IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 18, 2019, accepted July 30, 2019, date of publication August 5, 2019, date of current version August 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933268

Design and Optimization of a BCI-Driven Telepresence Robot Through Programming by Demonstration

BERDAKH ABIBULLAEV[1], (Member, IEEE), AMIN ZOLLANVARI[2], (Senior Member, IEEE), BATYRKHAN SADUANOV[1], AND TOHID ALIZADEH[1], (Member, IEEE)

[1]Department of Robotics and Mechatronics, Nazarbayev University, Nur-Sultan 010000, Kazakhstan
[2]Department of Electrical and Computer Engineering, Nazarbayev University, Nur-Sultan 010000, Kazakhstan

Corresponding author: Berdakh Abibullaev (berdakh.abibullaev@nu.edu.kz)

This work was in part supported by the Nazarbayev University Faculty Development Competitive Research under Grant SOE2018008.

ABSTRACT Improving the life quality of people with severe motor paralysis has a significant impact on restoring their functional independence to perform activities of daily living (ADL). Telepresence is a subfield of the robotic-assisted route, where human plays the role of an operator, sending high-level instructions to an assistive robot while receiving sensory feedback. However, for severely motor-impaired people, conventional interaction modalities may not be suitable due to their complete paralysis. Thus, designing alternative ways of interaction such as Brain-Computer Interfaces (BCI) is essential for a telepresence capability. We propose a novel framework that integrates a BCI system and a humanoid robot to develop a brain-controlled telepresence system with multimodal control features. In particular, the low-level control is executed by Programming by Demonstration (PbD) models, and the higher-level cognitive commands are produced by a BCI system to perform vital ADLs. The presented system is based on real-time decoding of attention-modulated neural responses elicited in the brain electroencephalographic signals and generating multiple control commands. As a result, the system allows a user to interact with a humanoid robot while receiving auditory and visual feedback from the robot's sensors. We validated our system across ten subjects in a realistic scenario. The experimental results show the feasibility of the approach in the design of a telepresence robot with high BCI decoding performances.

INDEX TERMS Brain-Computer interfaces, telepresence, programming by demonstration, EEG, event-related potentials, humanoid robots.

I. INTRODUCTION

The last decade has seen remarkable progress in Brain-Computer Interface (BCI) research for decoding electroencephalogram (EEG) signals that are generated in the brain to command external devices [1]–[5]. In particular, experimental demonstrations proved the potential of BCI technology in scenarios such as controlling a computer cursor [6], wheelchair [7], [8], robotic manipulators [9], [10] and/or humanoid control [11], [12].

BCI technology holds promise to improve the quality of life of people with restricted mobility due to paralysis. Numerous studies have demonstrated successful outcomes in controlling prosthetic limbs and manipulators by paralyzed people using BCI approaches [9], [13], [14]. Although in some situations using robotic manipulators with BCIs could be more convenient, the humanoids are logically more suitable for telepresence, since humans are psychologically more inclined to interact with them [15].

Several BCI methods have been used in human telepresence studies [16]–[20]. These studies have shown that the choice of an appropriate paradigm is an essential step in designing the telepresence system. Sensorimotor Rhythms (SMR) based BCIs are one of the best candidate paradigms [12], [21]–[23] because these types of BCI do not require an external visual or auditory stimulation to exert control signals, and therefore are considered to be more convenient for the user. However, such systems are usually limited to few task choices, require an exhaustive concentration of users, and have extended training time, which, altogether,

The associate editor coordinating the review of this manuscript and approving it for publication was Kezhi Li.

render their usage troublesome for disabled people. Furthermore, external stimuli driven BCI systems are regularly used in telepresence studies. For instance, Steady-State Visual Evoked Potential (SSVEP) BCIs could lead to a high information transfer rate (ITR) and a remarkable performance with little or no training session [24]–[27]. However, these advantages come at a price as the effective frequency range restricts the number of choices in SSVEP-based interfaces. Moreover, and more seriously, the intensive eye strains caused by stimuli flickering in SSVEP BCIs lead to eye/vision problems in patients.

In this study, we design a BCI paradigm based on event-related potentials (ERP) measured by Electroencephalography (EEG) recordings in healthy subjects to control an external humanoid robot for telepresence. ERPs are attention modulated brain responses to external target visual/auditory or haptic stimuli events and have different spatiotemporal resolutions and features compared to background events in EEGs. The difference lies in the P300 component, which is a positive wave observed when a user attends to a rare stimulus with a positive peak occurring 300-400 ms after the stimulus event onset [28]–[30]. One of the central tasks in the present study is to decode different ERP activity patterns produced by a user on a single trial basis and translate them into appropriate commands for a telepresence robot.

Human action performance can be divided into two major parts including decision and implementation [31]. A decision is merely a conscious high-level choice of possible actions, but the implementation of a task is a matter of unconsciousness. For instance, a human has a set of skills that are learned previously and may be applied in appropriate contexts. Depending on the situation, relevant skill and action are chosen to gain advantage from the current situation. Implementation of action is a complicated process that requires various control, trajectory generation, timing, and other systems to be involved. Although ITR of BCI systems does not allow such full supervision of the robot, low-level control may be redundant in such a pipeline, since even human beings are not decoding body motions consciously at mechanical and dynamical levels [32]. Programming by demonstration is a convenient way of teaching robots intuitively by imitation or enforced demonstrations. The trainer needs to know neither programming nor robotics, and the teaching process is similar to instructing a child.

In this work, we propose a robust BCI–to–telepresence system to interact with a humanoid robot that is expected to improve the social aspects of people with severe paralysis. Such health innovations, in turn, will help augment mobility and interaction capability for people in need.

II. MATERIALS AND METHODS

A. STRUCTURE OF THE PROPOSED BCI - TELEPRESENCE SYSTEM

The system architecture for the closed-loop real-time BCI system integrated with the physical telepresence robot is shown in Fig.1. Our system design is based on the client-server model. The fundamental parts of the BCI to telepresence system consists of data acquisition, signal processing, exoskeleton control, stimulus presentation, and data storage clients that interact with each other through a Buffer Server. All these clients are controlled by the "Experiment Control" client that contains the main graphical user interface (GUI) and is responsible for scheduling processing and determining all the sequences of operations during an experimental session. In particular, we implemented an event-driven programming paradigm where events generated by each client determines the flow of the experimental paradigm and inter-process communication. Each client in Fig.1 has an event trigger and event listener functions. For instance, a mouse click on GUI in BLOCK I to start EEG data acquisition sends an event to the Buffer Server, and BLOCK II's event listener catches the event to launch the EEG data acquisition. Then, the continuous data is sent to Buffer Server for temporary storage (up to 1 minute, 50 events) and subsequently putting the data into permanent storage. An example scenario for a BCI based robot control has the following sequence of operations: 1) start the buffer server and initiate all event listeners in each client in BLOCK I; 2) start acquiring data from BLOCK II; 3) launch the stimulus presentation software in BLOCK III; 4) start BCI control of the robot via BLOCK IV, and BLOCK V, where signal processing block processes the EEG data and send the results to BLOCK V.

B. PARTICIPANTS

Ten healthy adults (age range 22 to 35 years) with no history of neurological, physical, or psychiatric illness participated in this study. All the participants were naive, BCI users who had not participated in any related experiments before. Informed consent was received from all study participants. The Nazarbayev University Institutional Research Ethics Committee has approved the study.

C. ELECTROENCEPHALOGRAPHY

Scalp EEG was recorded using a 16-channel, active Ag/AgCl electrodes (g.USBamp, g.LADYbird, Guger Technologies OG, Schiedlberg, Austria) with a sampling frequency of 256 Hz. The EEG electrodes were positioned according to the International 10-20 system. The right earlobe of participants was used for a ground electrode, whereas the FCz location was used for a reference electrode (see Fig.2).

D. CALIBRATION SESSION

We implemented the Farwell & Donchin style speller [29] using a 4×4 grid of alphanumeric characters, presented through an LCD monitor as shown in Fig.3. Participants were seated in a comfortable chair facing the LCD monitor with a distance of about 60 cm in between. Each participant performed a single session during which their EEG signals were recorded. Every time the target symbol appears either in a flashed row or column, participants were instructed to attend mentally to the shown stimulus and count silently.

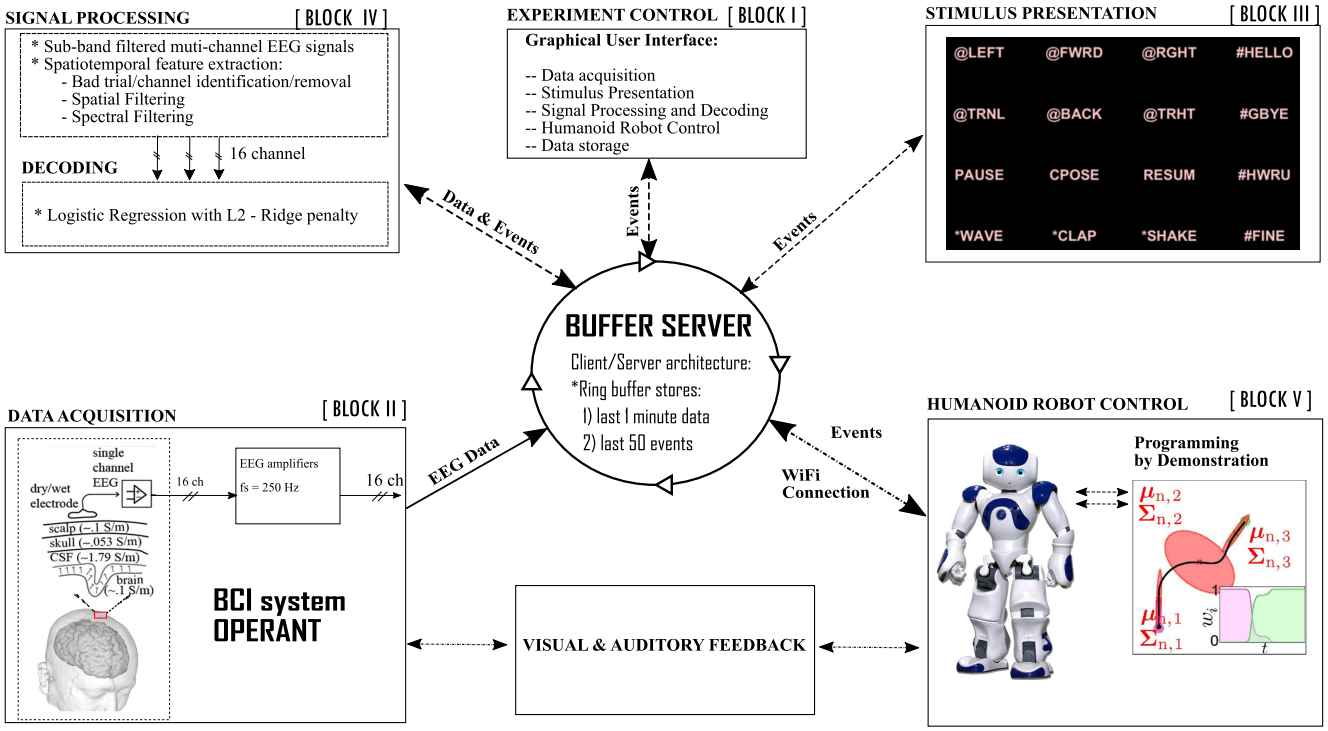

FIGURE 1. An overview of the system architecture developed for the closed-loop real-time BCI system integrated with the telepresence robot. The fundamental parts of the BCI to humanoid robot consists of data acquisition, signal processing, robot control, stimulus presentation, and data storage clients that interact with each other through a Buffer Server, which is based on a client-server model.

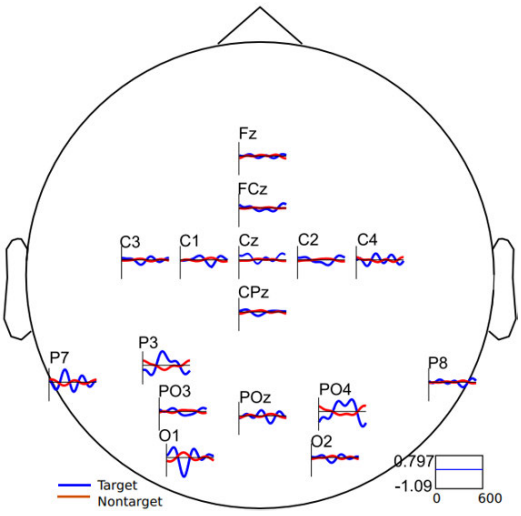

FIGURE 2. The electrode montage used in the current study. All channels are plotted with prototypical ERP waveforms.

FIGURE 3. A visual cue used to evoke P300 and provide direct control commands to the NAO robot.

The total number of sequences that a participant had to attend was equal to five with an inter-sequence duration of two seconds. Each sequence consisted of three complete row and column stimulus repetitions. In total, five repetitions of each target character were made where one repetition (or trial) consisted of a complete set of 12 random flashes of every row and column (six rows and six columns). The inter-stimulus interval (ISI) was set to 150 ms; likewise, the stimulus duration was 100 ms (i.e., the time length a row/col is highlighted). The minimum time between the same target letter highlights, which is generally referred to as target-to-target interval (TTI), was set to 600 ms. The stimulus onset asynchrony (SOA) was set to 250 ms. The SOA is the time period between the start of one stimulus event and the start of the next event (See Fig. 4).

E. FEEDBACK SESSION

Following the calibration task, continuous EEG data were segmented and processed using the steps described in Section II-F and subsequently, a classifier model was optimized on the acquired training data without taking off the EEG cap. This step was fast, which took 70 seconds on average for all participants. Once the BCI classifier model was obtained, subjects were instructed to control the humanoid robot on their preferences. The subjects could use one out of sixteen commands displayed on the stimulus grid in any order to achieve a certain task in sequence. For instance, to greet a

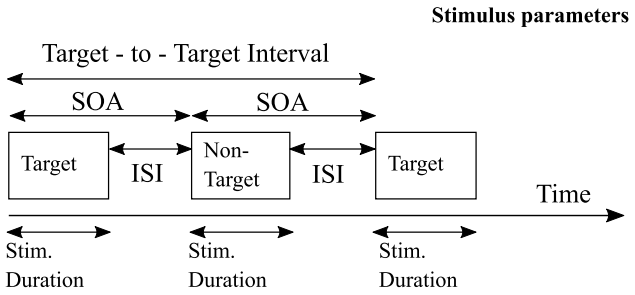

FIGURE 4. The parameters of the visual stimulus presentations. ISD: Inter-sequence duration, ISI: Inter-stimulus interval, TTI: target-to-target interval, SOA: Stimulus onset asynchrony.

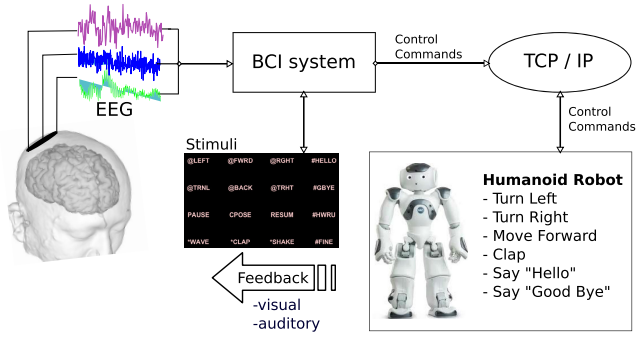

FIGURE 5. A simplified view of the BCI-to-telepresence system structure.

person in a remote location, a user could exert of the following commands

- **locomotion** i.e., move forward, turn right or left
- **interaction** i.e., say "Hello," "How are you," and "Shake hands," and say "Good-Bye."

A simplified version of the system architecture is shown in Fig.5 that depicts the user interaction with the telepresence robot via BCI. The users of the BCI system could also observe the environment in real-time using the sensors of the humanoid robot and interact whenever any cue obtained from other people in the interaction environment. Besides, the participants could stop the session at any time either by a mental selection of the "Pause" command or by informing the experimenter. The entire pipeline of the developed BCI-based telepresence robot is depicted in Fig.1. The humanoid robot was able to be trained for new tasks at any stage of the session since robot control was an independent buffer client and was not synchronized with the BCI pipeline. There are three slots for PbD training tasks, which can be updated at any time independently of the session stage. NAO landmarks with naoqi built-in functions to detect, recognize, and localize objects as task parameters (TPs) to pass as the arguments to the Task-parameterized Gaussian mixture model (TP-GMM) algorithm [33] were used. Reproduction trajectory is a seven-dimensional vector having time point, three-position, and three orientation values for the end-effector.

F. EEG SIGNAL PROCESSING

The following steps summarize the signal pre-processing methods applied for a single trial detection ERPs.

- *Segmentation*: Continuous EEG data were segmented into a target and non-target trials with 600 ms duration after the stimulus event onset markers.
- *Detrending*: Arbitrary offsets in data were removed by subtracting the total mean from each channel. This step ensures removal of noise in the form of slow drifts which might occur due to sweating and a poor sensor-to-head contact [34].
- *Bad trial removal*: EEG data trials were artifact edited using a statistical thresholding procedure to eliminate bad trials associated with egregious movement artifacts. The criteria to detect bad trials was based on calculating the mean absolute value per trial and excluding trials with values higher than 3-standard deviations from the median trial.
- *Bad channel removal:* All channels were analyzed across trials to identify electrodes that are corrupted with excessive noise arising from improper connection to the scalp of a participant. Channels with excessively high power were determined by computing the total power for each channel overall epochs as well as the mean and the variance in channel power. Any channels with power more than three standard deviations were removed and replaced with common averaged reference channel.
- *Spatial Filtering*: A spatial whitening filter was applied to minimize the noise due to source mixing and volume conduction. The whitening filter uses linear re-weighting of the electrodes and maps raw electrode readings to a new space where the sensors are uncorrelated and have unit power.
- *Spectral Filtering*: EEG data has been band-pass filtered between 0.5-12 Hz range using a Fourier filter. First, the signal was Fourier transformed, and then a weighting is applied to suppress and remove unwanted frequencies outside the frequency of interest range [35]. The weighted signal was inverse Fourier transformed to obtain the filtered signal.

While an abundant number of pre-processing methods exist in the literature for ERP detection, we note that the aforementioned steps were adopted based on the study in [36], [37] which empirically verifies a near-optimal approach. All the methods have been implemented in the MNE toolkit [38].

1) CLASSIFICATION

In [35], [39], we showed that Logistic Regression with L_2-Ridge (LRR) penalty [40] can generally outperform a number of other classifiers including naive Bayes [41, p.99], tree-augmented naive Bayes [42]), support vector machines with radial basis function [43, p. 190-191], and logistic regression with L_1-least absolute shrinkage and selection operator penalty [44]. We observed this relative superiority of LRR on both extracted spatiospectral features and temporal features of ERP waveforms [35], [39]. In what follows, we describe the application of LRR on temporal features, which, as observed in [39], generally lead to a higher

TABLE 1. Number of training trials for each participants.

Subjects	1	2	3	4	5	6	7	8	9	10
Target	296	292	277	287	286	277	283	295	299	571
Non-Target	880	871	840	859	863	844	851	883	891	3404

classification accuracy than spatiospectral features in decoding ERP waveforms.

Let the set of pairs $S_{\text{train}} = \{(\mathbf{z}_1, y_1), \ldots, (\mathbf{z}_n, y_n)\}$ denote n trials of EEG recordings for each subject where $y_i \in \{0, 1\}$ is the realization of class variable Y representing target or non-target, and $\mathbf{z}_i \in \mathbb{R}^{(pc)\times 1}$ is an instance of EEG observation over c channels and p time points defined as

$$\mathbf{z}_i = [\mathbf{x}_{i1}^T, \mathbf{x}_{i2}^T, \ldots, \mathbf{x}_{ic}^T]^T, \quad i = 1, \ldots, n, \tag{1}$$

with $\mathbf{x}_{ij} \in \mathbb{R}^{p\times 1}, j = 1, \ldots, c$, and T denoting the transpose operator. The index The index i in z_i, $\mathbf{x}_{ij}$, and y_i are used to represent the i^{th} trial used to train the classifier. Once the classifier is trained, it is applied to classify an observation $\mathbf{z}$ = an observation $z = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_c^T]^T$, which is collected similar to $\mathbf{z}_i$ except that it is collected on one trial (therefore, no index i). The number of training trials training trials for each participant is shown in Table 1.

In logistic regression, it is assumed that for an observed $\mathbf{z}$ with an unknown class variable Y, the class posteriors $P(Y|\mathbf{z})$ is represented as the logistic function; that is to say,

$$p(\mathbf{z}; \beta_0, \boldsymbol{\beta}) \triangleq P(Y = 1|\mathbf{z}) = \frac{e^{(\beta_0 + \mathbf{z}^T\boldsymbol{\beta})}}{1 + e^{(\beta_0 + \mathbf{z}^T\boldsymbol{\beta})}}, \tag{2}$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{pc}]^T \in \mathbb{R}^{(pc)\times 1}$ is the vector of the regression coefficients, β_0 is an intercept, and $P(Y = 0|z) = 1 - P(Y = 1|z)$. We full length of waveforms in each trial, which contains $p = \lfloor 256\,\text{Hz} \times 600\,\text{ms} \rfloor = 153$ time points. Therefore, each observation z in the temporal analysis contains $p \times c = 153 \times 16\,(\text{channels}) = 2448$ features. The regression coefficients in logistic estimated by maximizing the log-likelihood function; in other words,

$$\{\hat{\beta}_0, \hat{\boldsymbol{\beta}}\} = \underset{\{\beta_0, \boldsymbol{\beta}\}}{\arg\max}\ l(\beta_0, \boldsymbol{\beta}), \tag{3}$$

where

$$l(\beta_0, \boldsymbol{\beta}) = \sum_{i=1}^{n} \big[y_i \log\big(p(\mathbf{z}_i; \beta_0, \boldsymbol{\beta})\big) + (1 - y_i)\log\big(1 - p(\mathbf{z}_i; \beta_0, \boldsymbol{\beta})\big)\big]. \tag{4}$$

The main difference between LRR and classical logistic regression that makes LRR an attractive choice for high-dimensional setting such as in BCI applications (e.g., in the present study pc is comparable to n), is the use of the penalized log-likelihood function to estimate regression coefficients as

$$\{\hat{\beta}_0, \hat{\boldsymbol{\beta}}\} = \underset{\{\beta_0, \boldsymbol{\beta}\}}{\arg\max}\ l^{\lambda}(\beta_0, \boldsymbol{\beta}) \triangleq l(\beta_0, \boldsymbol{\beta}) - \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2, \tag{5}$$

where $\|\boldsymbol{\beta}\|_2 = \left(\sum \beta_j^2\right)^{1/2}$ is the l_2 norm of $\boldsymbol{\beta}$. This maximization is generally solved using stochastic gradient descent (SGD) algorithm [45], [46]. Moreover, we estimate the parameter λ, which regulates the amount of shrinkage [40], using 5-fold cross-validation in a predetermined range of λ. Once LRR is trained, it is used in real-time sessions to infer user's mental intent and to exert control commands to NAO robot.

G. PROGRAMMING BY DEMONSTRATION

1) OBJECT LOCALIZATION

NAO Landmarks were used to determine the TPs by placing them in different locations on the worktop. They were detected using naoqi package built-in function, which returns various information about the landmark found. The required output parameters for geometric localization of landmarks in 3D robot body space include the position of the center, angular and theoretical size of the landmark on the captured image and head orientation (yaw, pitch roll) values. Distance to the landmark and landmark position with respect to the body of the humanoid is calculated as:

$$d = \frac{R_t}{\tan(R_a)}, \tag{6}$$

$${}^bP = {}^b_cT\,{}^c_lT\,{}^lP, \tag{7}$$

where d is the distance between landmark and camera, R_t and R_a are theoretical and angular radii, respectively, ${}^bP$ and ${}^lP$ are positions of a landmark with respect to the robot body and 2D camera space, respectively, and ${}^b_cT$ and ${}^c_lT$ are camera-to-body and landmark-to-camera transformation matrices, which were determined using built-in naoqi functions.

2) DATA ACQUISITION

Humanoid arms are set up to zero stiffness mode to allow enforced guidance. The robot is placed in front of its workspace where all landmarks are localized before the demonstration phase. At the demonstration step, robot hands are guided by the task demonstrator. With maximum possible frequency (about 10 data points per second) end-effector 6D position and orientation vectors are calculated by forward kinematics routine and stored in the demonstration matrix. For each task, several demonstrations (from 3 to 5) were performed while task parameters were varying.

3) SIGNAL PROCESSING

All demonstrations have to be of the same length in terms of data points for consistent matrix operations

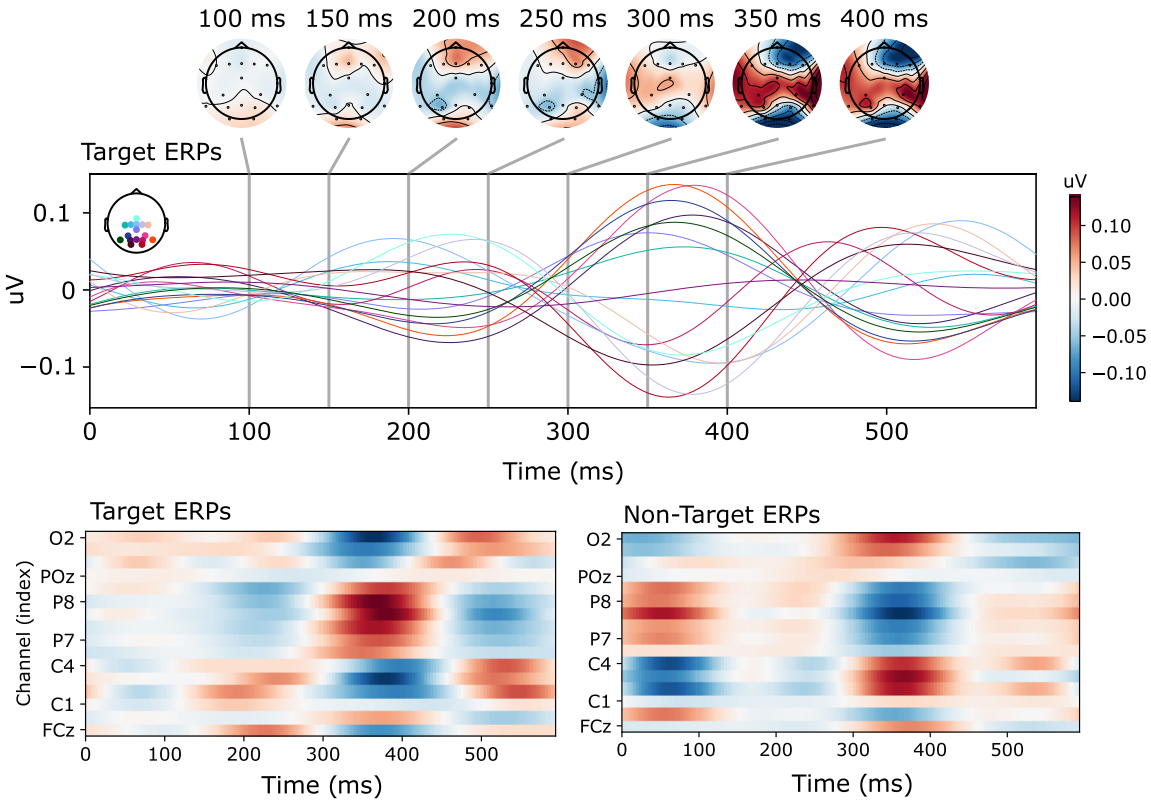

FIGURE 6. A time course evolution of grand averaged sixteen channel ERPs from Subject #10. Upper panel shows target ERPs along with its topographic distributions across different time points. The lower left panel shows color-coded representation target ERPs amplitude values. The lower right panel shows the colored representations of the non-target ERPs amplitude values. Event-related brain responses patterns is pronounced clearly between 300 - 400 ms time interval.

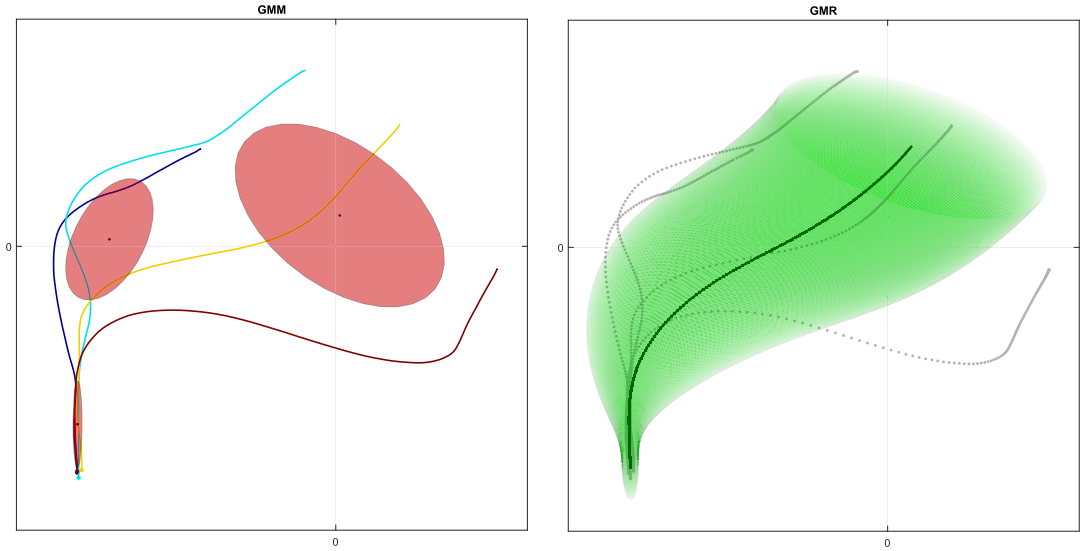

FIGURE 7. (a) The Gaussian mixture model with three components encoding the four trajectory samples, and (b) reproduction of the task (shown in a darker color), without considering the task parameters.

and manipulations. Furthermore, demonstration noises were eliminated to reproduce smoother trajectories.

- *Interpolation:* A cubic Hermite spline algorithm [47] is used in order to fit all demonstrations to a specific size. The raw data is sliced where each piece of the data is considered as a third-degree polynomial specified in Hermit form. It predicts new data points to decrease or increase the number of acquired samples and makes data points equally spaced [47].
- *Smoothing filter:* To increase the signal-to-noise ratio Savitzky—Golay filter was applied to each position and orientation axis. Since data points are equally spaced,

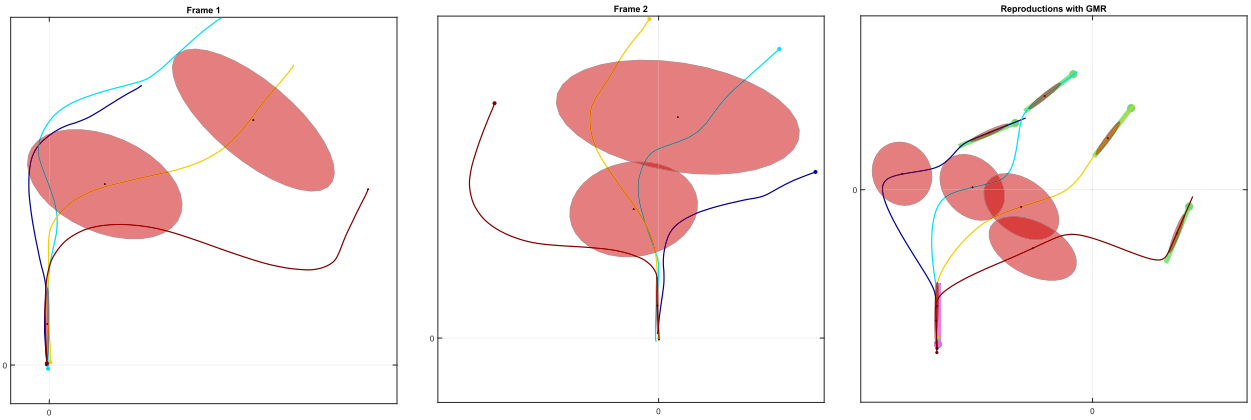

FIGURE 8. Assuming two task parameters or frame of references for the same data. Encoded GMM and observation of the data from the (a) first, (b) second frame of reference, and (c) resulting GMMs and reproductions of the task.

least-squares convolution coefficients were found and applied to demonstration pieces to evaluate new smooth data trajectories [48].

4) TP-GMM

Task-parameterized Gaussian Mixture Model (TP-GMM) is an extended version of the classic Gaussian mixture model algorithm, which has been widely used to encode robot motions in recent years [49], [50]. TP-GMM considers environmental objects' positions and orientations as task parameters (TP) [33], [51]. TP-GMM has better performance than GMM for encoding and reproducing task modulated trajectories, as illustrated in Figs. 7 and 8, where there are four demonstration samples shown in different colors. In Fig. 7-(a), TPs are not taken into account and the classic GMM is used to encode the whole trajectories, and then in Fig. 7-(b), Gaussian mixture regression (GMR) is used to generate a new trajectory, which is basically taking the average of all the trajectories. On the other hand, in Fig. 8-(a) and (b), the trajectories are observed from two different frames of references and a GMM model is fit to each of them. In Fig. 8-(c), the two GMM models are combined and GMR is used to generate a new trajectory. For the reproductions shown in Fig. 8-(c), the TPs are the same as the ones in the demonstration, however, in general, they could be different. TPs can be expressed as coordinate systems by $\{\mathbb{TP}_j : \boldsymbol{b}_j, \boldsymbol{A}_j\}_{j=1}^{P}$, where position vector $\boldsymbol{b}_j$ is the origin of the observer, $\boldsymbol{A}_j$ denotes the transformation matrix, P is the maximum number of TPs, and j is the index of the TP. In our case, TPs are considered as NAO round landmarks, having identity matrix as for any $\boldsymbol{A}_j$. All possible task parameters are assumed to be known beforehand and provided by the demonstrator. After TP-GMM model is fitted to an arbitrary task, unnecessary TPs will be automatically eliminated by the system. A set of M demonstrations, each with T_m datapoints and D dimensions are coupled to construct the trajectory $\{\boldsymbol{\xi}\} \in \mathbb{R}^{D\times N}$ with $N = \sum_{m=1}^{M} T_m$. These demonstrations are then observed from (or projected into)

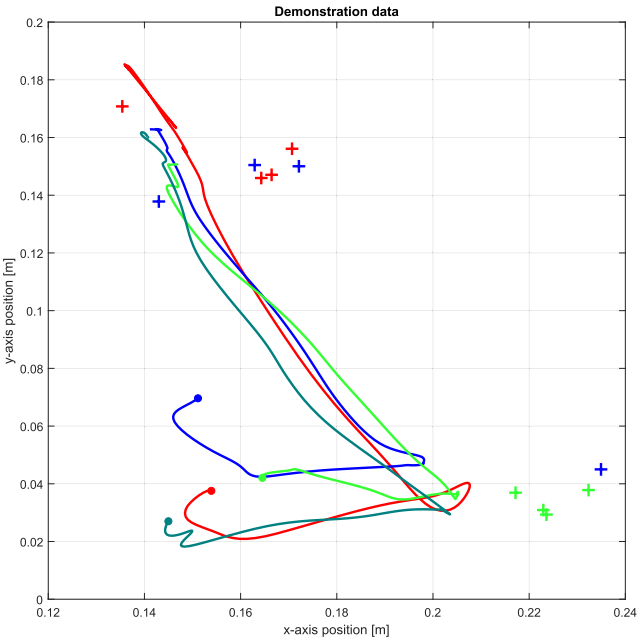

FIGURE 9. Demonstration samples for one of the tasks, in the form of trajectories of the left-hand end-effector, projected onto a 2D vertical plane. The color-coded + signs (green, red, and blue) represent the TPs.

different frames of references to form P trajectory samples $X^{(j)} \in \mathbb{R}^{D\times N}$. The TP-GMM model parameters include $\bigcup_{i=1}^{K}\left\{\pi_i, \bigcup_{j=1}^{P}\{\boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\}\right\}$, where π_i denotes the mixing coefficients, K denotes the number of Gaussian components, and $\boldsymbol{\mu}_i^{(j)}$ and $\boldsymbol{\Sigma}_i^{(j)}$ are the center and covariance matrices of the i^{th} Gaussian state in frame j, respectively. To determine π_i, $\boldsymbol{\mu}_i^{(j)}$ and $\boldsymbol{\Sigma}_i^{(j)}$, Expectation-Maximization (EM) is used with K-means clustering initialization and a stopping criterion based on the maximization of log-likelihood.

The TP-GMM model is used for the reproduction of new trajectories based on new task parameters (i.e., different position and orientation of objects). A GMM model is retrieved from the multiplication of all linearly transformed GMMs

corresponding to different TPs:

$$\mathcal{N}(\boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{t,i}) \propto \prod_{j=1}^{P} \mathcal{N}(\boldsymbol{A}_{t,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{t,j}, \boldsymbol{A}_{t,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{t,j}^{\top}). \quad (8)$$

GMR uses GMM model in order to generate a new trajectory for new TPs. For each time step t, the data and GMM parameters are partitioned as:

$$\boldsymbol{\xi}_t = \begin{bmatrix} \boldsymbol{\xi}_t^{\mathbf{I}} \\ \boldsymbol{\xi}_t^{\mathbf{O}} \end{bmatrix}, \quad \boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^{\mathbf{I}} \\ \boldsymbol{\mu}_i^{\mathbf{O}} \end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^{\mathbf{I}} & \boldsymbol{\Sigma}_i^{\mathbf{IO}} \\ \boldsymbol{\Sigma}_i^{\mathbf{OI}} & \boldsymbol{\Sigma}_i^{\mathbf{O}} \end{bmatrix}, \quad (9)$$

where the superscripts **I** and **O** represent input (e.g. time values) and output (e.g. position in 3D) dimensions, respectively, and

$$\boldsymbol{\mu}_i^{\mathbf{I}} = \mathrm{E}[\boldsymbol{\xi}_t^{\mathbf{I}}], \quad (10)$$

$$\boldsymbol{\Sigma}_i^{\mathbf{I}} = \mathrm{E}\Big[(\boldsymbol{\xi}_t^{\mathbf{I}} - \boldsymbol{\mu}_i^{\mathbf{I}})(\boldsymbol{\xi}_t^{\mathbf{I}} - \boldsymbol{\mu}_i^{\mathbf{I}})^T\Big], \quad (11)$$

$$\boldsymbol{\Sigma}_i^{\mathbf{IO}} = \mathrm{E}\Big[(\boldsymbol{\xi}_t^{\mathbf{I}} - \boldsymbol{\mu}_i^{\mathbf{I}})(\boldsymbol{\xi}_t^{\mathbf{O}} - \boldsymbol{\mu}_i^{\mathbf{O}})^T\Big], \quad (12)$$

$$\boldsymbol{\Sigma}_i^{\mathbf{OI}} = \mathrm{E}\Big[(\boldsymbol{\xi}_t^{\mathbf{O}} - \boldsymbol{\mu}_i^{\mathbf{O}})(\boldsymbol{\xi}_t^{\mathbf{I}} - \boldsymbol{\mu}_i^{\mathbf{I}})^T\Big], \quad (13)$$

where $\boldsymbol{\mu}_i^{\mathbf{O}}$ and $\boldsymbol{\Sigma}_i^{\mathbf{O}}$ are defined by replacing $\boldsymbol{\xi}_t^{\mathbf{I}}$ with $\boldsymbol{\xi}_t^{\mathbf{O}}$ in definition of $\boldsymbol{\mu}_i^{\mathbf{I}}$ and $\boldsymbol{\Sigma}_i^{\mathbf{I}}$, respectively. The joint distribution of $\mathcal{P}(\boldsymbol{\xi}_t^{\mathbf{O}}, \boldsymbol{\xi}_t^{\mathbf{I}}) \sim \sum_{i=1}^{K} \pi_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is obtained from product GMM found in (8), where K was empirically set to 5. For each time step t, the distribution of the conditional random vector $\boldsymbol{\xi}_t^{\mathbf{O}} \mid \boldsymbol{\xi}_t^{\mathbf{I}}$ is:

$$\boldsymbol{\xi}_t^{\mathbf{O}} \mid \boldsymbol{\xi}_t^{\mathbf{I}} \sim \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_t^{\mathbf{I}}) \mathcal{N}(\hat{\boldsymbol{\mu}}_i^{\mathbf{O}}(\boldsymbol{\xi}_t^{\mathbf{I}}), \hat{\boldsymbol{\Sigma}}_i^{\mathbf{O}}), \quad (14)$$

$$\text{with} \quad \hat{\boldsymbol{\mu}}_i^{\mathbf{O}}(\boldsymbol{\xi}_t^{\mathbf{I}}) = \boldsymbol{\mu}_i^{\mathbf{O}} + \boldsymbol{\Sigma}_i^{\mathbf{OI}} \boldsymbol{\Sigma}_i^{\mathbf{I}^{-1}} (\boldsymbol{\xi}_t^{\mathbf{I}} - \boldsymbol{\mu}_i^{\mathbf{I}}), \quad (15)$$

$$\hat{\boldsymbol{\Sigma}}_i^{\mathbf{O}} = \boldsymbol{\mu}_i^{\mathbf{O}} - \boldsymbol{\Sigma}_i^{\mathbf{OI}} \boldsymbol{\Sigma}_i^{\mathbf{I}^{-1}} \boldsymbol{\Sigma}_i^{\mathbf{IO}}, \quad (16)$$

$$h_i(\boldsymbol{\xi}_t^{\mathbf{I}}) = \frac{\pi_i \mathcal{N}(\boldsymbol{\xi}_t^{\mathbf{I}} | \boldsymbol{\mu}_i^{\mathbf{IO}}, \boldsymbol{\Sigma}_i^{\mathbf{IO}})}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\xi}_t^{\mathbf{I}} | \boldsymbol{\mu}_k^{\mathbf{IO}}, \boldsymbol{\Sigma}_k^{\mathbf{IO}})}, \quad (17)$$

where $\hat{\boldsymbol{\mu}}_i^{\mathbf{O}}$, $\hat{\boldsymbol{\Sigma}}_i^{\mathbf{O}}$, and h_i are the estimated mean vector, the covariance matrix, and the weight for the contribution of the i^{th} component of the resulting GMM, respectively. Finally, regression trajectory $\hat{\boldsymbol{\xi}}_t^{\mathbf{O}}$, which is used as position-orientation trajectory for the end-effector is estimated using $\hat{\boldsymbol{\xi}}_t^{\mathbf{O}} = \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_t^{\mathbf{I}}) \hat{\boldsymbol{\mu}}_i^{\mathbf{O}}$ for a 6D space.

5) IDENTIFYING IRRELEVANT FRAMES OF REFERENCES

TP-GMM training stage initially considers all TPs as relevant frames of reference. Using the first estimated model, the trial trajectory will be generated on the fly, and the trajectory and covariance matrices will be estimated for each frame of reference using GMR. The normalized determinant of the precision (inverse of covariance) matrix determines the frame importance for each time step [52], [53]. Frame with the importance of less than regulated threshold (0.1 in our case) value is eliminated from the task dataset, and new TP-GMM model is trained with updated TPs. In case that all frames are important, the model will remain without updating. The updated TP-GMM model will be used for generating trajectories in new situations.

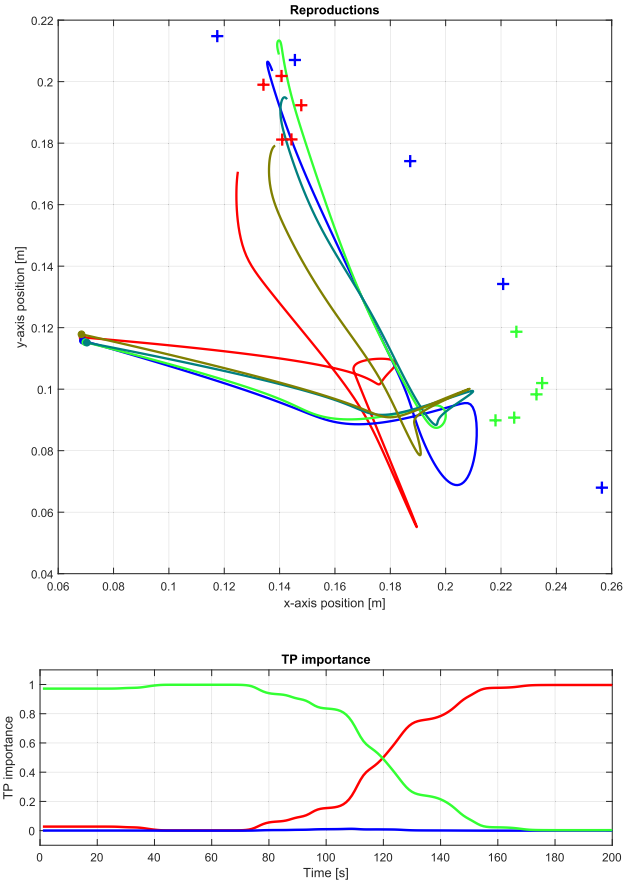

FIGURE 10. The plot on top shows examples of reproduction trajectories of the left hand end-effector projected onto a 2D vertical plane. The plot at the bottom shows the estimated TP importance as a function of time. The green TP is the most important one at the beginning, the red TP becomes important at the end, and the blue TP is not important at all. Blue TP is removed from the model database and ignored during the reproduction stage.

III. RESULTS

The humanoid robot was trained before the calibration stage by the human experimenter. After calibration and BCI training stages, participants were instructed to control the NAO robot voluntarily on a self-paced mode. They preferred to walk around, greet people, and perform trained tasks on a 2D vertical plane. PbD pre-trained tasks were performed smoothly on a 2D vertical plane with landmark objects perceived as task-parameters. The total number of PbD tasks was 3 per experiment, where the humanoid left hand was trained to follow through different landmarks. Each time there was a landmark that was not passed through, it was successfully identified as non-important and was removed from the task database. Demonstration samples for one specific task are illustrated in Fig. 9. Reproductions trials for the same task, as well as frame importance plot, can be seen in Fig. 10.

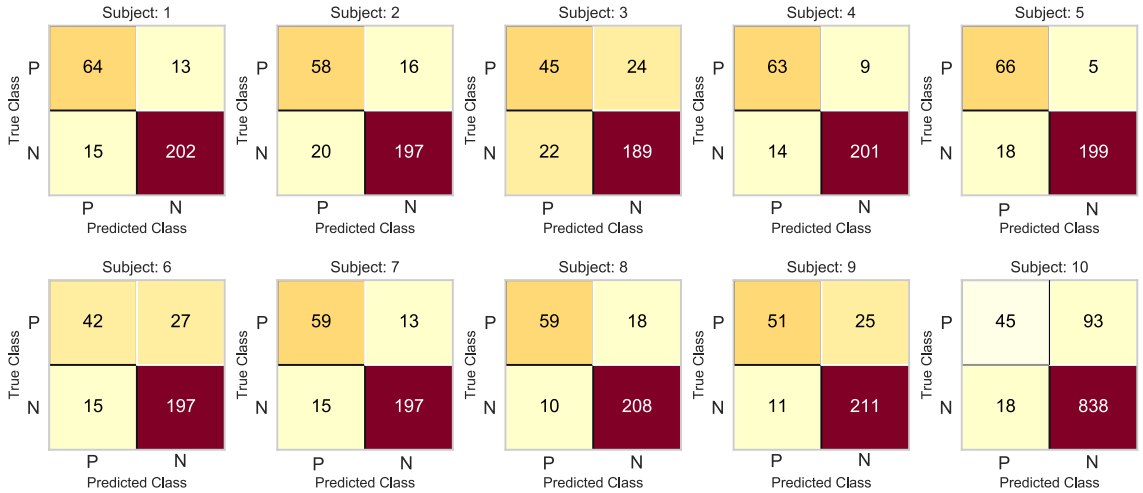

FIGURE 11. Confusion matrix of constructed subject-specific classifier models during ten-fold cross-validation. Here P and N represent the target ERPs and non-target ERP examples, respectively.

The goal was to reach the left-bottom object at the beginning, then finish trajectory at the right-bottom frame. The top landmark was not reached at any training demo, so it was considered as irrelevant TP and ignored by TP-GMM algorithm. From the importance plot at the bottom of Fig. 10, it is observed that the green and the red frames are essential at the beginning and the end of the task, respectively. However, the blue frame importance was always less than 0.1 value.

Prototypical ERPs obtained after signal processing steps described in Section II-F and features used for training the classifiers and for online BCI are shown in Fig.6. The figure shows a time course evolution of grand averaged sixteen channel ERPs from Subject 10. Upper panel shows target ERPs along with its topographic distributions across different time points. The lower left panel shows color-coded representation target ERPs amplitude values. The lower right panel shows the colored representations of the non-target ERPs amplitude values. Event-related brain responses patterns are pronounced clearly between 300 - 400 ms time interval. One can observe the spatial distribution of target and non-target ERPs across time with discriminative patterns, in particular around 300 - 400 ms. Table 1 shows the total number of ERP events acquired for each subject (including both target and non-target) that were used for training subject-specific classifiers used for online BCI. Fig.11 shows the confusion matrix of all constructed classifiers. In the figure, **P** and **N** represent the Target ERPs and Non-Target ERPs, respectively.

To test the significance of the BCI classification score, we applied empirical permutation-based p-value test [54]. To generate the null distribution, we obtained the cross-validated AUCs after randomizing the labels and repeating the procedure many times. Here, the p-value measures the probability that under the null hypothesis, an AUC score would be equal or greater than the AUC observed using the true labels. Therefore, if the p-value is small (we assume, $p < 0.05$), the performance of the constructed classifier is statistically significant, and the null hypothesis is rejected (see Appendix). We used 5-folds, and 500 permutations yielding a p-value $= (C + 1)/(500 + 1)$, where C is the number of permutations whose scores are greater than or equal to the true score. The minimum achievable p-value in this setting is $p_{\min} \stackrel{\text{def}}{=} 1/(500 + 1) \approx 0.002$, which corresponds to the case where the constructed classifier is such that none of the constructed classifiers with shuffled labels achieved a better score. Fig.12 shows the results of the permutation test where solid vertical lines represent the scores without label permutation (the AUC values are given in the horizontal axis of each plot) and the distribution of scores for each permutation is shown as a histogram (i.e., the null-distribution). The figure confirms that constructed classifiers for all subjects achieved remarkable performances as p-value $= p_{\min}$ for all subjects.

Fig.13 shows the real-time performance of the BCI system in terms of the AUC score. It is remarkable to observe that the BCI performance in terms of AUC is greater than 80% for all participants. On average, all users were able to exert seven control commands per minute to interact with and control the telepresence robot. Although the primary goal of the present work is not enhancing BCI decoders, it is interesting to observe that the overall performance of the constructed BCI system is close to the state-of-the-art reported in the BCI literature [1], [35], [36], [55], [56].

IV. DISCUSSION

The fusion of BCIs and PbD approaches is crucial to enable the telepresence robot to be more flexible and applicable to a broader range of situations [57]. The PbD allows the robot being pre-trained to perform essential activities of daily life, while BCI enables high-level control of the robot. The end user, potentially a paralyzed person, would issue commands to the remote robot through a BCI system to choose tasks to be performed and maneuver its movement. This scenario can be described as a shared-control strategy [58], [59] where the user makes a high-level decision using the BCI system, while the low-level implementation complexities are regulated by PbD methodologies, satisfying the principles of

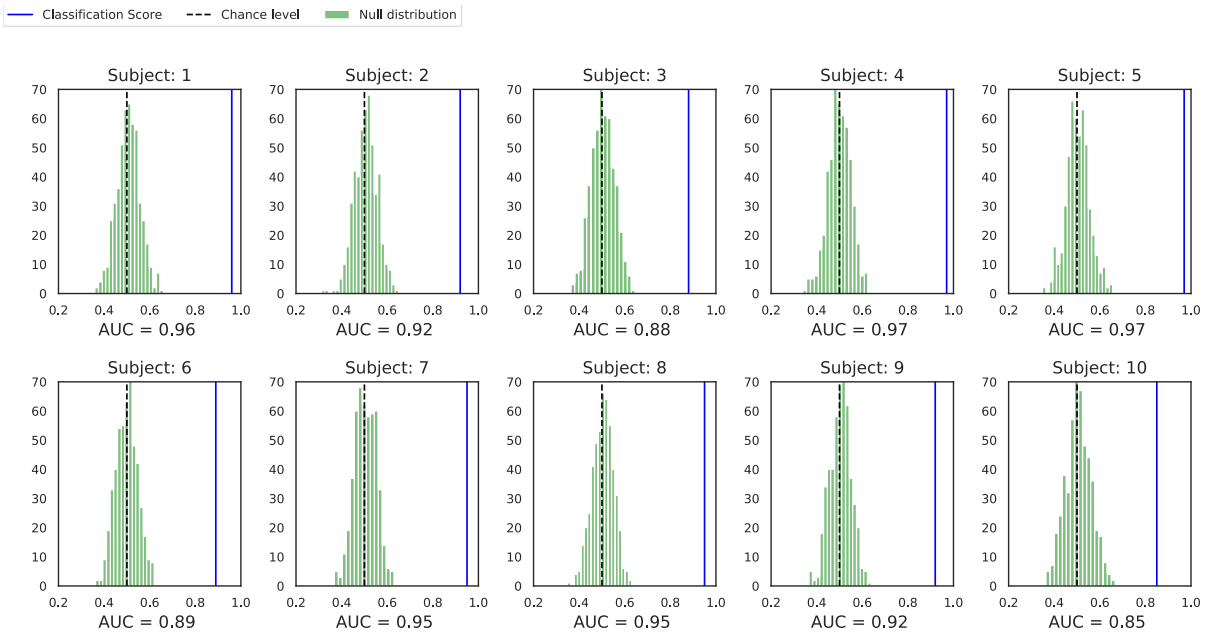

FIGURE 12. Permutation test results for the classifier obtained using 5-fold cross-validation. The (blue) solid vertical lines represent the classification scores without label permutation (the AUC values are given in the horizontal axis of each plot). The dashed lines represent the chance level classification score and the score distribution of the permutation test s shown as a histogram. The *p*-values for the test statistic (i.e., observed AUCs) were $p_{\min} \approx 0.002$ for all subjects.

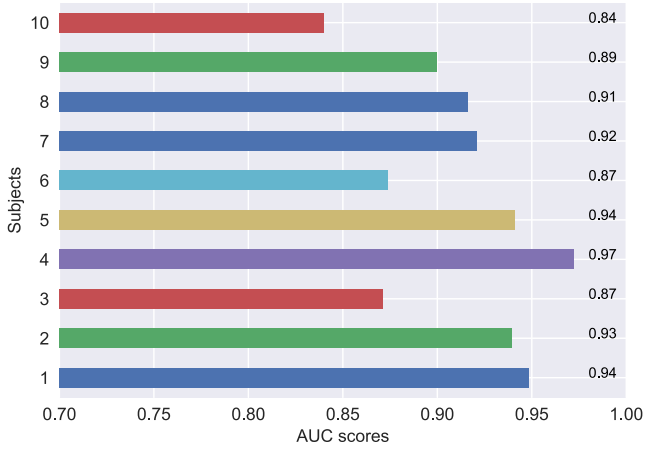

FIGURE 13. Real-time performance of the BCI system in terms of AUC score. The participants performed control of a humanoid robot in five tasks. The control commands consisted of moving the robot forward, backward, turning right and left and greeting "say Hello".

unconscious human learning and cognitive decision-making model. Indeed, the robot learning part could be facilitated by the caretakers of disabled individuals, who most likely would not be experts in robotics/programming. The PbD enables such people to train the robots, which would be then used by disabled individuals, giving them more autonomy in their ADL activities.

Handling BCI events from one task to another is not easy, especially when the tasks constitute a wide range of activities. Here, we define the event-driven programming approach with event generating clients (see Fig. 1). In the context of a shared control method, the server collects all the events to define the current state of the environment, including the telepresence robot's state. The approach is asynchronous, and the arrival of new events triggers the processing of new event-based situation. As soon as a new command is provided, the shared control transmits it to the client that executes the action via the server. This work not only improves the pure BCI-actuated Humanoid robot but also extends the shared-control strategy discussed in [17]. The proposed PbD based BCI system can alleviate the complexities of a telepresence robot that could lead to the eventual confluence of humans in need and the telepresence robots. We contend that the fusion of BCI and PbD systems allows disabled populations to interact with an environment more efficiently. Moreover, diverse tasks could be pre-learned by the telepresence robot via observing someone performing the function of interest.

V. CONCLUSION

This paper was motivated by the difficulties that motor-impaired people face in performing even quite simple tasks and interacting with others. Improving their quality of life using new technologies such as robotics and brain-computer interface is the main outcome of the presented work. We validated a novel application of programming by demonstration (PbD) learning techniques coupled with Brain-Computer Interfaces (BCI) to control a remote humanoid robot for telepresence. The proposed telepresence system aims to support severely motor disabled people to interact with their relatives or friends remotely and perform physical activities of daily life. We have shown that the developed BCI system achieves high accuracies in decoding participants mental intent and, at the same time, the PbD approach enables the robot to learn and perform complex grasping and moving

tasks. Although the study has enrolled healthy individuals, the system is applicable to target populations with motor paralysis, which remains to be our next logical step. The key contributions of the proposed work include constructing a novel BCI–to– telepresence system to control a humanoid robot that is expected to improve the social aspects of people with severe paralysis. The focus of our study is warranted because such health innovations lead to augmenting mobility and interaction capability for people in need.

APPENDIX

Test 1 (Permutation test) Let $D = \{(X_i, y_i)\}_{i=1}^{n}$ be the original data set and let π be a permutation of n- elements. One randomized version D' of D is obtained by applying the permutation π on the labels $D' = \{(X_i, \pi(y)_i)\}_{i=1}^{n}$. Compute the p-value as in Definition 1. **Definition 1 (Permutation-based p-value)** Let $\widehat{D}$ be a set of k randomized versions of D' of the original data D sampled from a given null distribution, and AUC metric $AUC(f, D)$. The empirical p-value for the classifier f is calculated as follows [54]:

$$p = \frac{\left|\left\{D' \in \widehat{D} : AUC\left(f, D'\right) \geq AUC(f, D)\right\}\right| + 1}{k + 1} \tag{18}$$

The empirical p-value of Definition 1 measures how likely the observed accuracy would be obtained by chance, only because the classifier identified in the training phase a pattern that happened to be random. Therefore, if the p-value is smaller than a certain threshold, for example, $\alpha = 0.05$ — we can say the classifier is significant under the given null hypothesis, that is, the null hypothesis is rejected.

REFERENCES

[1] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based brain–computer interfaces: A 10 year update," *J. Neural Eng.*, vol. 15, no. 3, 2018, Art. no. 031005.

[2] J. Wolpaw and E. W. Wolpaw, *Brain-Computer Interfaces: Principles and Practice*. New York, NY, USA: OUP, 2012.

[3] A. Ricco, L. Simione, F. Schettini, A. Pizzimenti, M. Inghilleri, M. O. Belardinelli, D. Mattia, and F. Cincotti, "Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis," *Front. Hum. Neurosci.*, vol. 7, no. 732, Nov. 2013.

[4] B. Abibullaev, "Learning suite of kernel feature spaces enhances SMR-based EEG-BCI classification," in *Proc. 5th Int. Winter Conf. Brain-Comput. Interface (BCI)*, Jan. 2017, pp. 55–59.

[5] Z. Zhang, F. Duan, J. Solé-Casals, J. Dinarés-Ferran, A. Cichocki, Z. Yang, and Z. Sun, "A novel deep learning approach with data augmentation to classify motor imagery signals," *IEEE Access*, vol. 7, pp. 15945–15954, 2019.

[6] G. E. Fabiani, D. J. McFarland, J. R. Wolpaw, and G. Pfurtscheller, "Conversion of EEG activity into cursor movement by a brain-computer interface (BCI)," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 12, no. 3, pp. 331–338, Sep. 2004.

[7] L. Bi, X.-A. Fan, and Y. Liu, "EEG-based brain-controlled mobile robots: A survey," *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 2, pp. 161–176, Mar. 2013.

[8] F. Galán, M. Nuttin, E. Lew, P. W. Ferrez, G. Vanacker, J. Philips, and J. D. R. Millán, "A brain-actuated wheelchair: Asynchronous and non-invasive brain–computer interfaces for continuous control of robots," *Clinical Neurophysiol.*, vol. 119, no. 9, pp. 2159–2169, 2008.

[9] A. S. Widge, C. T. Moritz, and Y. Matsuoka, "Direct neural control of anatomically correct robotic hands," in *Brain-Computer Interfaces*. London, U.K.: Springer, 2010, pp. 105–119.

[10] F. Duan, D. Lin, W. Li, and Z. Zhang, "Design of a multimodal EEG-based hybrid BCI system with visual servo module," *IEEE Trans. Auton. Mental Develop.*, vol. 7, no. 4, pp. 332–341, Dec. 2015.

[11] Y. Chae, J. Jeong, and S. Jo, "Toward brain-actuated humanoid robots: Asynchronous direct control using an EEG-based BCI," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1131–1144, Oct. 2012.

[12] R. Leeb, L. Tonin, M. Rohm, L. Desideri, T. Carlson, and J. D. R. Millán, "Towards independence: A BCI telepresence robot for people with severe motor disabilities," *Proc. IEEE*, vol. 103, no. 6, pp. 969–982, Jun. 2015.

[13] N. A. Bhagat, A. Venkatakrishnan, B. Abibullaev, E. J. Artz, N. Yozbatiran, A. A. Blank, J. French, C. Karmonik, R. G. Grossman, M. K. O'Malley, G. E. Francisco, and J. L. Contreras-Vidal, "Design and optimization of an EEG-based brain machine interface (BMI) to an upper-limb exoskeleton for stroke survivors," *Frontiers Neurosci.*, vol. 10, pp. 1–17, Mar. 2016.

[14] R. Spataro, A. Chella, B. Allison, M. Giardina, R. Sorbello, S. Tramonte, C. Guger, and V. La Bella, "Reaching and grasping a glass of water by locked-in als patients through a bci-controlled humanoid robot," *Frontiers Hum. Neurosci.*, vol. 11, p. 68, Mar. 2017.

[15] I. Kawaguchi, Y. Kodama, H. Kuzuoka, M. Otsuki, and Y. Suzuki, "Effect of embodiment presentation by humanoid robot on social telepresence," in *Proc. 4th Int. Conf. Hum. Agent Interact.*, 2016, pp. 253–256.

[16] C. Escolano, J. M. Antelis, and J. Minguez, "A telepresence mobile robot controlled with a noninvasive brain–computer interface," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 793–804, Jun. 2012.

[17] L. Tonin, R. Leeb, M. Tavella, S. Perdikis, and J. D. R. Millán, "The role of shared-control in BCI-based telepresence," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2010, pp. 1462–1466.

[18] A. Chella, E. Pagello, E. Menegatti, R. Sorbello, S. M. Anzalone, F. Cinquegrani, L. Tonin, F. Piccione, K. Prifitis, C. Blanda, E. Buttita, and E. Tranchina, "A BCI teleoperated museum robotic guide," in *Proc. Int. Conf. Complex, Intell. Softw. Intensive Syst.*, Mar. 2009, pp. 783–788.

[19] C. Escolano, A. R. Murguialday, T. Matuz, N. Birbaumer, and J. Minguez, "A telepresence robotic system operated with a P300-based brain-computer interface: Initial tests with ALS patients," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol.*, Aug./Sep. 2010, pp. 4476–4480.

[20] S.-J. Yun, M.-C. Lee, and S.-B. Cho, "P300 BCI based planning behavior selection network for humanoid robot control," in *Proc. 9th Int. Conf. Natural Comput. (ICNC)*, Jul. 2013, pp. 354–358.

[21] Y. Chae, S. Jo, and J. Jeong, "Brain-actuated humanoid robot navigation control using asynchronous brain-computer interface," in *Proc. 5th Int. IEEE/EMBS Conf. Neural Eng. (NER)*, Apr./May 2011, pp. 519–524.

[22] W. Li, C. Jaramillo, and Y. Li, "Development of mind control system for humanoid robot through a brain computer interface," in *Proc. 2nd Int. Conf. Intell. Syst. Design Eng. Appl. (ISDEA)*, Jan. 2012, pp. 679–682.

[23] A. Thobbi, R. Kadam, and W. Sheng, "Achieving remote presence using a humanoid robot controlled by a non-invasive BCI device," *Int. J. Artif. Intell. Mach. Learn.*, vol. 10, pp. 41–45, Oct. 2010.

[24] A. Güneysu and H. L. Akin, "An SSVEP based BCI to control a humanoid robot by using portable EEG device," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 6905–6908.

[25] D. Petit, P. Gergondet, A. Cherubini, and A. Kheddar, "An integrated framework for humanoid embodiment with a BCI," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 2882–2887.

[26] M. Bryan, J. Green, M. Chung, L. Chang, R. Scherer, J. Smith, and R. P. Rao, "An adaptive brain-computer interface for humanoid robot control," in *Proc. 11th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, Oct. 2011, pp. 199–204.

[27] E. Tidoni, P. Gergondet, A. Kheddar, and S. M. Aglioti, "Audio-visual feedback improves the BCI performance in the navigational control of a humanoid robot," *Frontiers Neurorobot.*, vol. 8, p. 20, Jun. 2014.

[28] R. Fazel-Rezai and W. Ahmad, *P300-based Brain-Computer Interface Paradigm Design*. Rijeka, Croatia: InTech, 2011.

[29] L. A. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalograph. Clin. Neurophysiol.*, vol. 70, no. 6, pp. 510–523, Dec. 1988.

[30] C. Başar-Eroglu, E. Başar, T. Demiralp, and M. Schürmann, "P300-response: Possible psychophysiological correlates in delta and theta frequency channels. A review," *Int. J. Psychophysiol.*, vol. 13, no. 2, pp. 161–179, 1992.

[31] S. van Gaal, F. P. de Lange, and M. X. Cohen, "The role of consciousness in cognitive control and decision making," *Frontiers Hum. Neurosci.*, vol. 6, p. 121, May 2012.

[32] B. Saduanov, T. Alizadeh, J. An, and B. Abibullaev, "Trained by demonstration humanoid robot controlled via a bci system for telepresence," in *Proc. 6th Int. Conf. Brain-Comput. Interface*, Jan. 2018, pp. 91–94.
[33] S. Calinon, "Robot learning with task-parameterized generative models," in *Robotics Research*. Cham, Switzerland: Springer, 2018, pp. 111–126.
[34] D. Talsma and M. G. Woldorff, "Methods for the estimation and removal of artifacts and overlap," in *Event-Related Potentials: A Methods Handbook*. Cambridge, MA, USA: MIT Press, 2005, p. 115.
[35] B. Abibullaev and A. Zollanvari, "Learning discriminative spatiospectral features of ERPs for accurate brain-computer interfaces," *IEEE J. Biomed. Health Inform.*, to be published. doi: 10.1109/JBHI.2018.2883458.
[36] J. Farquhar and N. J. Hill, "Interactions between pre-processing and classification methods for event-related-potential classification," *Neuroinformatics*, vol. 11, no. 2, pp. 175–192, 2013.
[37] H. Cecotti and A. J. Ries, "Best practice for single-trial detection of event-related potentials: Application to brain-computer interfaces," *Int. J. Psychophysiol.*, vol. 111, pp. 156–169, Jan. 2017.
[38] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. Hämäläinen, "MEG and EEG data analysis with MNE-python," *Frontiers Neurosci.*, vol. 7, p. 267, Dec. 2013.
[39] B. Abibullaev, Y. Orazayev, and A. Zollanvari, "Novel spatiospectral features of ERPs enhances brain-computer interfaces," in *Proc. 7th Int. Winter Conf. Brain-Comput. Interface*, Gangwon, Korea, Feb. 2019, pp. 1–4.
[40] S. Le Cessie and J. C. Van Houwelingen, "Ridge estimators in logistic regression," *Appl. Statist.*, vol. 41, no. 1, pp. 191–201, 1992.
[41] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Amsterdam, The Netherlands: Elsevier, 2017.
[42] J. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, pp. 131–163, Nov. 1997.
[43] A. R. Webb, *Statistical Pattern Recognition*, 1st ed. Hoboken, NJ, USA: Wiley, 2002.
[44] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.
[45] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Heidelberg, Germany: Springer, 2010, pp. 177–186.
[46] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*. Heidelberg, Germany: Springer, 2012, pp. 421–436.
[47] N. T. Hintzen, G. J. Piet, and T. P. Brunel, "Improved estimation of trawling tracks using cubic Hermite spline interpolation of position registration data," *Fisheries Res.*, vol. 101, nos. 1–2, pp. 108–115, 2010.
[48] J. Luo, K. Ying, and J. Bai, "Savitzky–Golay smoothing and differentiation filter for even number data," *Signal Process.*, vol. 85, no. 7, pp. 1429–1434, 2005.
[49] T. Alizadeh, "Statistical learning of task modulated human movements through demonstration," Ph.D. dissertation, Dept. Adv. Robot., Italian Inst. Technol., Genova, Italy, 2014.
[50] M. S. Malekzadeh, J. F. Queißer, and J. J. Steil, "Multi-level control architecture for bionic handling assistant robot augmented by learning from demonstration for apple-picking," *Adv. Robot.*, vol. 33, no. 9, pp. 469–485, 2019.
[51] T. Alizadeh, S. Calinon, and D. G. Caldwell, "Learning from demonstrations with partially observable task parameters," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Mar./Jun. 2014, pp. 3309–3314.
[52] T. Alizadeh and M. Malekzadeh, "Identifying the relevant frames of reference in programming by demonstration using task-parameterized Gaussian mixture regression," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2016, pp. 453–458.
[53] T. Alizadeh and B. Saduanov, "Robot programming by demonstration of multiple tasks within a common environment," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Nov. 2017, pp. 608–613.
[54] M. Ojala and G. C. Garriga, "Permutation tests for studying classifier performance," *J. Mach. Learn. Res.*, vol. 11, pp. 1833–1863, Jun. 2010.
[55] P. Yuan, X. Gao, B. Allison, Y. Wang, G. Bin, and S. Gao, "A study of the existing problems of estimating the information transfer rate in online brain–computer interfaces," *J. Neural Eng.*, vol. 10, no. 2, 2013, Art. no. 026014.
[56] D. J. Krusienski, M. Grosse-Wentrup, and F. Galán, D. Coyle, K. J. Miller, E. Forney, and C. W. Anderson, "Critical issues in state-of-the-art brain–computer interface signal processing," *J. Neural Eng.*, vol. 8, no. 2, 2011, Art. no. 025002.
[57] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*. Heidelberg, Germany: Springer, 2008, pp. 1371–1394.
[58] G. R. Müller-Putz, C. Breitwieser, F. Cincotti, R. Leeb, M. Schreuder, F. Leotta, M. Tavella, L. Bianchi, A. Kreilinger, A. Ramsay, M. Rohm, M. Sagebaum, L. Tonin, C. Neuper, and J. D. R. Millán, "Tools for brain-computer interaction: A general concept for a hybrid BCI," *Frontiers Neuroinform.*, vol. 5, p. 30, Nov. 2011.
[59] T. Carlson and J. Del R Millan, "Brain-controlled wheelchairs: A robotic architecture," *IEEE Robot. Autom. Mag.*, vol. 20, no. 1, pp. 65–73, Mar. 2013.

BERDAKH ABIBULLAEV (M'12) received the M.Sc. and Ph.D. degrees in electronic engineering from Yeungnam University, South Korea, in 2006 and 2010, respectively. He held researcher positions at the Daegu Gyeongbuk Institute of Science and Technology, from 2010 to 2013, and the Neurology Department, Samsung Medical Center, Seoul, South Korea, from 2013 to 2014. In 2014, he received National Institute of Health (NIH), USA Postdoctoral Research Fellowship to join a multi-institutional research project between the University of Houston and Texas Medical Center in developing neural interfaces for post-stroke patients. He is currently an Assistant Professor with the Robotics Department, Nazarbayev University, Nur-Sultan, Kazakhstan. His research interests include developing brain–computer interfaces, signal processing, and machine learning algorithms for brain state decoding.

AMIN ZOLLANVARI (M'10–SM'19) received the B.Sc. and M.Sc. degrees in electrical engineering from Shiraz University, Iran, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, in 2010. He held a postdoctoral position at the Harvard Medical School and Brigham and Women's Hospital, Boston, MA, from 2010 to 2012. He joined the Department of Statistics, Texas A&M University, as an Assistant Research Scientist, from 2012 to 2014. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Nazarbayev University, Nur-Sultan, Kazakhstan. His research interests include machine learning, statistical signal processing, and biomedical informatics.

BATYRKHAN SADUANOV received the B.Sc. degree in robotics and mechatronics from Nazarbayev University, Nur-Sultan, Kazakhstan, in 2017. He is currently a Software Engineer with Computer Vision Technologies, Nur-Sultan.

TOHID ALIZADEH received the B.Sc. degree in electrical engineering from the Sahand University of Technology, Tabriz, Iran, the M.Sc. degree in automation and instrumentation engineering from the Petroleum University of Technology, Tehran, Iran, and the Ph.D. degree in robotics, cognition and interaction technologies from the Italian Institute of Technology (IIT), Genoa, Italy, in 2004, 2007 and 2014, respectively. He held an instructor position in Bonab University, Bonab, Iran, from 2009 to 2011, before starting his Ph.D. studies. He is currently an Assistant Professor with the School of Science and Technology, Robotics and Mechatronics Department, Nazarbayev University, Nur-Sultan, Kazakhstan. His research interests include robotics, programming by demonstration, and statistical machine learning.