Journal of Neural Engineering

PAPER • **OPEN ACCESS**

A shared robot control system combining augmented reality and motor imagery brain–computer interfaces with eye tracking

To cite this article: Arnau Dillen *et al* 2024 *J. Neural Eng.* **21** 056028

Journal of Neural Engineering

PAPER

A shared robot control system combining augmented reality and motor imagery brain–computer interfaces with eye tracking

Arnau Dillen[1,2,3], **Mohsen Omidi**[3,4], **Fakhreddine Ghaffari**[2], **Bram Vanderborght**[3,4], **Bart Roelands**[1,3], **Olivier Romain**[2], **Ann Nowé**[5] and **Kevin De Pauw**[1,3,*]

[1] Human Physiology and Sports Physiotherapy Research Group, Vrije Universiteit Brussel, 1050 Brussels, Belgium
[2] Équipes Traitement de l'Information et Systémes, CY Cergy Paris Université, École Nationale Supérieure de l'Électronique et de ses Applications (ENSEA), Centre national de la recherche scientifique (CNRS), UMR 8051, 95000 Cergy, France
[3] Brussels Human Robotics Research Center (BruBotics), Vrije Universiteit Brussel, 1050 Brussels, Belgium
[4] imec, 1050 Brussels, Belgium
[5] Artificial Intelligence Research Group, Vrije Universiteit Brussel, 1050 Brussels, Belgium
[*] Author to whom any correspondence should be addressed.

E-mail: kevin.de.pauw@vub.be

Keywords: shared robot control, brain–computer interface, augmented reality, motor imagery, electroencephalogram, user evaluation, eye tracking

OPEN ACCESS

RECEIVED
10 June 2024

REVISED
30 August 2024

ACCEPTED FOR PUBLICATION
25 September 2024

PUBLISHED
8 October 2024

Abstract

Objective. Brain–computer interface (BCI) control systems monitor neural activity to detect the user's intentions, enabling device control through mental imagery. Despite their potential, decoding neural activity in real-world conditions poses significant challenges, making BCIs currently impractical compared to traditional interaction methods. This study introduces a novel motor imagery (MI) BCI control strategy for operating a physically assistive robotic arm, addressing the difficulties of MI decoding from electroencephalogram (EEG) signals, which are inherently non-stationary and vary across individuals. *Approach*. A proof-of-concept BCI control system was developed using commercially available hardware, integrating MI with eye tracking in an augmented reality (AR) user interface to facilitate a shared control approach. This system proposes actions based on the user's gaze, enabling selection through imagined movements. A user study was conducted to evaluate the system's usability, focusing on its effectiveness and efficiency. *Main results*. Participants performed tasks that simulated everyday activities with the robotic arm, demonstrating the shared control system's feasibility and practicality in real-world scenarios. Despite low online decoding performance (mean accuracy: 0.52 9, F1: 0.29, Cohen's Kappa: 0.12), participants achieved a mean success rate of 0.83 in the final phase of the user study when given 15 min to complete the evaluation tasks. The success rate dropped below 0.5 when a 5 min cutoff time was selected. *Significance*. These results indicate that integrating AR and eye tracking can significantly enhance the usability of BCI systems, despite the complexities of MI-EEG decoding. While efficiency is still low, the effectiveness of our approach was verified. This suggests that BCI systems have the potential to become a viable interaction modality for everyday applications in the future.

1. Introduction

Brain–computer interface (BCI) technology has been gaining increased attention due to its potential to transform human-computer interaction. From enabling communication for those with severe motor disabilities [1] to teleoperation of industrial robots for healthy individuals [2], the applications of BCI are numerous, enabling previously impossible or impractical user scenarios [3]. This is especially useful in environments where movement is restricted and other interaction modalities cannot be used. One application of interest is the control of assistive robotic devices that aim to support users who suffer from motor-function impairing disabilities in their daily lives [4].

Typically, BCI systems measure the neural activity of the user and try to determine their intention from the chosen neural signal using machine learning (ML) methods [5]. A commonly employed signal is the electroencephalogram (EEG) which measures the electrical activity of the brain using electrodes on the surface of the head [6]. Enabling intention detection using BCI requires a mental task that remains consistent over time and across individuals. This is especially important as EEG is a non-stationary signal that has a large inter- and intra-individual variability [7].

Multiple modalities are available for EEG-based BCI applications [8]. Two of the most commonly used modalities are steady-state visually evoked potential (SSVEP) and motor imagery (MI). SSVEP is brain activity triggered by viewing a steadily flickering visual stimulus [9] and is often used for BCI speller applications that allow users to type text using a virtual keyboard where the keys are stimuli flickering at different frequencies [10]. However, this method does not allow users to initiate commands spontaneously. Additionally, other non-BCI modalities like eye tracking can achieve the same type of interaction more reliably, without needing to acquire EEG data for training decoding models. However, hybrid systems have been proposed to improve the reliability of target identification [11].

MI is defined as 'the mental simulation of an action without the corresponding motor output' [12], meaning it involves imagining a movement without actually performing it. By linking specific imagined movements to device commands, a BCI control system can be created. However, decoding MI signals is complex, requiring extensive training data for accurate decoding and frequent recalibration due to the non-stationarity of EEG [8].

Thus, strategies are necessary to mitigate the issues of MI decoding without sacrificing the advantages of BCI. One approach that can offer a possible solution is to integrate BCI with a shared control system [13]. Shared control systems use sensors to determine the current state of the environment and a semi-autonomous robot [14]. The control system can use this information to propose high-level actions to the user. This limits the user interaction to selecting objects from the environment to interact with and choosing one of the available actions.

The research question for this paper is therefore what the most suitable design is for a shared control system that uses MI-based BCI for the operation of a physically assistive robotic arm. We chose MI as our BCI modality because it can provide rich multimodal interaction when combined with eye tracking. The objective of this research is to design and validate a proof-of-concept BCI control system in scenarios that simulate real-world conditions. Our system integrates an augmented reality (AR) UI with eye tracking and MI BCI to build a shared control system for real-life applications. The control system proposes actions based on the object the user is currently looking at thanks to spatial awareness and object detection allowing the user to select the desired action with MI.

To investigate the real-world usability of the developed control system, a user study was devised balancing the need for fast iteration required of software development and the structured approach required in human studies. To assess the real-world usability of the control system, participants completed tasks where they had to operate a robotic arm using our control system. The usability of a control system depends on its effectiveness, efficiency, and user experience [15]. In the field of BCI research, usability and user-centered design are gaining prominence as researchers strive to connect lab experiments with real-world applications [16, 17]. The current study focuses on the effectiveness and efficiency aspects of the control system. A detailed analysis of overall system usability including user experience and a comparison with a control system variant that relies solely on eye tracking is presented in [18].

2. Materials and methods

2.1. Control strategy

The proposed control strategy provides a flexible approach, allowing users to select items, such as user interface (UI) menu options or physical objects in the real world, using their gaze, and then choose an associated action through MI. We evaluate this control strategy by applying it to the specific use case of operating a robotic arm. An overview of the shared control strategy utilized by our BCI control system is depicted in figure 1.

To interact with an object in the environment, the user gazes at the object they want to interact with. The shared control system then employs an internal world model that is based on spatial mapping and computer vision to determine the type, location, and state of the objects. For this proof-of-concept, this is simulated with an AR environment. Based on these factors, the system proposes multiple actions to the user.

To select an action, the user must think about the associated movement while maintaining their gaze on the object for 2 s. The decoding pipeline subsequently processes the EEG signal using a sliding window. When one of the MI classes has been predicted n times, where n is the threshold for decision-making, the system prompts the user with a dialog to either accept the decoded movement or reject the decoding result. For this study, the threshold was set to 5, as this value was determined to provide an optimal balance between quick decision-making and reliable decoding in preliminary testing.

If the user accepts the decoding result, the robot executes the associated action. If the user rejects the result, decoding resumes, and the user imagines the intended movement again. This method prevents the

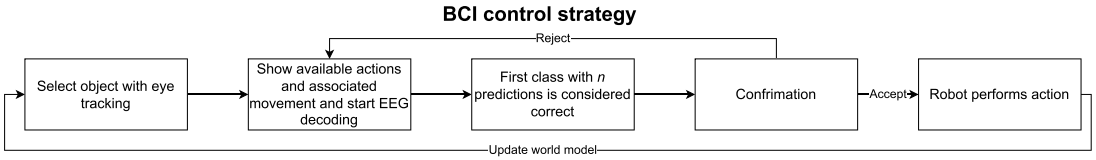

Figure 1. The control strategy of the MI BCI control system. The system starts in the object selection stage where the user can select an object by looking at it for 2 s. A menu showing which movement the user should imagine to select an action appears and EEG decoding is initiated. When an MI class is predicted *n* times, this is considered as the intended movement and a menu asking the user to confirm the action is shown. If the user rejects the action, the system goes back to the decoding stage. If the user accepts the action, the robot performs this action and the system returns to the selection stage after updating the world model.

robot from performing an unintended action, avoiding the need for the user to cancel it. Once the robot completes the selected action, the system returns to object selection mode.

2.2. Hardware

To achieve the intended functionality, several hardware components need to work jointly. An overview of the necessary hardware components and how they communicate with each other is shown in figure 2(a).

A suitable EEG device is necessary to measure the user's neural activity. The only requirements for this device are that it should be compatible with the headwear used for AR display and provide an application programming interface that enables access to the raw real-time EEG data stream from our custom decoding software. A dry electrode system would make the donning of the device more convenient in daily use cases, and active electrodes are recommended to ensure adequate signal quality. The device should also preferably be mobile to avoid limiting the use of the BCI system to the user's home.

An AR head-mounted display (HMD) that supports eye-tracking is necessary for displaying the UI and selecting objects. The device should preferably use waveguide-based AR technology to avoid isolating the user from reality. A pass-through AR display can also be used if it can accurately reproduce the real-world environment. The device must be portable and ideally stand-alone for a fully mobile experience. Easy access to built-in sensor and camera data would also enable advanced spatial computing to obtain an accurate representation of the environment.

The assistive robot should be a cobot that can safely operate around humans without risk of injury. It should also be able to pick up objects and precisely place them in a specific position. The software to send robot commands should be compatible with the HMD to enable a stand-alone system.

Finally, the computational hardware used for EEG decoding should be powerful enough to run the chosen decoding model and be able to communicate with both the robot and the HMD. Ideally, this hardware is portable and has a long autonomy. If the decoding software can run on the same hardware as the HMD control software, that would even simplify the system more.

The hardware that was selected for this study is shown in figure 2(b).

This study used the Smarting ProX from mbt (mBrainTrain; Serbia) for EEG acquisition. This device uses a cap equipped with 64 active gel electrodes positioned according to the 10–20 system. It can wirelessly connect to the recording device over Bluetooth and can sample up to 4000 Hz. For this study, the sampling rate was limited to 250 Hz to reduce the computational cost of real-time processing of the data stream.

Microsoft™ HoloLens 2.0 was chosen for the head-mounted AR display. It uses waveguide-based AR and provides the necessary eye-tracking and spatial awareness functionality through its sensors. This device was selected for its mature hardware and extensive software tools. This facilitated the development of the AR environment and UI [19].

The Franka Research robot was employed as the cobot in this experiment. This robotic arm offers 7 degrees of freedom and is equipped with a gripper for picking up objects. Safety is ensured by the robot's built-in sensors, which halts its movement upon detecting a collision [20].

Real-time EEG decoding was handled by a laptop personal computer with a 6-core Intel(R) Core(TM) i7-10 850 H CPU with a clock speed of 2.70 GHz. The computer has 16 GB of working memory and an NVIDIA GeForce RTX 2070 with a Max-Q design graphical processing unit. The computer is connected to the robot with an Ethernet connection and communicates with HoloLens through a USB-C cable.

2.3. Software

Two distinct software components were developed for the BCI control system, the decoding application that runs on a dedicated computer and a UI application deployed on the HoloLens. The communication between the decoding application and the UI is handled by the ZeroMQ (ZMQ) messaging software. A publish/subscribe strategy was used for communication where one component (the publisher) sends

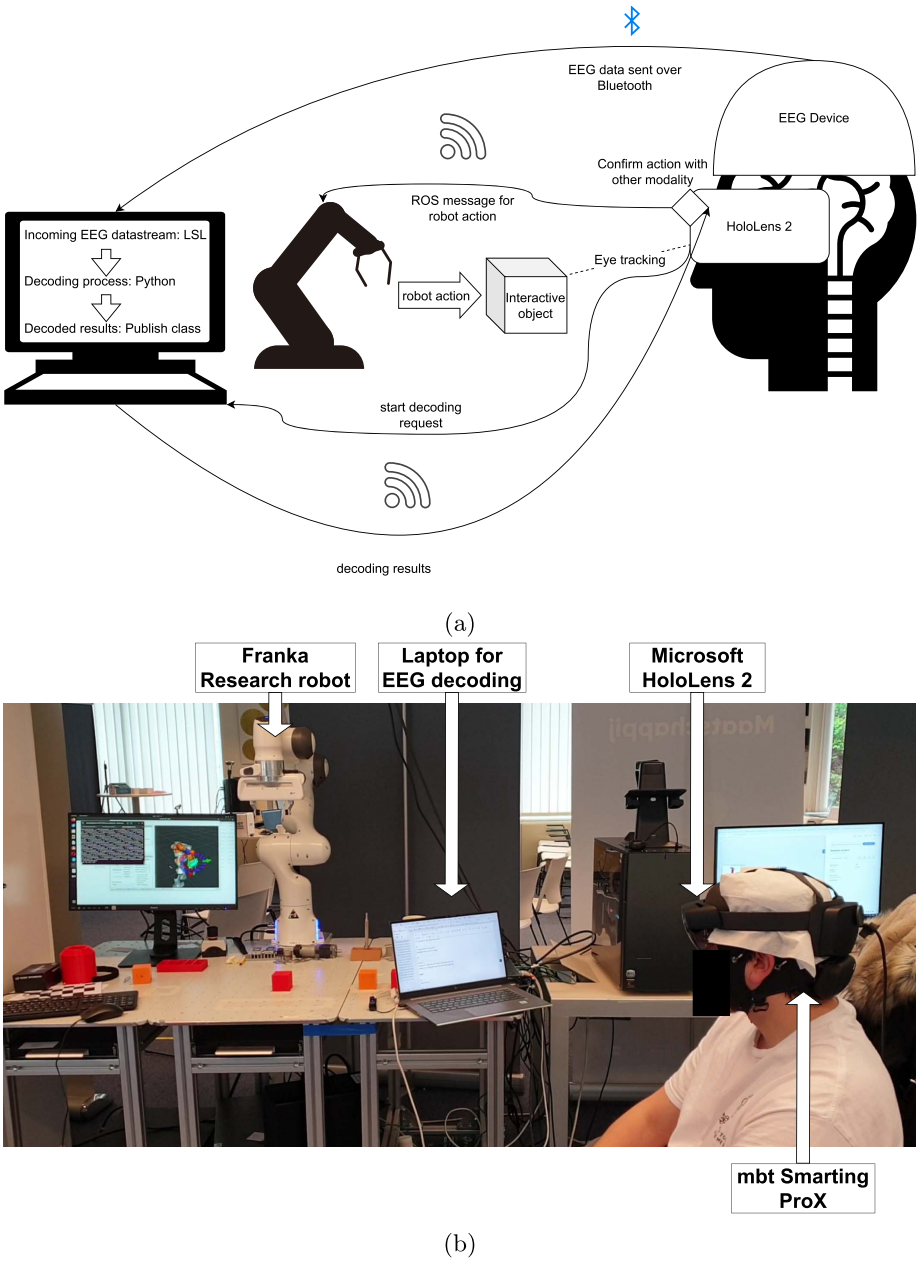

Figure 2. Schematic representation of the different hardware components (a) and the hardware setup of the BCI control system that was used for the proof-of-concept (b). The EEG device is connected via Bluetooth to a decoding computer that monitors the data stream. The HoloLens 2 headset and the robot arm are connected to the same local network as the decoding computer. This local network facilitates message exchange between the computer and the HoloLens, as well as sending messages to the robot arm when an action is selected.

messages to a message queue provided by ZMQ while the other listens (the subscriber) for messages.

The intent of the sent message is determined by assigning a topic to the queue. Two topics are used in our software. The *decoding control* topic is responsible for passing messages concerning starting and stopping the decoding process and the *predictions* topic handles the communication of decoding results to the UI process. The decoding application subscribes to the *decoding control* topic and publishes results to the *predictions* topic while the UI publishes to the former and subscribes to the latter.

Sending commands to the robot is handled by the robot operating system (ROS) using the provided transport control protocol (TCP) messaging functionality [21]. We used Unity-Technologies-ROS-TCP-Connector, developed by the Simens company as the ROS# package, ensuring stable communication between Unity3D and ROS [22]. To execute each action/movement by the robot, we used the MoveIt package, which made us able to prerecord each movement of the robot, for each action that the user could select [23].

Figure 3 provides a schematic overview of the BCI control system's software architecture.

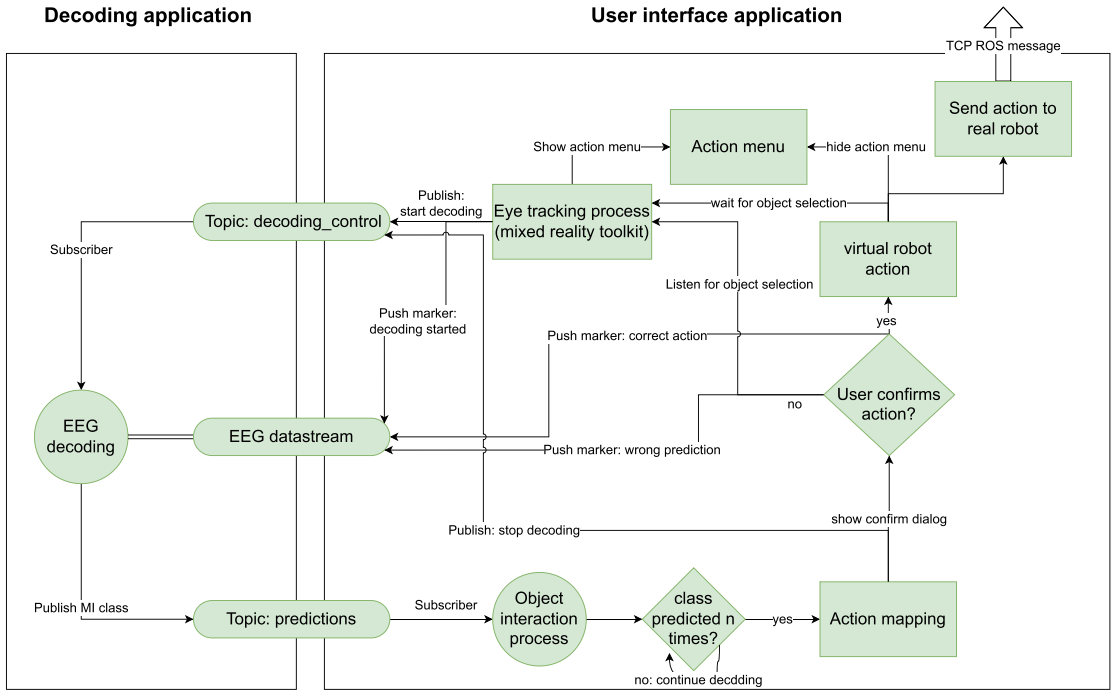

Figure 3. Software architecture of the brain–computer interface (BCI) control system. Two separate components handle real-time electroencephalogram (EEG) decoding and user interface interaction, communicating through a publish/subscribe model. Both processes can access the EEG data stream to sample windows for decoding and push markers when the system changes states respectively. When an action is selected it is sent to the robot controller software using the Robot Operating System (ROS) transport carriere protocol (TCP) messaging interface.

Our custom decoding application handles real-time EEG decoding and sends decoding results to the UI, which sends the appropriate command to the robot motion control software. In its initial state, the eye-tracking process monitors the user's gaze and displays the menu depicting which movement the user should think of to trigger a specific action. Meanwhile, the decoding process waits for the message to start decoding, sent by the UI process when the user's gaze intersects with an interactive object.

The decoding process will keep decoding and sending the results to the UI process until *n* predictions are achieved for the same class. When the UI process receives the *n*th message, it sends the stop decoding message to the decoding process and displays the confirmation dialog. If the user accepts the decoded action, the appropriate commands are sent to the robot via ROS TCP to perform the chosen action. The program subsequently returns to the initial state until the user selects a new object with their gaze. If the user rejects the result, the start decoding message is sent again, and decoding resumes. This is repeated until the user accepts the decoding result. When state changes occur, the UI application pushes markers to the EEG stream to indicate which event occurred at what time.

2.3.1. Real-time EEG decoding application

The real-time EEG decoding component of the control system was implemented with the Python programming language. The MNE library [24] was used for operations relating to EEG data processing such as reading data files, filtering, and epoching among others. The lab streaming layer (LSL; [25]) software was used with its Python client PyLSL to access the real-time EEG data stream and ensure all data are synchronized. The implementation of ML models was achieved with the Scikit Learn (SKLearn) library [26].

Decoding is achieved with a sliding window of 2 s with a stride of 0.25 s over the EEG data stream. When the start decoding message is received, the decoding process starts sampling chunks of EEG data from the LSL stream. The size of the sampled chunk is determined by the *chunk_prop* parameter by multiplying the provided value with the window size. The whole chunk is filtered followed by the extraction of sliding windows which are subsequently sent to the decoding pipeline for classification. The decoding results are finally transmitted to the UI process in batches containing the results for the whole chunk. For this study, the *chunk_prop* parameter was set to 2 resulting in a chunk size of 4 s.

The decoding pipeline uses a finite impulse response band-pass filter from 8–30 Hz with a windowed time-domain (firwin) design as suggested by [27]. The filter is a one-pass, zero-phase, non-causal bandpass filter. It uses a Hamming window with 0.0194 passband ripple and 53 dB stopband

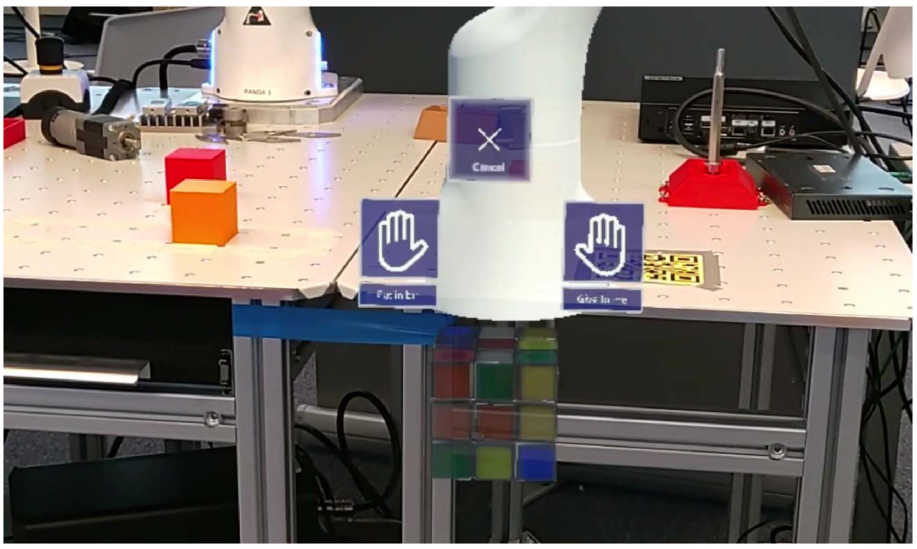

Figure 4. An example of the augmented reality interface when the Rubik's cube is selected. The icon on the left signals to the user that imagining closing their left hand will command the robot to put the cube away. The icon on the right shows that imagining closing their right hand will prompt the robot to give the cube to the user. The cross icon at the top serves as a button that can be selected using eye tracking to return to the object selection stage of the control system.

attenuation, a lower passband edge of 8, and a lower transition bandwidth of 2.00 Hz for a −6 dB cutoff frequency of 7.00 Hz. The upper passband edge was set to 30.00 Hz and the upper transition bandwidth was set to 7.50 Hz for a −6 dB cutoff frequency of 33.75 Hz. The filter length was automatically set to 413 samples (1.652 s) according to the default settings provided by the MNE filter functions.

The MI decoding pipeline consists of an SKLearn Pipeline object containing the feature extraction and classification stages. The first step of this pipeline consists of a common spatial pattern feature extraction model provided by MNE. The number of components used by common spatial patterns was set to 6. The classifier used for predicting the imagined movement was linear discriminant analysis. This model was implemented using the related class provided by SKLearn.

2.3.2. AR UI application

Our AR eye-tracking app uses eye-tracking technology to help with studies in human-computer interaction, with the main goal of making it easier for virtual and real-world robots to connect with each other. The app uses an eye-gaze pointer to detect virtual items that it collides with and BCI to figure out what the user wants to do to control a virtual robot. After that, the appropriate movements are sent to a real robot if it is connected. The app was made with Unity3D [28] and has a simple, easy-to-use UI so that people can connect with the robot without any problems. Mixed reality components of the UI, such as eye tracking buttons, are provided by the mixed reality toolkit for Unity [29].

Initially, the UI shows the virtual robot and objects related to the current task. When the eye-tracking gaze pointer detects a collision with an interactive object, the menu is shown and the start decoding message is sent. Figure 4 shows an example of the decoding menu. Subsequently, a flag is set in the Unity software that indicates that there should be a check for new decoding results at each frame update. When decoded movements come in, a counter is increased for each detected movement. Each counter update is followed by a check to verify if the threshold was reached for the given action. If this is the case, the confirmation menu is shown as a holographic modal.

When the user accepts the action, the confirmation modal is hidden, the virtual robot performs the chosen action, and a message is sent to the real robot over ROS TCP if it is connected. If the real robot is listening, it executes the same action as the virtual robot. If the user refuses the decoded action, the confirmation modal remains visible and decoding is resumed when the user looks at the object again. Only the counter for the proposed action is reset to 0 in this case. The counter for the other action remains at the same value which allows for faster convergence to the correct action when correct predictions were received along with the incorrect ones.

2.4. Validation study

The prototype control system was validated with a user study where participants had to operate the robotic arm using our control system. The study was split into 3 distinct phases to allow for iterative improvement of the control system between phases,

ensuring a stable experience for the final phase. Participants had to complete several tasks, which are presented in section 2.4.1, in a virtual AR environment showing a virtual robot with virtual objects to enable the assessment of the system's performance. When the real robot is connected, it reproduces the actions of the virtual robot using cubes representing the virtual objects which are located at predetermined positions. The procedure followed the same template for each phase, with the difference being in the number of sessions and the tasks that had to be completed. Performance was assessed using a set of performance metrics that are representative of the real-world suitability of the control system.

This study was approved by the Medical Ethics Committee of UZ Brussel and VUB (BUN1432023000232) and adheres to the principles of the Declaration of Helsinki for medical research involving human participants [30]. Participants received a detailed explanation of the procedure when they registered for the study and provided their written informed consent at the beginning of the first session. Instructions were orally recited before starting each task. In total, 20 participants (4 female, 16 male; aged 23–30) participated in the study. All participants were individuals with no known cognitive impairments.

2.4.1. Tasks

Before being able to use the control system, a decoding pipeline should be trained on data from the participant. Hence, each session started with a calibration task where EEG data was recorded while the participant imagines movements. The employed data acquisition procedure is an AR implementation of the procedure that was used in our previous research [31]. For this task, the participant is seated while equipped with the EEG cap and the AR HMD.

A white cross appears in front of them to indicate that they should get ready and to provide a fixation point, thus avoiding head and eye movement. After 3 s, a textual cue appears above the cross with either the word *Left* or *Right* indicating that the next imaginary movement should be closing the left or right hand respectively. The cue is shown for 1.5 s after which the text disappears and the cross turns green, indicating that the participant should start imagining the requested movement. The cross subsequently turns red after 2.5 s to tell the participant that they can stop imagining. This is shown for 1 s, followed by a break of randomized length between 3 and 4 s. Finally, the fixation cross reappears indicating that the next trial will start. This procedure is repeated until the necessary training data are acquired. For this study, each calibration run consisted of 20 samples of each movement, totaling 40 trials per run. The order of movements is randomized at the start of the run.

Following calibration, the acquired data are used to train an MI decoding pipeline that is used to operate the robot arm in evaluation tasks. Training the decoding pipeline always uses all calibration data that were acquired during the session and never uses data from other sessions. The first evaluation task consists of sorting a cube using the robot arm. For this task, the participant is seated while equipped with the EEG and AR hardware. A virtual robot arm is positioned on a virtual table in front of them with two differently colored baskets on the sides. A colored cube appears at a predetermined location in front of the robot, indicating which basket should be selected. The user then chooses the correct side by following the instructions indicated by the UI. Each run consists of several repetitions of the sorting task until the end conditions are met. Possible end conditions are presented in section 2.4.2.

The second task is a pick-and-place scenario where the participant is once again seated while equipped with the EEG and AR hardware. In addition to the the robot arm in front of them, there are 4 objects on the table; an orange, a Rubik's cube, a TV remote, and a smartphone. When the participant selects an object and an action, the robot picks up the object and places it in a predetermined location associated with this action. The participant receives a previously determined sequence of object-action pairs that they need to perform to complete the run. The order of objects and which action should be selected is randomized for each run.

2.4.2. Procedure

Each phase consisted of multiple sessions where the first sessions were intended to help participants familiarize themselves with the concept of MI BCI and the use of the control system. Participants were encouraged to ask questions during the experiment and were frequently asked about their experience to ensure optimal training. Subsequent sessions were intended to train the user further in the use of the control system with the last session culminating in a performance assessment using one of the evaluation tasks.

Phase 1 consisted of a validation study using the OpenBCI EEG headset. This phase was intended to validate the overall feasibility of our control system and investigate the possibility of using a consumer-grade EEG device. The methodological details and results from this phase are discussed in [18].

For phase 2, 5 participants (1 female, 4 male; aged 23–29) visited our lab for 3 sessions each. The performance assessment employed the same sorting task as in Phase 1, with the main difference being that the participant had to complete 10 repetitions of sorting to consider a run successful. For the first session, participants performed two consecutive calibration runs with a 5 min break in between. Subsequently, they were asked to perform at least one sorting run with the possibility of a second run if there was time

and they were willing. The second session consisted of an alternation of calibration and sorting runs with a maximum of 2 calibration runs. Participants were encouraged to perform a third sorting run if possible. There was no time limit for the first two sessions. For the final session, only one calibration run was performed. The participant subsequently had to perform three consecutive sorting runs that were timed. The time limit was set to 15 min and a run was considered failed if the participant was unable to complete 10 sorting tasks within this limit.

The final phase expanded the procedure further by introducing the pick-and-place task and using a real robot in the final session of each participant. 12 participants (3 female, 9 male; aged 22–30) visited our lab for 3 sessions each in this phase. The first session was identical to the first session in Phase 2. The second session started with a calibration run followed by a sorting run. After a second calibration run, the pick-and-place task was introduced. Participants were asked to choose the order of objects and the actions themselves. They were requested to announce their choice before initiating the object selection stage of the control system. Afterward, they were encouraged to perform an optional run where the experimenter provided the sequence of object-action pairs to prepare them for the final session. Finally, the last session introduced the real robot and took place at the AI Experience Center at VUB, located in Brussels, Belgium, where the robot arm is located. This resulted in more noisy, although more realistic, conditions. The sessions started with a single calibration run followed by 3 consecutive pick-and-place runs where the real robot replicates the virtual robot's actions. Participants had to complete the provided sequence within 15 min to consider a run successful.

2.4.3. Performance metrics

To assess the control system's objective performance, several metrics were used. To compute these metrics, markers were placed in the EEG data at each event of a run. When a user selects an object to interact with, a start decoding marker is placed in the data. Subsequently, when the decoding is finished and the user either accepts or declines the proposed action, a marker stating the user's choice is added to the data. If the user accepts the action, the marker contains the predicted, and hence ground-truth, class. If the action is rejected, the marker will contain both the predicted and true classes, which is considered as the alternative class since we only employ two classes for this system.

Using the prediction markers, the online decoding accuracy can be extracted. The chosen accuracy measures were the balanced decoding accuracy, Cohen's Kappa, and F1 score. The formula used for computing balanced accuracy is

$$\text{balanced} - \text{accuracy} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP}+\text{FN}} + \frac{\text{TN}}{\text{TN}+\text{FP}}\right)$$

where TP is the number of true positives, FN is the number of false negatives, TN is the number of true negatives and FP is the number of false positives. When referring to accuracy, balanced accuracy is intended. Cohen's Kappa measures the agreement between two raters where one is a random classifier, i.e. 0.50 for a binary classifier and our prediction pipeline [32]. The metric can be computed as

$$\kappa = \times \frac{2 \times (\text{TP} \times \text{TN} - \text{FN} \times \text{FP})}{(\text{TP}+\text{FP}) \times (\text{FP}+\text{TN}) + (\text{TP}+\text{FN}) \times (\text{FN}+\text{TN})}$$

where TP is the number of true positives, FN is the number of false negatives, TN is the number of true negatives and FP is the number of false positives. The F1 score is the harmonic mean of the precision and recall [33] which is computed as

$$F1 = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}$$

where TP is the number of true positives, FN is the number of false negatives, and FP is the number of false positives.

Since we also know exactly when decoding was initiated and when a prediction was made, we can also extract the decision time. Note that this differs from the decoding time as several predictions are used to make a decision. Finally, we also consider the success rate, calculated as

$$\text{success} - \text{rate} = \frac{n_c}{n_r}$$

where n_c is the number of completed runs and n_r is the total number of runs performed. Each participant completed 3 runs of the evaluation task.

3. Results

In Phase 1, all 3 participants were able to complete at least a single sorting task. Hence, the feasibility of our control approach was shown. This also shows the feasibility of using the OpenBCI cap for EEG measurement with our control system.

Table 1 provides an overview of the online decoding performance during the sorting task for each participant in Phase 2.

We can observe that the online decoding performance is low, even below random chance for all but Participant 7. The Kappa value even goes to 0 for Participant 5 and close to 0 for Participant 6, indicating that the system repeatedly made wrong predictions, resulting in many erroneous decisions. This is also clearly reflected by the low success rate for those participants. Nevertheless, even with this low decoding performance, participants 4, 7, and 8 had a perfect success rate of 1, showing that they were always able to complete the task within the given time limit.

Table 1. Overview of Phase 2 online decoding performance during the sorting evaluation task and each participant's success rate in completing the task. The employed metrics are the balanced accuracy, F1 score and Cohen's Kappa (κ). Only 1 participant could achieve an accuracy above random chance, which is reflected by the other performance metrics. Nevertheless, a mean success rate of 0.73 could be achieved.

Participant	Accuracy	$F1$	κ	Success rate
4	0.50	0.29	−0.04	1.00
5	0.20	0.15	0.00	0.33
6	0.36	0.16	0.00	0.33
7	0.65	0.53	0.25	1.00
8	0.50	0.46	0.02	1.00
Mean	0.44	0.32	0.05	0.73
Standard deviation	0.15	0.16	0.11	0.33

Table 2. Overview of Phase 3 online decoding performance during the pick-and-place evaluation task and each participant's success rate in completing the task. The employed metrics are the balanced accuracy, F1 score and Cohen's Kappa (κ). Online decoding performance is also low though the mean decoding accuracy was slightly higher than for Phase 2 at 0.52. The success rate was also higher at 0.83 with the majority of participants achieving a perfect success rate.

Participant	Accuracy	$F1$	κ	Success rate
9	0.25	0.17	−0.08	NA
10	0.45	0.19	−0.04	1.00
11	0.53	0.10	0.01	0.33
12	0.64	0.42	0.31	0.67
13	0.36	0.09	−0.00	0.33
14	0.53	0.33	−0.03	1.00
15	0.51	0.36	−0.00	1.00
16	0.60	0.32	0.07	1.00
17	0.69	0.51	0.30	NA
18	0.58	0.26	0.06	1.00
19	0.54	0.31	0.03	1.00
20	0.56	0.42	0.09	1.00
Mean	0.52	0.29	0.06	0.83
Standard deviation	0.12	0.13	0.12	0.27

Table 2 shows the performance overview for the participants of Phase 2. Note that the success rate for Participants 9 and 17 is unavailable due to the timing data being lost after a technical problem, prohibiting the computation of the success rate.

We can observe that the decoding performance is slightly improved over Phase 1 from the higher mean accuracy and Kappa values with lower variance for the accuracy. However, the mean F1 is lower in Phase 3 even though the accuracy is higher, indicating that there were more events where one class could not reliably be predicted in this phase. This occurs when the decoding pipeline often predicts the same class regardless of the true value. This indicates that while participants got stuck for several consecutive predictions less frequently, there were often random mistakes. The success rate was similar for both phases.

To assess the BCI control system's effectiveness, we investigated if the success rate is reliable for different time limits, we computed the success rate for various cutoff times. The resulting mean success rates across all participants are shown in figure 5.

Figure 5 shows that given sufficient time, the majority, 73%, and 83% for Phase 2 and Phase 3 respectively, of users could complete the given task successfully. The success rate peaks at a cutoff time of 13 min for Phase 2. In Phase 3's pick-and-place task, most participants succeeded after 6 min, but a cutoff time of over 10 min was required to ensure that the majority of participants were able to complete the sorting task in Phase 2. This is expected due to the sorting task requiring more actions (10) compared to the pick-and-place task (4) though the actions in the pick-and-place task were slightly more complex as it requires selecting an object that is not known beforehand. The standard deviation was large for both phases, highlighting the large variability in BCI aptitude across participants.

Beyond effectiveness, a system's efficiency is also an important aspect. Thus, we investigated the completion times that were achieved during the user study. Figure 6(a) shows the mean completion time over all participants for both tasks and figure 6(b) shows each participant's best completion time.

We can observe that the pick-and-place completion times are lower than the sorting completion times, which is expected as there are fewer actions to perform in the pick-and-place task. We also notice that most participant's best completion time is below the average in both cases.

4. Discussion

The aim of this study was to demonstrate the validity of our novel control system which integrates BCI with eye tracking in an AR UI. To this end, we conducted a user study that balances the fast iteration needs of software development with the thorough user evaluation required in human-computer interaction research. Participants completed tasks simulating daily activities using the BCI control system to operate a robotic arm, allowing us to assess the system's usability, defined by its effectiveness, efficiency, and user experience [15]. Usability and user-centered design are increasingly crucial in BCI research, bridging the gap between laboratory experiments and real-world applications [16, 17]. This study focused on objective measures of the effectiveness and efficiency of the developed BCI control system.

Our main findings indicate that while the current system can be used to control a robotic arm, several improvements are necessary before it can be deployed for real-world applications. The first phase of the user study demonstrated the feasibility of using a 16-channel OpenBCI cap with passive electrodes to

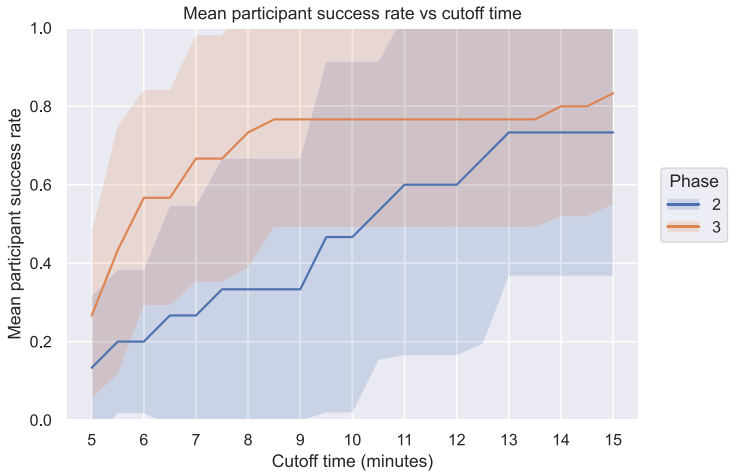

Figure 5. Grand mean of participant success rates plotted against different cutoff times for run completion. The color bands surrounding the lines show the standard deviation over all participants. Phase 2 shows sorting task success rates while Phase 3 shows pick-and-place task success rates. We observe that at least 11 and 6 min are necessary to achieve a success rate above 0.5 in Phase 2 and Phase 3 respectively.

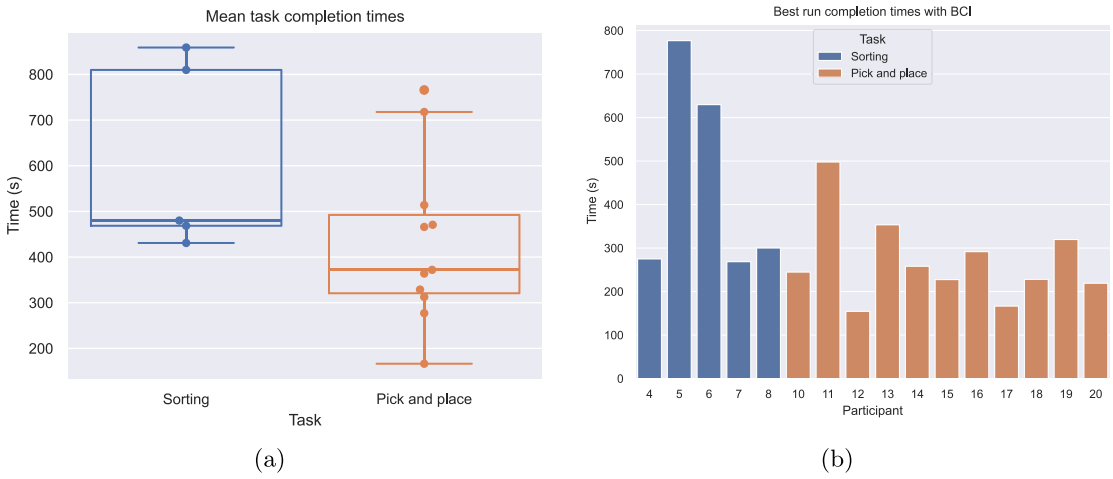

Figure 6. The (a) grand mean completion time over all participants with individual participant means represented by dots and (b) individual participant's best completion time. Completion times are lower for pick-and-place, which is expected due to having fewer tasks in a run. Most participants achieved a below-average completion time on their best run.

operate a robotic arm. Previous research has shown that using a reduced number of sensors to decode MI is feasible if they are appropriately located according to the 10–20 system [31]. These results, along with research into consumer BCI [34], suggest that an effective BCI control system could be devised using consumer-grade hardware given an appropriate control system design.

4.1. Control system performance

In subsequent phases, we assessed the effectiveness and efficiency of our BCI control system. Phase 2 revealed a discrepancy between expected offline and online decoding performance. Our decoding pipeline typically achieves an accuracy above 0.7 with similar data [31], which was confirmed in preliminary analyses using calibration data from this study. However, in real-time settings, accuracy dropped to below random chance with a mean MI decoding accuracy of 0.44 for Phase 2 participants. Phase 3 results were similar, with a mean accuracy of 0.52. This low accuracy can be attributed to the use of basic MI decoding methods and the lack of pipeline customization for individual participants

Despite this, participants achieved success rates of 0.73 and 0.83 in Phases 2 and 3, respectively. These success rates were obtained thanks to the shared control approach that includes eye tracking, thus limiting the reliance on accurate MI decoding However, these success rates were only achieved when participants were given ample time to complete tasks. Reducing the time limit significantly lowered the success rate, indicating that effectiveness may be achieved at the expense of efficiency. This trade-off could negatively impact user experience, as users might become frustrated if tasks take too long to complete.

Comparing our results with previous literature is challenging due to the lack of a standardized BCI prototype evaluation framework. Nevertheless, despite our lower decoding accuracies, our success rates align with previous studies on BCI for robotic arm control [4, 35, 36]. It is noteworthy that these studies use more advanced decoding methods and customize the decoding pipeline hyperparameters for each participant, while we used the same basic decoding pipeline for all participants. This suggests significant potential for improving our BCI control system by enhancing decoding accuracy.

4.2. Strengths and limitations

Several challenges remain for the real-world deployment of BCI systems [3]. Our control system's novel design aims to mitigate the inherent difficulties in decoding MI from EEG data using a shared control approach. Shared control employs a dynamic world model updated through various sensors monitoring the environment [14]. Although not the first attempt at robotic arm operation using a shared control BCI system [13, 37], our design is distinguished by its portability and multimodal interaction through eye tracking. Another distinctive feature of our BCI control system is the limited calibration time. In the final evaluation session, we utilized only 20 training samples per MI class, leading to a recording time of 7 min and 35 s. Due to the simplicity of the decoding methods used in our BCI pipeline, model training takes less than 5 s, resulting in a total calibration time of under 8 min. Fast setup and quick familiarization are crucial for a user-friendly control system [38–40].

Our BCI control system's strengths include enabling complex interactions with the environment through a simple UI. The multimodal combination of eye tracking and BCI separates object selection from robot action selection, resulting in a loose coupling of software components, which is desirable [41]. Moreover, the need to focus on an object to trigger BCI decoding minimizes the users's head and eye movement, thus limiting EEG artifacts. The system's flexibility allows for easy improvement or replacement of specific software or hardware components while keeping others fixed. Furthermore, our thorough evaluation procedure provides results indicative of real-world performance.

However, limitations include the low decoding performance of the real-time pipeline, leading to low system efficiency. Additionally, user training was limited in both duration and methods, preventing an assessment of optimal performance and potential improvements for weak performers. The low number of sessions also meant that participants had limited opportunities to achieve better results if they were tired during their final session. The effect of user training was not directly considered in the study. The current system simulates spatial awareness and object detection, so the computational requirements of these methods are also unknown.

4.3. Future perspectives

Future improvements could focus on optimizing the decoding pipeline to increase decision accuracy and efficiency, potentially through more advanced methods such as deep learning. However, these methods may require more training data and introduce longer decoding times. Promising techniques to address these issues include transfer learning [42], continual learning [43], and novel data augmentation methods like diffusion models [44]. Automated user customization tools, such as AutoML [45], could also enhance the decoding pipeline through extensive hyperparameter optimization for each user.

Utilizing advanced methods could also support additional MI classes, enabling a broader range of actions. Furthermore, reliable decoding could eliminate the accept/reject dialog to enhance efficiency, but this would require a mechanism to interrupt erroneous actions. Monitoring error-related potentials after the selection of an action could serve this purpose [46]. Improving calibration and user training through gamification could enhance user performance. Observations during the study suggest that users perform best when intuitively thinking of required movements in familiar contexts. Gamification could provide an immersive and motivating context for training users on the necessary movements [47, 48]. We believe that with enhanced MI decoding, the system could achieve significantly higher success rates and reduced completion times. If reliable decoding can be ensured, the confirmation stage in our control strategy could be eliminated, further boosting the system's efficiency.

Regarding the evaluation of the control system, assessing user experience is also necessary for a comprehensive evaluation of our control system. User experience outcomes can also assist in prioritizing work. Future work will thus include a user experience questionnaire [49] and comparisons with non-BCI control approaches to investigate user preferences. Alternatively, comparing the performance of our control system with MI-based control strategies that do not employ eye tracking should provide further insights into the added value of using eye tracking in combination with BCI.

Currently, our system uses off-the-shelf components, demonstrating the adequacy of existing technology. In the long-term, integrating all aspects of our BCI control system into a single all-in-one device, such as an AR headset with EEG sensors and embedded computing hardware, could provide a self-contained, comfortable, and privacy-guaranteeing solution. If commercialized at an affordable price, consumer BCI could become a reality.

5. Conclusion

The results obtained in this user study showed the viability of our BCI control strategy. The effectiveness of the software prototype that we developed was confirmed with most participants achieving a perfect success rate. However, the efficiency of the system still needs to be improved before real-world deployment becomes possible. Moreover, we also demonstrated the viability of using consumer-grade EEG devices with our control system. Thus, BCI is on the cusp of becoming a widely available interaction modality that could replace classical interaction modalities such as the mouse and keyboard. Though current applications are limited to niche scenarios where other modalities are unavailable, the future holds the possibility of BCI becoming ubiquitous in everyday interaction with electronic devices.

Data availability statement

The data cannot be made publicly available upon publication because they contain sensitive personal information. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgment

This research was made possible thanks to the EUTOIA alliance of universities supporting the PhD of AD and the Strategic Research Program 'Exercise and the Brain in Health and Disease: The Added Value of Human-Centered Robotics'. Bart Roelands is a Collen-Francqui research professor. We thank all the participants of the user study for their participation and for their motivation which ensured a thorough assessment of the capabilities of our control system. We also gratefully acknowledge the undergrad students and interns who assisted with the evaluation study.

Ethical statement

This study was approved by the Medical Ethics Committee of UZ Brussel and VUB (BUN1432023000232) and adhered to the principles of the Declaration of Helsinki for medical research involving human participants

Funding statement

This research received no external funding.

ORCID iDs

Arnau Dillen https://orcid.org/0000-0001-5840-8920
Mohsen Omidi https://orcid.org/0000-0002-0691-4183
Fakhreddine Ghaffari https://orcid.org/0000-0002-0928-7963
Bram Vanderborght https://orcid.org/0000-0003-4881-9341
Bart Roelands https://orcid.org/0000-0002-2808-044X
Olivier Romain https://orcid.org/0000-0002-2172-1865
Ann Nowé https://orcid.org/0000-0001-6346-4564
Kevin De Pauw https://orcid.org/0000-0002-6901-7199

References

[1] Willett F R, Avansino D T, Hochberg L R, Henderson J M and Shenoy K V 2021 High-performance brain-to-text communication via handwriting *Nature* **593** 249–54
[2] Liu Y, Habibnezhad M and Jebelli H 2021 Brain-computer interface for hands-free teleoperation of construction robots *Autom. Constr.* **123** 103523
[3] Värbu K, Muhammad N and Muhammad Y 2022 Past, present and future of EEG-based BCI applications *Sensors* **22** 3331
[4] Kuhner D *et al* 2019 A service assistant combining autonomous robotics, flexible goal formulation and deep-learning-based brain–computer interfacing *Robot. Auton. Syst.* **116** 98–113
[5] Hosseini M-P, Hosseini A and Ahi K 2021 A review on machine learning for EEG signal processing in bioengineering *IEEE Rev. Biomed. Eng.* **14** 204–18
[6] Ramadan R A and Vasilakos A V 2017 Brain computer interface: control signals review *Neurocomputing* **223** 26–44
[7] Saha S and Baumert M 2020 Intra- and inter-subject variability in EEG-based sensorimotor brain computer interface: a review *Front. Comput. Neurosci.* **13** 87
[8] Rashid M, Sulaiman N, Abdul Majeed A P P, Muazu Musa R, Ahmad Fakhri A Nasir B S B and Khatun S 2020 Current status, challenges and possible solutions of EEG-based brain-computer interface: a comprehensive review *Front. Neurorobot.* **14** 25
[9] Zhang Y, Xie S Q, Wang H and Zhang Z 2021 Data analytics in steady-state visual evoked potential-based brain–computer interface: a review *IEEE Sens. J.* **21** 1124–38
[10] Li M, He D, Li C and Qi S 2021 Brain–computer interface speller based on steady-state visual evoked potential: a review focusing on the stimulus paradigm and performance *Brain Sci.* **11** 450
[11] Naeem Mannan M M, Ahmad Kamran M, Kang S, Soo Choi H and Yung Jeong M 2020 A hybrid speller design using eye tracking and SSVEP brain–computer interface *Sensors* **20** 891
[12] Decety J 1996 The neurophysiological basis of motor imagery *Behav. Brain Res.* **77** 45–52
[13] Yang X, Ding C, Shu X, Gui K, Bezsudnova Y, Sheng X and Zhang D 2019 Shared control of a robotic arm using non-invasive brain–computer interface and computer vision guidance *Robot. Auton. Syst.* **115** 121–9
[14] Losey D P, McDonald C G, Battaglia E and O'Malley M K 2018 A review of intent detection, arbitration and communication aspects of shared control for physical human–robot interaction *Appl. Mech. Rev.* **70** 010804
[15] Nielsen J 1994 *Usability Engineering* (Morgan Kaufmann)
[16] Kübler A *et al* 2014 The user-centered design as novel perspective for evaluating the usability of BCI-controlled applications *PLoS One* **9** e112392
[17] Nohemi Ortega-Gijon Y and Mezura-Godoy C 2019 Usability evaluation of brain computer interfaces: analysis of methods and tools *2019 IEEE Int. Conf. on Engineering*

Veracruz (ICEV) (*Boca del Rio, Mexico, 14–17 October 2019*) vol I pp 1–8
[18] Dillen A, Omidi M, Ghaffari F, Romain O, Vanderborght B, Roelands B, Nowé A and De Pauw K 2024 User evaluation of a shared robot control system combining BCI and eye tracking in a portable augmented reality user interface *Sensors* **24** 5253
[19] Evans G, Miller J, Iglesias Pena M, MacAllister A and Winer E 2017 Evaluating the Microsoft HoloLens through an augmented reality assembly application *Degraded Environments: Sensing, Processing, and Display 2017* (*Anaheim, California, USA, 11–12 April 2017*) vol 10197 pp 282–97
[20] Haddadin S, Parusel S, Johannsmeier L, Golz S, Gabl S, Walch F, Sabaghian M, Jähne C, Hausperger L and Haddadin S 2022 The franka emika robot: a reference platform for robotics research and education *IEEE Robot. Autom. Mag.* **29** 46–64
[21] Toris R, Shue C and Chernova S 2014 Message authentication codes for secure remote non-native client connections to ros enabled robots *2014 IEEE Int. Conf. on Technologies for Practical Robot Applications (TePRA)* (*Woburn, MA, USA, 14–15 April 2014*) (IEEE) pp 1–6
[22] Allspaw J, LeMasurier G and Yanco H 2023 Comparing performance between different implementations of ros for unity
[23] Chitta S 2016 Moveit!: an introduction *Robot Operating System (ROS) The Complete Reference* vol 1 (Springer International Publishing) pp 3–27
[24] Gramfort A *et al* 2013 MEG and EEG data analysis with MNE-Python *Front. Neurosci.* **7** 267
[25] LSL contributors 2023 Sccn/labstreaminglayer GitHub Swartz Center for Computational Neuroscience (Retrieved 13 December 2023)
[26] Pedregosa F *et al* 2011 Scikit-learn: machine learning in Python *J. Mach. Learn. Res.* **12** 2825–30
[27] Widmann A, Schröger E and Maess B 2015 Digital filter design for electrophysiological data – a practical approach *J. Neurosci. Methods* **250** 34–46
[28] Unity Technologies 2024 *Unity 2021.3.28f Game Development Platform* (Unity Technologies)
[29] Microsoft 2022 *MRTK2-Unity Developer Documentation - MRTK 2* (Microsoft Developer Portal)
[30] World Medical Association 2013 World medical association declaration of helsinki: ethical principles for medical research involving human subjects *JAMA* **310** 2191–4
[31] Dillen A, Ghaffari F, Romain O, Vanderborght B, Marusic U, Grosprêtre S, Nowé A, Meeusen R and De Pauw K 2023 Optimal sensor set for decoding motor imagery from EEG *Appl. Sci.* **13** 4438
[32] McHugh M L 2012 Interrater reliability: the kappa statistic *Biochem. Med.* **22** 276–82
[33] Powers D 2011 Evaluation: from precision, recall and F-factor to ROC, informedness, markedness & correlation *J. Mach. Learn. Technol.* **2** 37–63
[34] Peterson V, Galván C, Hernández H and Spies R 2020 A feasibility study of a complete low-cost consumer-grade brain-computer interface system *Heliyon* **6** e03425
[35] Xu B, Li W, Liu D, Zhang K, Miao M, Xu G and Song A 2022 Continuous hybrid BCI control for robotic arm using noninvasive electroencephalogram, computer vision and eye tracking *Mathematics* **10** 618
[36] Sanna A, Manuri F, Fiorenza J and De Pace F 2022 BARI: an affordable brain-augmented reality interface to support human–robot collaboration in assembly tasks *Information* **13** 460
[37] Tonin L, Leeb R, Tavella M, Perdikis S and Millán J del R 2010 The role of shared-control in BCI-based telepresence *2010 IEEE Int. Conf. on Systems, Man and Cybernetics* (*Istanbul, Turkey, 10–13 October 2010*) (IEEE) pp 1462–6
[38] Castaño Arranz M, Birk W and Kadhim A 2017 On guided and automatic control configuration selection *2017 22nd IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)* (*Limassol, Cyprus, 12–15 September 2017*) (IEEE) pp 1–6
[39] Bromley C 2020 Complete control systems, not complex *Ind. Veh. Technol. Int.* **28** 88
[40] Sadeghian Borojeni S, Flemisch F, Baltzer M and Boll S 2018 Automotive UI for controllability and safe transitions of control adjunct *AutomotiveUI '18: 10th Int. Conf. on Automotive User Interfaces and Interactive Vehicular Applications* (Toronto, ON, Canada, *23–25 September 2018*) pp 23–29
[41] Pei Breivold H and Larsson M 2007 Component-based and service-oriented software engineering: key concepts and principles *33rd EUROMICRO Conf. on Software Engineering and Advanced Applications (EUROMICRO 2007)* (*Lubeck, Germany, 28–31 August 2007*) pp 13–20
[42] Wan Z, Yang R, Huang M, Zeng N and Liu X 2021 A review on transfer learning in EEG signal analysis *Neurocomputing* **421** 1–14
[43] Lee P-L, Chen S-H, Chang T-C, Lee W-K, Hsu H-T and Chang H-H 2023 Continual learning of a transformer-based deep learning classifier using an initial model from action observation EEG data to online motor imagery classification *Bioengineering* **10** 186
[44] Klein G, Guetschel P, Silvestri G and Tangermann M 2024 Synthesizing EEG signals from event-related potential paradigms with conditional diffusion models (arXiv:2403.18486)
[45] He X, Zhao K and Chu X 2021 AutoML: a survey of the state-of-the-art *Knowl.-Based Syst.* **212** 106622
[46] Wang X, Chen H-T, Wang Y-K and Lin C-T 2023 Implicit robot control using error-related potential-based brain–computer interface *IEEE Trans. Cogn. Dev. Syst.* **15** 198–209
[47] Roc A, Pillette L, Mladenovic J, Benaroch C, N'Kaoua B, Jeunet C and Lotte F 2021 A review of user training methods in brain computer interfaces based on mental tasks *J. Neural Eng.* **18** 011002
[48] Vourvopoulos A, Andrés Blanco-Mora D, Aldridge A, Jorge C, Figueiredo P and Badia S B i 2022 Enhancing motor-imagery brain-computer interface training with embodied virtual reality: a pilot study with older adults *2022 IEEE Int. Conf. on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)* (*Rome, Italy, 26–28 October 2022*) pp 157–62
[49] Laugwitz B, Held T and Schrepp M 2008 Construction and evaluation of a user experience questionnaire *HCI and Usability for Education and Work* (*Lecture Notes in Computer Science*) ed A Holzinger (Springer) pp 63–76