teach a robot to navigate a path which can later be executed with a single command. M. Bryan, *et al.* [12] also used a hierarchical approach to maneuver a robot's arm to manipulate an object, and then learn the sequence of commands using grammar-based structure [5]. In our previous work [2], not only we taught a robotic system (home appliances) via tele-operation, but we also incorporated external sensor information to learn user preferences under different environmental conditions, and make the system adaptable to new situations. Although significant progress has been made to develop intelligent BMI systems, until now not much work has been done in considering user feedback to evaluate the robot and potentially improve its learning performance.

In this paper, we also take into consideration user feedback based on error perception in order to evaluate robot's performance. In particular we use an event related potential called error related negativity (ERN) that is characterized by a negative deflection in the EEG signal that appears from 80-150ms after the person perceives erroneous actions [14]. ERN responses have been identified when the subject makes a mistake, or when the subject observes a mistake made by another person or a machine [15]. A number of studies have proposed ERN as a way to improve the performance of BCIs [16], but most of these studies propose the use of ERN following an error made by the subject himself when using an interface. More recently a few researchers have started to use ERN for providing feedback to an external agent. Iturrate I. *et al.* [17], for example, used ERN to provide rewards to a model-based reinforcement learning algorithm in a simulation platform, and Chavarriaga R. *et al.* [18] used ERN to monitor the behavior of an intelligent agent to minimize erroneous actions. Their research demonstrates that ERN can effectively be used to monitor an agent's actions and provide error feedback when the correct actions are *pre-defined*.

In our work, we do not pre-define correct or incorrect robot actions. Instead, we allow the user to teach a robot certain actions to manipulate objects, and evaluate robot's performance using user's own criteria. We consider this is important because each user may teach a robot in a different way according to his/her preferences. For instance, some users may teach a robot to prepare a cup of coffee by serving coffee powder first and then water, while other users may prefer it the other way around. For this reason we consider that evaluation should be performed by the same teacher. In overall, in this work we demonstrate that our framework can be used for both: 1) teaching a robot new skills based on teacher's preferences, and 2) evaluating and potentially improve robot's learning performance by considering error feedback from the user based on ERN responses.

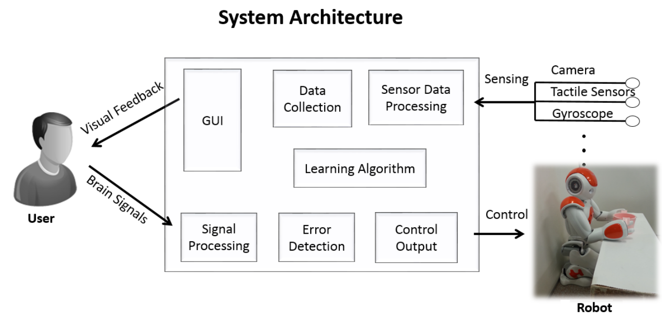

Fig. 1. System architecture. User receives visual feedback from robot sensors (i.e. camera, tactile sensors, etc.) and then controls robot actions using brain signals while observing a graphic user interface.

III. SYSTEM ARCHITECTURE

The proposed system shown in Fig. 1 describes an architecture in which a user receives visual feedback from robot sensors (i.e. camera, tactile sensors, etc.) and then controls robot actions using brain signals while observing a graphic user interface. The framework features two operation modes: teaching and evaluation mode. Different from our previous work [2], the current system architecture is composed of 7 modules described as follows:

- *Graphic user interface (GUI)* - provides visualization of three items:
  - Robot control menu: hierarchical menu based on robot actions and low-level commands.
  - Robot head camera view: allows the user to see what robot sees.
  - Scene camera view: allows the user to see what the robot is doing from an external point of view.
- *Sensor Data Processing module* - receives data from robot's sensors such as tactile sensors, gyroscope and camera.
- *Data collection module* - constructs a training dataset of the correlation of sensor data and actions.
- *Learning algorithm* - uses a Bayes Point Machine (BPM) learning approach given a training set.
- *Control output module* - sends action commands to the physical robot.
- *Signal Processing* - Receives raw EEG data from the electrodes and performs proper processing to be used by GUI and Error detection modules.
- *Error detection module* - monitors EEG signal and detects ERN potentials when the user perceives an error made by the robot. This module is only enabled during evaluation mode.

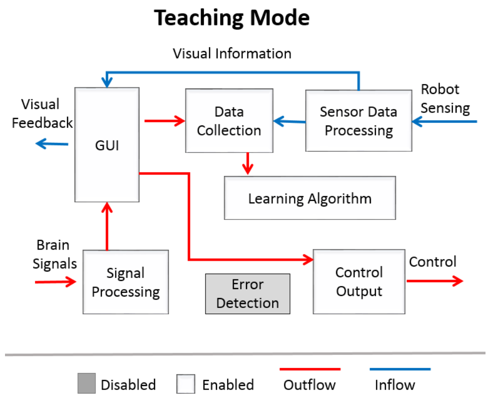

Fig. 2. Teaching mode:. Robot's camera view is displayed in the GUI. The user assigns brain-signal based action commands to the robot. At the same time, a training dataset is constructed with the correlation of sensor data and actions.

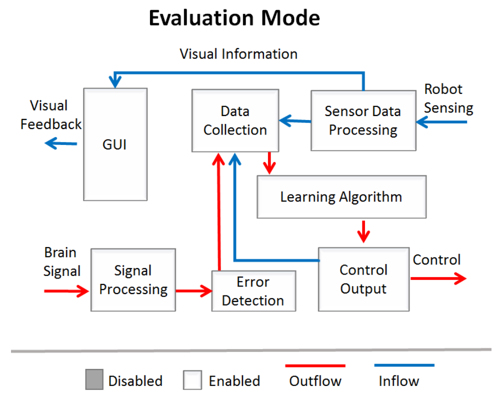

Fig. 3. Evaluation mode: Learning algorithm module autonomously decides robot actions based on input sensor data. At the same time, EEG data is monitored by the error detection module in order to detect mistakes made by the robot while the user is observing.

A. Teaching Mode

During teaching mode, user sees robot's camera view through the GUI, and navigates a hierarchical menu to select actions to be sent to the robot. We used a graphic interface for controlling diverse robotic platforms using BMI that we previously proposed in [19]. This interface employs a hierarchical menu structure that allow the user to navigate through different menus to select robot actions (i.e. walk, move_head) and corresponding commands (forward, turn_head_left, etc.). We modified the interface to operate the Aldebaran Robotics NAO robot [20] as described more in detail in section IV-A.

Figure 2 shows a representation of the way data flows throughout the framework during teaching mode. Sensor Data processing module receives sensing data from robot's sensors and camera, and transmits it to the Data Collection module and the GUI to be graphically displayed. The user observes the GUI and assigns brain-based commands that are collected by electrodes placed on the user's scalp. Raw EEG signal data is received by the Signal Processing module where it is processed to identify control commands to select buttons in the GUI. Actions selected in the GUI are transferred directly to the Control Output module that sends wireless commands to the physical robot, but also they are sent to the Data Collection module that constructs a training dataset with the correlation of sensor data and actions. Lastly, the Learning Algorithm module uses training set to learn the posterior probability of commonly performed actions when encountering similar sensing data.

Although a general framework description is provided to be used in different applications, for our experiment we designed a teaching task in which a robot is presented with objects that have different properties and it needs to learn to manipulate objects based on their properties. In order to represent object properties we used color markers that can be recognized through a vision algorithm implemented within the Sensor Data Processing module. The user can observe the object through the GUI and teach a manipulation skill (push_left, lift_up, etc.) to the robot, as described more in detail in section VI.

B. Evaluation Mode

Figure 3 shows a representation of the way data flows throughout the framework during evaluation mode. In this mode, robot actions are autonomously decided by the Learning Algorithm module depending on the sensor information obtained from robot's camera and sensors. The learning algorithm employed is based on the Bayes Point Machine (BPM) which is a Bayesian network used in our previous work [2], and briefly explained in section V.

In the case of our application, when a new object is presented to the robot, object properties are recognized by the Sensor Data Processing module and the Learning Algorithm module assigns the action with the highest probability score to manipulate the object while the user observes the robot through the GUI. During this mode, the EEG data from the user is collected by the Signal Processing module and monitored by the Error Detection module. In case the user perceives a mistake made by the robot, this module triggers a signal to the Data Collection module to record current sensing information and previously emitted action command. This application considers a finite set of possible actions, so in case an error is detected, the emitted action is flagged as erroneous and the next action with the highest probability score among non-flagged actions is assigned and stored in the dataset as a new training sample. Finally, BMP learning algorithm module is retrained using newly collected data.

IV. BMI SIGNAL DETECTION

The core system functionality depends on signal processing and detection in order to 1) select buttons of a GUI, and 2) detect mistakes made by the robot during an object manipulation task.

In the case of selecting buttons of a GUI, common approaches involve the use of ERP such as P300 [6] or SSVEP [12]. Other researchers such as Ferreira *et al.* have employed electromyogram signals (EMG) based on eye-blinks which are easier to detect by a classifier [21]. Although, ERP based approaches for clicking a button in a interface are totally feasible for our application, for experimental purposes we

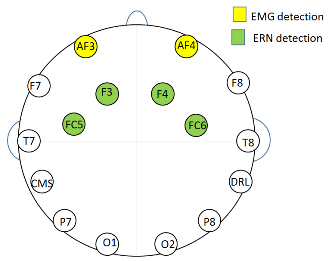

Fig. 4. Electrode locations marked in yellow color were used to detect EMG signals, and electrodes marked in green were used to detect ERN.

decided to use EMG signals based on eyebrow movement that avoids the confusion between detecting voluntarily eye-blinks and involuntarily eye-blinks (those that occur for eye lubrication). For detailed description of EMG detection process please refer our previous work in [2].

On the other hand, ERN detection was used to detect the mistakes made by the robot during an object manipulation task while the user passively observed the robot. A background of ERN along with the procedure we implemented for its detection is detailed in subsection IV-B.

The device used for signal acquisition is a high resolution multichannel Emotiv EPOC neuroheadset [22]. Emotiv EPOC measures brain activity and transmits EEG data to a USB dongle connected to the PC. It has 14 channels (plus CSM/DRL references) whose placement is based on the international 10-20 locations: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, and AF4. It is able to detect slow cortical potentials, EEG oscillations in the alpha and beta band, and P300 responses. Figure 4 shows electrode locations used to detect EMG (yellow color) and ERN signals (green color).

A. Selecting Robot Actions using EMG signals

We used an interface that we previously presented in [19] in which we employed a hierarchical menu structure that allows the user to navigate through different menus to select robot actions and corresponding commands. We modified the interface to be able to control NAO robot and included video windows to display robot's camera view and external view from a camera looking at robot. We also added menus with buttons to select the following actions *move_left_arm, move_right_arm, move_both_arms* and low-level commands: *push_left, push_right, lift*. We decided to use a small number of actions-commands to facilitate the experimental task to the user, but certainly a larger number could be implemented.

During teaching mode, the user is initially presented with 'Actions' menu. In order to allow the user to select a particular option, the interface is designed to highlight each of the options in sequential order every 1.5 sec. The user only has to send an EMG signal by performing eyebrow movement while the desired option is highlighted. After selecting a particular action, the 'Command Menu' is prompted and the user can then select a desired command.

B. Detecting Error Related Negativity (ERN)

In neuroscience it is well known that ERN is the result of the underlying mechanism of human error processing. ERN is characterized by a negative deflection appearing from 80-150ms after the person perceives erroneous actions [14]. ERN has been identified in several cases such as: 1) when the subject realizes that he committed an error, 2) when the subject receives feedback when he committed an error, or 3) when the subject perceives an error committed by other person or inclusive a machine [18]. The amplitude of ERN is large when the user clearly perceives the error and small when the user is confused or when he/she was not aware of the error [23]. This means that appropriate ERN detection does not depend on the observed erroneous behavior itself but in the user's clear perception of it.

Different methods for detecting ERN have been proposed in the past. One of the most common approaches is offline averaging over multiple trials in which experiment participants are asked to perform multiple choice tasks. Each task has pre-determined erroneous responses that are randomly triggered when the user selects a particular option. By using an erroneous behavior when the user expects a different result, it is possible to clearly identify ERN [23]. More recent approaches have achieved online single trial detection in which ERN is detected while the user performs a task [17], [24]. Single trial detection consists in pre-training a classifier that can be used to identify ERN in online trials. In order to train the classifier, a preliminary experiment is conducted to collect samples of EEG signals during tasks that involve error perception.

In order to achieve ERN detection, we implemented a single trial detection approach in which positive and negative ERN samples were collected using the Flanker Task that has previously been used in similar experiments [23].

1) EEG Data Collection using Flanker Task: The Flanker Task consists in having the user to press one of two mouse buttons (left or right) to specify the direction of a central symbol that appears within a sequence of characters that are flashed in random order. Two types of sequence characters were used: same direction ({{{{{{{ and }}}}}}}) and contrary direction (}}}{}}} and {{{}{{{). For each trial, each sequence of characters was flashed in the screen for 100ms and then disappeared for 2000ms until a different sequence of characters appeared again. The participant had to click the corresponding mouse button before the next sequence of characters appeared, making the user prone to make mistakes and be aware of the mistake made.

Five participants: 3 male, 2 female (ages M=27.2, STD=2.68) participated in the Flanker Task. Each participant was seated in front of the screen and was asked to minimize eye blinking in order to reduce noise in data collection while wearing the headset. Each participant performed 100 trials, giving a total of 500 trials from all participants. EEG signal samplings were collected for a duration of 1000ms directly after the participant clicked the button. Figure 5 shows EEG signal sampling in the case the participant clicked the correct

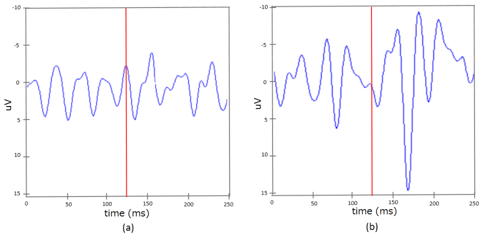

Fig. 5. Examples of ERN during: a) correct and b) incorrect responses of the user. The red line indicates the moment when the user pressed the button.

TABLE I
ERN DETECTION PERFORMANCE

	$P1$	$P2$	$P3$	$P4$	$P5$	Avg
SVM	84.63%	82.41%	81.12%	84.52%	77.20%	**81.97%**
LDA	82.62%	78.21%	79.33%	82.63%	75.45%	**79.64%**

button (a) and when he realized he made a mistake (b).

2) ERN classification: According to neuroscience literature, ERN originate in the fronto-central area close to the Anterior Cingulate Cortex (ACC) [14]. This fact matches our visual inspection of EEG data obtained from electrodes F3, F4, FC5 and FC6 during Flanker Task. Although there is also significant activity in other channels, it could be related to other cognitive processes and thus we do not take it into consideration.

We then use EEG data from F3, F4, FC5 and FC6 (Fig. 4) to perform feature extraction and classification as follows:

- For each trial, consider a time window of 200ms **after** the button was pressed.
- Construct a feature vector by concatenating data from selected channels.
- Normalize features on the range 0-1.
- Assign a class label (non-ERN / ERN).
- Train linear classifiers.

We used EEG data from correct responses as negative training samples (non-ERN) and EEG data from incorrect responses as positive training samples (ERN) to train both a support vector machine (SVM) and linear discriminant analysis (LDA) classifiers. For each participant, EEG data from 75 trials was used for training, and the remaining 25 trials for evaluation. ERN detection performance for both: SVM and LDA are shown in Table I.

Results show an average detection rate of roughly 80% for both classifiers, demonstrating that it is feasible to detect ERN responses when the participant perceives an error. The final ERN detection algorithm employed in our system consists of a sensor fusion scheme 2oo2 (two out of two) - both classifiers need to agree - to minimize false positives.

V. LEARNING ALGORITHM

The learning algorithm employed in our system is a two layer Bayesian Network that encodes the probability distribution of multiple variables using Bayes Point Machine [25]. We then use the Expectation Propagation method devised by [26] to perform Bayesian inference. In this work, we used a similar approach presented in our previous work and implementation details can be found in [2].

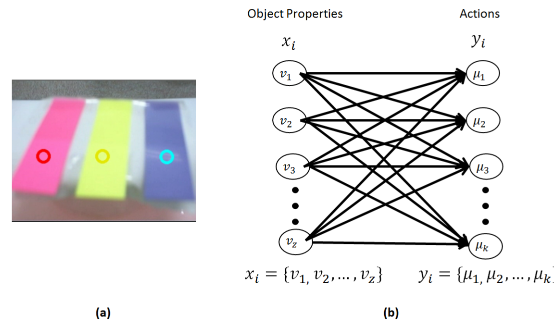

Fig. 6. (a) Robot's camera view: each object had color markers that represented the object's properties. A multi-color detection algorithm was used to identify object properties (b) Bayesian Network

For our application, the algorithm was used to approximate a posterior probability of commonly performed actions for observed object properties presented during teaching mode. Although complex computer vision algorithms can be used to identify properties such as shape, size, orientation, grasping points, etc., we decided to simplify this task by placing color markers in the objects. We implemented multi-color detection algorithm to identify the different properties (Fig. 6a) that were used to construct network input data. In order to define network outputs, we labeled robot actions using the combination of action-commands selected by the user within the GUI described in section IV-A. For example, the option *move_left_arm* from the 'Action' menu would be combined with the option *push_right* from the 'Commands' menu as the new action *push_right_with_left_arm*, and so on.

In overall, the network considers input data for a particular object x_i with z properties v, $x_i = \{v_1^i, v_2^i, \ldots, v_z^i\}$ to assign a posterior probability to each action in set $y_i = \{\mu_1^i, \mu_2^i, \ldots, \mu_k^i\}$ where k is the total number of possible actions as shown in Fig. 6b. During teaching mode, each property v was assigned a binary state depending on whether the color assigned to that property was detected '1' or not '0'. In the same manner, binary states were assigned to output nodes μ depending on whether the action-command was selected '1' or not '0'. During evaluation mode, after Bayesian inference was performed, the action μ with the highest probability score was performed by the robot.

VI. EXPERIMENT

Experiments were conducted to evaluate 1) robot's learning performance after it was taught object manipulation skills, 2) error rate when considering user perception error feedback during evaluation, and 3) error rate after detected mistakes were flagged out and highest probability score among non-flagged actions was re-assigned into the training set. The experiment was conducted in two parts: teaching and evaluation. Five participants: 3 male, 2 female (ages M=27.2, STD=2.68) participated in the experiment that lasted approximately 60 minutes per participant.

The experimental setup consisted of a robot standing behind a stationary table and a camera located in front of the robot, as shown in Fig. 7a Each participant was seated

Fig. 7. (a) Experimental setup: robot was located behind a stationary table and a camera was located in front of the robot. (b) Objects with color markers that were presented to the robot in random order.

TABLE II
PROPERTIES - ACTIONS

Properties: *Blue, Green, Yellow, Pink, Purple*
Actions: *push_left_with_left_arm, push_right_with_left_arm, push_left_with_right_arm, push_right_with_right_arm, push_left_with_both_arms, push_right_with_both_arms, lift_with_left_arm, lift_with_right_arm, lift_with_both_arms.*

in front of a computer screen and was assisted to wear the EEG headset. The participant was not able to directly see the physical robot, but was able to see robot's camera view and external camera view through the GUI. Participant was given instructions of how to operate the interface by performing eyebrow movement in order to select a button of the menu.

3) Teaching mode: During teaching mode, five objects with one or more color markers, shown in Fig. 7b, were presented in front of the robot one at a time in random order. Participants were not told the meaning of the color markers but instead they were instructed to look at the physical properties of the object and perform a preferred action. Rather than having the robot learn one particular action per object, we considered that learning the relationship of object properties-actions is more realistic, given the fact that a particular object can be manipulated in multiple ways depending of its properties. For example, it might be more appropriate to *lift* a cup that has a left handle using robot's *left arm*, while a cup with two handles could be lifted with either *left arm, right arm, or both arms*. For this reason, objects' physical properties were assigned colors and objects could share color markers. The list of properties and possible actions are presented in Table II. We define a *trial* as the event in which the robot is presented with an object and the user tele-operates the robot to perform a manipulation action. Each trial lasted 30-40secs depending on the participant's ability to control the interface.

4) Evaluation mode: During evaluation mode, participants were asked to passively observe the object randomly presented to the robot without performing any face/head movement. A timer set to 5 seconds appeared on the screen and started to count down. When the timer was up, object markers were detected and corresponding properties were assigned to the BPM network input layer. The action with the highest probability score inferred from previous training dataset was performed by the robot. ERN detection was conducted while the participant observed the action being performed by the robot. In case an error was detected for a

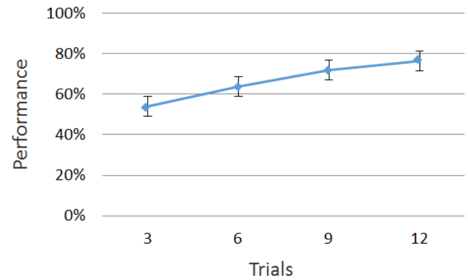

Fig. 8. Mean learning performance for 5 participants. A gradual increment can be noticed from an initial ∼45% after 3 trials to ∼75% after 12 trials.

particular action, the action was flagged as 'erroneous' and the action with the highest probability score among non-flagged actions was re-assigned into the training set. Finally, the robot was re-trained with the new training set.

A. Data Collection and Evaluation

During teaching mode, a total of 36 trials were conducted. Each trial resulted on a set of input object properties x and output action y, where $x \rightarrow y$. The total number of trials was split in three sets; each containing 12 trials: *TrainingSet-1* (X_{train_1}&Y_{train_1}), *EvaluationSet-1* (X_{eval_1} & Y_{eval_1}) and *EvaluationSet-2* (X_{eval_2} & Y_{eval_2}). *TrainingSet-1* was used to initially train the robot after teaching was conducted.

During evaluation, X_{eval_1} was used as input properties to test robot actions while ERN detection was conducted on the participant. We collected a resulting set of actions Y_{result_1} that correspond to the actions inferred from *TrainingSet-1*. We then generated a second set (Y_{result_2}) after incorrect actions were flagged as erroneous and the action with the highest probability score among non-flagged actions was re-assigned. The new training set was defined as *TrainingSet-2* (X_{eval_1}&Y_{result_2}).

In order to visualize learning performance, the learning algorithm was gradually trained using data from 3 trials at a time. Initial performance evaluation (when the robot was trained using *TrainingSet-1*) was performed by comparing Y_{result_1} to true data Y_{eval_1}, and defining accuracy as learning performance. For the final evaluation we trained the system using 1) *TrainingSet-1*, and 2) *TrainingSet-1* + *TrainingSet-2*. X_{eval_2} was used as input properties, giving as a result two sets of corresponding results: $Y_{result_{3-1}}$ and $Y_{result_{3-2}}$. In the same manner, each of these results was compared to true data Y_{eval_2} and error rate was computed based on the number of mistakes obtained.

B. Results

Figure 8 illustrates the mean learning performance for all participants, after robot was trained using *TrainingSet-1* and evaluated using *EvaluationSet-1*. A gradual increment can be noticed from an initial ∼45% after 3 trials to ∼75% after 12 trials. Although a higher learning performance was expected, the fact that the number of possible actions outnumbered objects' properties might be a reason why longer training would be needed to achieve a higher performance.

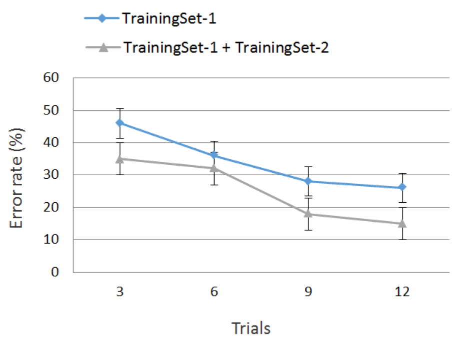

Fig. 9. Mean error rate for all participants after the robot was trained using *TrainingSet-1* and *TrainingSet-1* + *TrainingSet-2*, and evaluated using *EvaluationSet-2*. When training the robot with *TrainingSet-1* + *TrainingSet-2*, it can be noticed that error rate decreased - from an initial ∼35% after 3 trials to ∼15% after 12 trials - as compared to the initial *TrainingSet-1*.

Figure 9 shows the mean error rate for all participants after the robot was trained using *TrainingSet-1* and *TrainingSet-1* + *TrainingSet-2*, and evaluated using *EvaluationSet-2*. By using error perception feedback from the user to identify errors made by the robot and assigning the next action with highest probability score, it can be noticed that error rate decreased - from an initial ∼35% after 3 trials to ∼15% after 12 trials - as compared to the initial *TrainingSet-1*. These results suggest that in order to decrease error rate and increase learning performance it is very convenient to take into consideration user's error perception feedback.

VII. CONCLUSIONS

In this paper we presented a BMI-based framework that incorporates learning capabilities and error-perception feedback. We described in detail a general system architecture that can be adapted to multiple applications. We showed that our framework can be used by a human to teach a robot new skills by tele-operating the robot using a EEG-headset. Moreover, the framework also allows the user to evaluate robot's performance by identifying error related negativity while the person observes robot actions. Experimental results demonstrate that by using human error perception, it is possible to identify robot's mistakes and use this information to improve robot's learning performance.

REFERENCES

[1] J.R. Wolpaw. "Brain-computer interface technology: a review of the first international meeting", IEEE Trans. Rehabilitation Engineering, vol. 8, no. 2, pp. 164-173, 2000.
[2] C.Penaloza, Y. Mae, K. Ohara, and T. Arai: "BMI-based Learning System for Appliance Control Automation", IEEE International Conference on Robotics and Automation (ICRA 2013), Karlsruhe, Germany. May 6 - 10, 2013.
[3] I. Iturrate, J. Antelis, and J. Minguez. 2009. Synchronous EEG brain-actuated wheelchair with automated navigation. In Proceedings of the 2009 IEEE international conference on Robotics and Automation (ICRA'09). IEEE Press, Piscataway, NJ, USA, 2530-2537.
[4] Frizera-Neto A, Celeste WC, Martins VR, Bastos-Filho TF, Sarcinelli-Filho M: Human-Machine Interface Based on Electro-Biological Signals for Mobile Vehicles. Proceedings of the International Symposium on Industrial Electronics (ISIE 2006), Montreal, Canada 2006:2954-2959.
[5] Matthew Bryan, Griffin Nicoll, Vibinash Thomas, Mike Chung, Joshua R. Smith, and Rajesh P.N. Rao, Automatic Extraction of Command Hierarchies for Adaptive Brain-Robot Interfacing. ICRA, Jul 2012
[6] J. Park, K.-E. Kim, and S. Jo, "A POMDP approach to P300-based brain computer interfaces", In Proc Int Conf Intelligent User Interfaces, Hong Kong, China, Feb 2010.
[7] G. Pfurtscheller and C. Neuper, "Motor imagery activates primary sensorimotor area in humans," Neuroscience letters, vol. 239, pp. 65-68, 1997.
[8] G. Pfurtscheller, C. Brunner, A. Schlogl, and F. H. Lopes de Silva, "Mu rhythm (de) synchronization and EEG single-trial classification of different motor imagery tasks," Neuroimage, vol. 31, pp. 153-159, 2006.
[9] Wijesuriya, N.; Tran, Y.; Thuraisingham, R.A.; Nguyen, H.T.; Craig, A.; , "Effects of mental fatigue on 8-13Hz brain activity in people with spinal cord injury," Engineering in Medicine and Biology Society, 2008. IEEE EMBS 2008, vol., no., pp.5716-5719, 20-25 Aug. 2008
[10] Lopes, A.C.; Nunes, U.; Vaz, L.; , "Assisted navigation based on shared-control, using discrete and sparse human-machine interfaces," Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE , vol., no., pp.471-474, Aug. 31 2010-Sept. 4 2010
[11] M. Chung, W. Cheung, R. Scherer, R. Rao: A Hierarchical Architecture for Adaptive Brain-Computer Interfacing. IJCAI 2011: 1647-1652
[12] M. Bryan, J. Green, M. Chung, R. Scherer, L. Chang, J. R. Smith, R. Rao, "An Adaptive Brain-Computer Interface for Humanoid Robot Control," 11th IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids 2011), Bled, Slovenia, October 26-28, 2011
[13] Fabien Lotte, Aurelien van Langhenhove, Fabrice Lamarche, Thomas Ernest, Yann Renard, Bruno Arnaldi, and Anatole Lecuyer, Exploring Large Virtual Environments by Thoughts Using a Brain-Computer Interface Based on Motor Imagery and High-Level Commands,Presence: Teleoperators and Virtual Environments 2010 19:1, 54-70
[14] H.T. van Schie, R.B. Mars, M. G. H. Coles, H. Bekkering, "Modulation of activity in medial frontal and motor cortices during error observation," Nat. Neuroscie., vol. 7, pp. 549-554, 2004.
[15] P.W. Ferrez and J.R. Millan. Error-related EEG potentials generated during simulated Brain-Computer Interaction, IEEE transactions on Biomedical Engineering, vol. 55, no. 3, pp.923-929, 2008.
[16] P.W. Ferrez and J.R. Millan. Error-related EEG potentials in brain-computer interfaces. Towards Brain-Computer Interfacing, MIT Press, Cambridge,Massachusetts, 2007.
[17] I. Iturrate, L. Montesano, and J. Minguez. "Robot reinforcement learning using EEG-based reward signals" IEEE International Conference on Robotics and Automation (ICRA 2010), Anchorage, Alaska, May 3 - 8, 2010.
[18] Chavarriaga, Ricardo; Millan, Jose del R., "Learning from EEG Error-related Potentials in Noninvasive Brain-Computer Interfaces," IEEE Trans on Neural Systems and Rehabilitation Engineering (ISSN: 1534-4320), vol. 18, num. 4, p. 381-388, 2010.
[19] C.Penaloza, Y. Mae, K. Ohara, and T. Arai: "Software Interface for Controlling Diverse Robotic Platforms using BMI", IEEE/SICE International Symposium on System Integration. Fukuoka, Japan. December 16-18, 2012.
[20] NAO For Research, Aldebaran Robotics. http://www.aldebaran-robotics.com/
[21] A. Ferreira, R.L. Silva, W.C. Celeste, T.F. Bastos Filho and M. Sarcinelli Filho, Human-machine interface based on muscular and brain signals applied to a robotic wheelchair,J. Phys.: Conf. Ser. 2007 90 012094
[22] EmotivSystems. Emotiv - brain computer interface technology. http://emotiv.com.
[23] Coles., M.K.S.a.M.G.H. Performance Monitoring in a Confusing World: Error-Related Brain Activity, Judgments of Response Accuracy, and types of Errors. Proc. of JEPHPP.26.1: (2000), 141-151.
[24] Iturrate I, Montesano L, Minguez J. Single trial recognition of error-related potentials during observation of robot operation. Proc IEEE Eng Med Biol Soc. 2010:4181-4
[25] Brian Potetz (2011), Estimating the Bayes Point Using Linear Knapsack Problems, Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp.257–264
[26] Minka, T. (2001) "Expectation Propagation for Approximate Bayesian Inference". Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, August 2-5, 2001. pp 362-369