A Closed-Loop Brain-Computer Interface with Augmented Reality Feedback for Industrial Human-Robot Collaboration

Zhenrui Ji[1,2], Quan Liu[1,2], Wenjun Xu[1,2]*, Bitao Yao[2,3], Jiayi Liu[1,2], Zude Zhou[1,2]

[1] School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China

[2] Hubei Key Laboratory of Broadband Wireless Communication and Sensor Networks, Wuhan University of Technology, Wuhan 430070, China

[3] School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China

[*] Corresponding Author: Wenjun Xu with phone: +86-13707181116; fax: +86-27-87651800; email: xuwenjun@whut.edu.cn

Abstract

Industrial human-robot collaboration (HRC) aims to combine the human intelligence and robotic capability to achieve the higher productiveness. In industrial HRC, the communication between human and robot is essential to enhance the understanding the intent of each other to make a more fluent collaboration. Brain-computer interface (BCI) is a technology that could record the user's brain activity that can be translated into interaction messages (e.g., control commands) to the outside world, which is able to build a direct and efficient communication channel between human and robot. However, due to lacking of information feedback mechanism, it is challenging for BCI to control robots with high degree of freedom with the limited number of classifiable mental state. To address this problem, this paper proposes a close-loop BCI with contextual visual feedback by an augmented reality (AR) headset. In such BCI, the electroencephalogram (EEG) patterns from the multiple voluntary eye blinks are considered as the input and the its online detection algorithm is proposed whose average accuracy can reach 94.31%. Moreover, an AR-enable information feedback interface is designed which enable to achieve an interactive robotic path planning. A case study of an industrial HRC assembly task is also develop which shows that the proposed closed-up BCI could shorten the time of user input in human-robot interaction.

Keywords: Industrial human-robot collaboration, Brain-computer interface, Augmented reality, Interactive robotic path planning

1. Introduction

In recent years, industrial human-robot collaboration (HRC), which is a promising production paradigm that enable human and robot work side-by-side to jointly complete same production goals, has enjoyed the more and more interest from industry and academia. In order to ensure the fluency and productivity of industrial HRC, an efficient and intuitive communication channel is essential for human and robot to understand each other's intent. In the earlier literature, various modal human-robot interfaces have been introduced into this context, such as gestures recognition [1], haptic interaction [2], speech processing [3] and et al, to fulfill this end. The above interaction methods are more or less based on the human limb movements or verbal actions which might be difficult to ensure its robustness in a relatively uncertain environment (e.g., the visually obstructed or noisy

environment).

With the last advances in neuroscience and brain-imaging technologies, some researchers have proposed the potential application of brain-computer interface (BCI) as an auxiliary channel for existing interaction solutions to support the industrial HRC [4,5]. Brain-computer interface (BCI) is a technology that monitors the activity of the human's central nervous system (CNS) to interpreting the users' intention into the readable information [6]. With BCI, users can explicitly manipulate their brain activity instead of limb movements to produce signals that control external machines, which is a more direct and natural way to convey their intention [7]. However, it is still challenging for BCI to control a robot with high degree of freedom because of the limited number of mental states that could be classified in the existing BCIs which is unable to generate the high-dimensional commands. To overcome this problem, a potential solution can be to offer sufficient information feedback for the BCI users who can then perform their limited types of input to accomplish the generation of complex commands.

In this paper, we introduce augmented reality feedback to achieve a close-loop BCI to control the robot in industrial HRC scenario. Firstly, from a practical point of view, we use the voluntary eyeblinks, which can be easily performed without intensive user-training and can generate relatively strong signals with stable patterns, as the input of the BCI user to communicate with robot. Secondly, an Optical See-Through (OST) AR headset is used to provide a contextual visual feedback according to the input of BCI user, which can achieve the complex robotic operation, such as the specify a trajectory of robot movement. To verify the effectiveness of the proposed method, we also deploy a BCI-AR interface in an industrial HRC task and compare its performance with [8], a hand gesture-based AR interface.

The rest of the paper is organized as follows. Section 2 reviews previous work in the area of BCI and its combination with AR for human-robot communication. Section 3 gives an overview of the closed-loop BCI system. Section 4 describes the proposed EEG-based voluntary blinks online detection approach, while the design of the AR-enabled feedback interface is presented in Section 4. In Section 5, the performance analysis of the proposed detection algorithm and a case study in an industrial HRC task is developed. Conclusions and future work are summarized in Section 7.

2. Related work

Historically, the application of BCI mainly focus on the assistance for the people with motion disabilities, e.g., wheel chair [9], home automation [10], word typing [11]. Focusing on one form of noninvasive BCI, electroencephalography (EEG), the brain activity is monitored as electrical signals measured on specific areas of the BCI user's scalp though the electrodes. In the past decades, many paradigms of EEG-based BCI have been developed, such as steady-state visual evoked potentials (SSVEP) [12], P300 [13], motor-imagery [14] and et al. These paradigms either require extra visual stimulus (SSVEP and P300) or require intensive user training (motor-imagery) which impose a challenge in the practical application for non-disabled users in some practical scenarios, like the industrial HRC.

In EEG-based BCI, eye blinks are generally seen as artifacts that should be removed before the

further EEG signal processing. In fact, the eye blinks are typically classified into three categories: one is a *spontaneous* (or *normal*) eye blink which occurs frequently, another is a *reflexive* eye blink which is evoked by an external stimulus (e.g., some hazard situations), and the other is a *voluntary* eye blink which is caused by intentional eye closing [15]. The voluntary eye blink, which can be seen as the intentional action of the BCI users and don't requirement external stimulus nor intensive user training, has been used as the trigger signal of BCI in some research [16-18]. In the view of the advantages of the voluntary eye blink's application in BCI, in this paper, it is used as the user input of the proposed closed-loop BCI interface.

In the EEG-based eye blink detection, the algorithms mainly categorized into (1) threshold-based approaches; (2) template matching-based approaches; and (3) learning-based approaches. As for the threshold-based approaches, [19] proposed a multi-window summation of derivatives approach whose performance was compared with the Root Mean Square Error (RMSE), Dynamic Time Warping (DTW) and other approaches. [20] adopted a threshold-based peak detection to switch on/off the home lighting system. [21] proposed a Power Spectrum Density (PSD) threshold-based method, where a moving window of PSD was compared to a threshold to detect eye blinks pattern. These threshold-based approaches are of highly sensitivity to the feature selection and the current threshold, which could diff highly across users and hardware. As for the template matching-based approaches, [22] proposed a Dynamic Positional Warping (DPW) based method, which is a variant of DTW and its accuracy was between than the DTW-based and the RMSE-based algorithms. The template matching-based approaches depends on the manual template inspection or the specific algorithm for the template generation. As for the supervised learning-based approaches, [23] proposed a Support Vector Machines (SVMs)-based eye-blink identification algorithm with a moving window. [24] proposed an RBF network-based eye-blink feature extraction method with a segmentation of a fixed window of 1 second. Such learning-based methods require user training and machine learning which also needs a lot of labelled training data. [25] proposed a fully automated unsupervised blinks detection algorithm *Blink* based on waveform feature detection and cluster-analysis, which do not need any user training or manual labelling requirements. However, this algorithm is focusing on the normal blink detection instead of the voluntary one. In this paper, inspired by *Blink*, we propose a multiple voluntary detection algorithm with a fully automated unsupervised manner.

In the combination of BCI and AR, the most of the existing study are based on the SSVEP or P300 paradigm, where, on the one hand, BCI offers a hand-free interaction paradigm for AR; on the other hand, the AR offers a channel to integrate and feedback the real-world information for BCI users [26]. In [27], focusing on Industry 4.0, the authors proposed a wearable monitoring system for inspection based on an AR glasses and a single-channel BCI. The system applied the SSEVP paradigm, whose visual stimuli was displayed though the AR glasses while displaying the inspection information. Some other robotic application also has been developed for assisting the disabled people. [28] proposed a BCI-based telepresence mobile robot with a monitor-based AR displayer,

where the monitor could tele-present the task-related information and the BCI could detect the P300 signal to determine the task selected by users. In [29], a BCI for high-level remote robotic control is proposed, which combined the motor-imagery paradigm and the monitor-based AR. The monitor displays the objects that robot could grasp for user, and the user can perform different movement imaginary to select the object to be grasped. These researches are mainly focusing on the high-level control which are unable to specify the detailed configuration of the robot operation, like the path planning. In [30], an AR-based robot path planning interface has been proposed, which is based on a monitor-based AR device and a hand-held tool for input the planning command. A similar monitor-based interface proposed in [31] uses hand gesture as input with the recognition by a camera. In [32], an AR HMD (HoloLens) is used to develop an interactive robotic pathing programming with the 3D visual trajectory feedback and hand gesture inputs. In this paper, inspired by the AR-based robot path planning in [30-32], we propose an AR-enable robot path planning with the input from BCI and without any extra hand-held interaction tool and hand gesture inputs.

3. System architecture

The developed system makes full of the advantage of brain-computer interface (BCI) and augmented reality head-mounted display (AR HMD). On the one hand, the BCI brings a direct input channel for human to non-verbally convey their control intention to robot without any limb movement. On the other hand, the equipped AR HMD serves as a visual aid tool to provides human an on-site information feedback (the EEG detection result, the robotic motion trajectory, the robotic motion preview, etc.) via the visual elements which is an efficient way to improve the human's understanding to the motion intent of collaborative robot [33].

As shown in Fig.1, the developed system consists of an EEG-based multiple voluntary blinks detector, an augmented reality feedback interface and a bi-direction information pipeline.

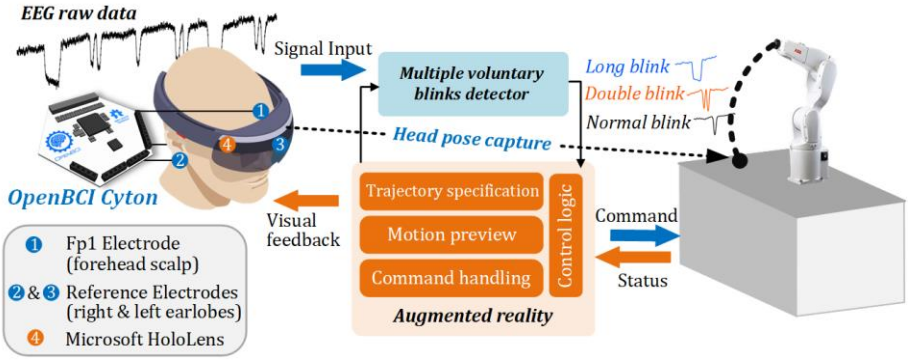

Fig.1 System overview.

In the EEG-based multiple voluntary blinks detector, the input EEG raw data is collected by OpenBCI Cyton [34] with three wet gold cup electrodes (one is on the Fp1 of international 10-20 electrode system (see Fig.2) as signal source, and two are on the earlobes as reference electrodes) and is analyzed by the blinks detection algorithm. The detector algorithm consists of pre-processing, features extraction and unsupervised clustering, which can online detect the user's voluntary eye blinks of two types (long blink and double blink) without any user training session.

The voluntary blinks are regarded as the input of user with control intention.

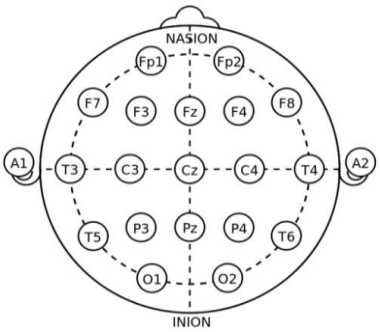

Fig.2 The international 10-20 EEG electrodes placement system [35].

The augmented reality feedback interface consists of a control logic module, a robotic trajectory specification module, a motion preview module and a command handling module. In the control logic module, the detected voluntary blinks result is as the input signal of a finite state machine which can control the transition of the interaction status (including initial status, interactive path planning, motion preview, and command handling). The trajectory specification module and the motion preview module are both powered by the augmented reality rendering and are integrate into an interactive robotic path planning; the former uses a head pose capture built in the AR headset and the editable visual trajectory of robot's end-effector as the spatial information of path planning from the user, and the voluntary eye blink can switch the interactive status to modify the path planning configuration; the latter is an AR simulation of the robot modified configuration which can give human a preview of the robot's ongoing motion. In the command handler module, the modified robot path planning is translated into the robotic executable command, which enable human to on-the-fly specify the robot's motion target without programming.

The bi-direction information pipeline includes the signal captured from human (EEG data and head pose data) and translated into command for robot, and the AR elements (visual trajectory, 3D simulation, and etc.) corresponding to the robot's status as the visual feedback for the human.

4. EEG-based voluntary eye blink online detection

Eye blinks can be classify as spontaneous (or normal), reflexive and voluntary, among which, generally, the normal blinks and the voluntary blinks are the relatively common observable behavioral phenomenon [15]. As the voluntary blinks are driven by the intentional eye closing which can reflect human's intention to some extent, we focus on using it as the human's input of the human-robot communication channel. In this paper, we choose two kinds specific of voluntary blinks: 1) *double blink* and 2) *long blink*. Double blink includes two eye blinks with a short interval; long blink is the eye blink with a long duration of eye closing. The rest of this section will introduce how to implement the voluntary blinks detection from the real-time EEG signal.

4.1 Eye blink patterns of EEG signal

When an eye blink occurs, the frontal EEG signal shows a typical trough pattern in voltage-

time domain. Fig. 3 shows such EEG pattern of normal blink, double blink and long blink at frontal electrode position (Fp1 according to the 10-20 electrode system) with two earlobe referenced electrodes. In the same-user case, these three kinds of eye blinks have similar amplitude with two stable points (which represents the starting moment of closing eyes and the ending moment of re-opening eyes) and one or two middle local minimum point (which represents the moment of full eye closure). Through the two stable points adjacent to the same local minimum point(s), the blink duration of different types can be computed; through two adjacent local minimum points, the blink interval can be obtained. According to the characteristics of EEG signal of eye blinks, we find that the amplitude, duration and interval can be used to identify the different kinds of eye blinks: (1) the amplitude can be used to detect the local minimum points and corresponding stable points; (2) a short interval between local minimum points represents a double blink; (3) in the non-double blink case, a long duration represents a long blink. To achieve the above detection, the threshold of amplitude, interval and duration should be determined to classify the different types of blinks.

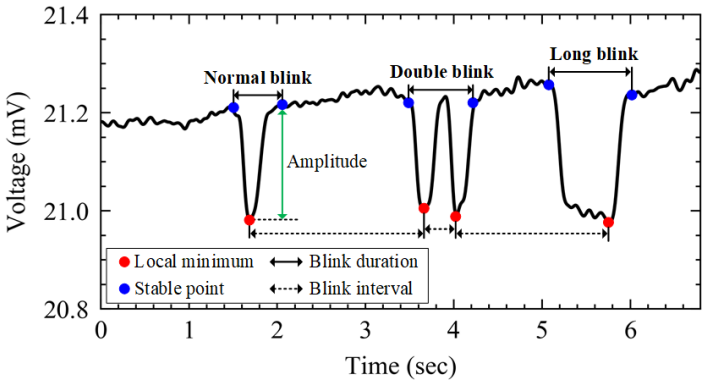

Fig.3 The typical EEG waveforms of normal blink, double blink and long blink.

The earlier literature [25] has found that the variability of amplitude occurs in the intra-subject cases (blinks of the same user in different time periods) and inter-subject cases(blinks from different users). Thus, it's challenging to pre-determine a fixed threshold of amplitude. As for the duration and interval of blinks, the earlier literature [15] has found that the spontaneous blinks has a characteristic duration and interval distribution. In order to determine reasonable thresholds of blink interval and duration for voluntary blinks detection, it's necessary to find out the boundary of interval and duration of spontaneous blinks. To this end, we collected EEG dataset of 12 subjects (age range between 21 to 25; 1/3 are female and 2/3 are male) who performed video-watching tasks. As shown in Fig.4, the subject, attaching wet electrodes with Ten20 conductive paste on the Fp1 and two earlobes, sat in front of two monitors, one plays videos and one visualize the EEG data though the OpenBCI GUI [36]. The PC of the monitors is in the same LAN as the OpenBCI Cyton which collecting and sending EEG data to the OpenBCI GUI though a WiFi shield.

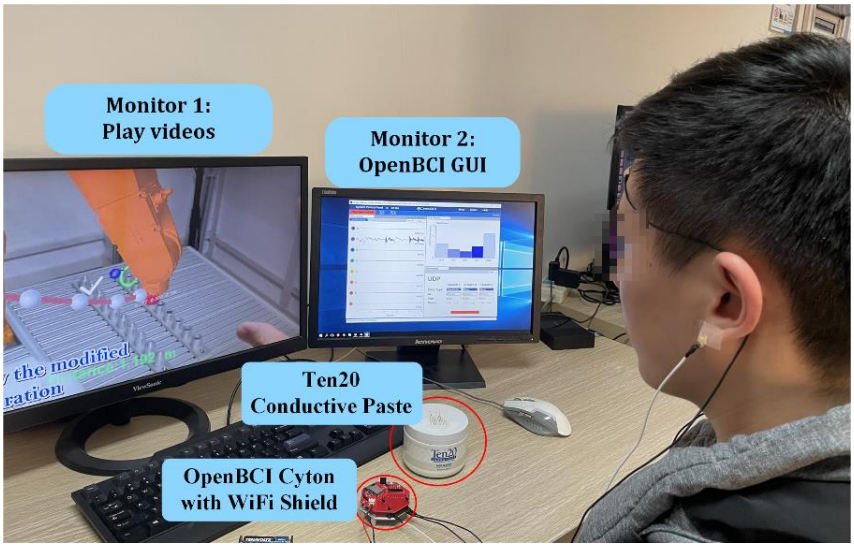

Fig.4. The setup of EEG data collection.

The dataset then was processed by a normal blinks detection algorithm *Blink* [25], and the detection result of the subject 1 is shown in Fig.5 (x axis: time-domain in seconds; y axis: voltage-domain in microvolt; the gray shades indicate the all detected eye blinks and their widths represent the duration of each blink). The blink duration and interval of all blinks then was obtained. After that, we sample randomly sample 70 sets of data from each subject for statistics, whose result is shown in Fig.6. From the result, we find that the duration of the normal blinks from all subjects hold a median between 200ms to 300ms with an upper limit lower than 500ms. As for blink interval, all medians are larger than 1s and the lower limit is around 700ms. With these finding, to clearly distinguish between normal and voluntary blink, we safely set the duration threshold to 600ms (a long blink should last over 600ms) and set the interval threshold to 600ms (two eye-closing actions of a double blink should perform within 600ms).

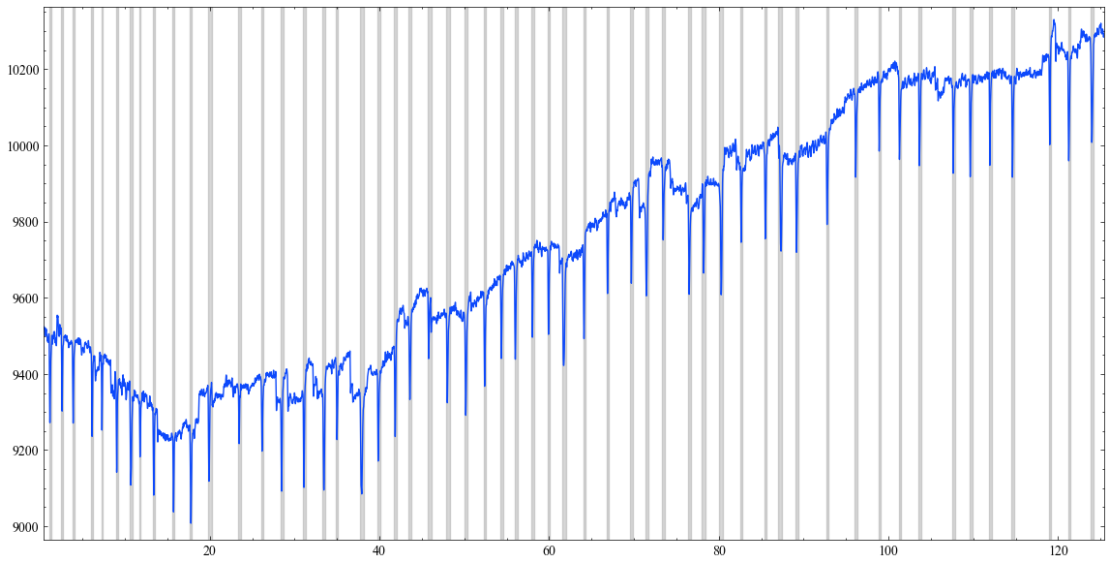

Fig.5. The result of normal blinks detection of the subject 1. Gray shades indicate all detected eye blinks and their widths represent the duration of each blink.

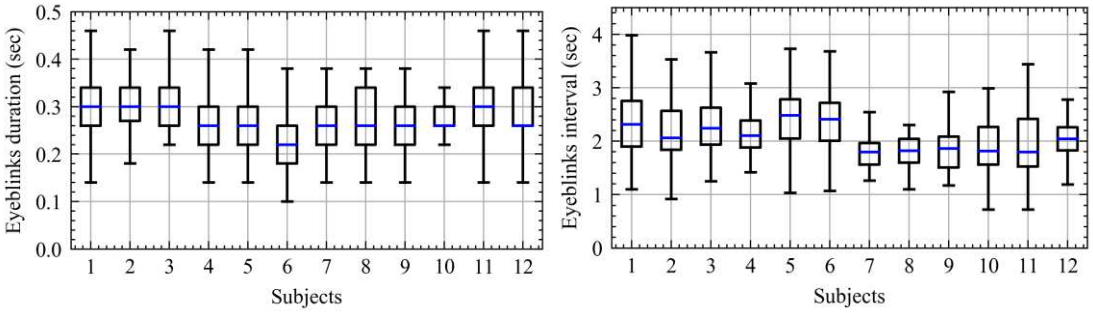

Fig.6. The duration (left) and interval (right) of the normal eye blinks.

4.2 Multiple voluntary eye blink detection algorithm

The aim of multiple voluntary eye blink detection algorithm is to recognize the long blink and the double blink from the EEG raw data without being disturbed by noise and the normal blink. Firstly, the patterns of the eye blinks of different kinds from a single subject in a short period (e.g., during data acquisition) have repeatability in waveform (e.g., the trough pattern) and voltage domain (e.g., the amplitude), which is the distinct feature compared to noise generally. Secondly, as mentioned above, the time-domain features of long blink, double blink and normal blink are distinguishable, especially the blink duration and interval.

Algorithm 1: Multiple voluntary eye blink detection

Parameter : *duration*, *interval*: the duration and interval threshold for detection; *threshold*1, *threshold*2: the correlation coefficient threshold in the first and second pass.

Input : $\mathbf{X}$: EEG raw data.

Output : $\mathbf{T}_{double}$, $\mathbf{T}_{long}$: the start and end time of all double blinks and long blinks.

1. Pre-process: $\mathbf{E} \leftarrow lowpass(\mathbf{X})$
2. Detect local minimums: $\mathbf{t}_{min} \leftarrow \text{dectect\_mins}(\mathbf{E}, delta{=}80)$
3. Classify Double-peaks: $\mathbf{t}_{signal}, \mathbf{t}_{double} \leftarrow \text{classify\_double\_peaks}(\mathbf{t}_{min}, interval)$
4. Detect blinks of all kinds:
   $\mathbf{T}_{double} = [\mathbf{t}_{start} \quad \mathbf{t}_{end}] \leftarrow \text{detect\_blinks}(\mathbf{X}, \mathbf{t}_{double}, threshold1, threshold2)$
   $\mathbf{T}_{singal} = [\mathbf{t}_{start} \quad \mathbf{t}_{end}] \leftarrow \text{detect\_blinks}(\mathbf{E}, \mathbf{t}_{signal}, threshold1, threshold2)$
5. Classify Long Blinks: $\mathbf{T}_{long}, \mathbf{T}_{normal} \leftarrow classify\_long\_blinks(\mathbf{T}_{singal}, duration)$
6. **return** $\mathbf{T}_{double}$, $\mathbf{T}_{long}$

Given the above knowledge, we propose a clustering-based unsupervised multiple voluntary eye blink detection algorithm which do not need any training session before testing. The algorithm is based on the assumption that (1) the EEG data to process contains more than one patterns of each type of eye blink, which requires the subjects to perform repeatedly all types of eye blink in the EEG recording period; and (2) there is no other repetitive noise with a similar waveform as eye blink. The second assumption is easy to achieve as the EEG signal from the electrode at forehead scalp is mainly affected by the motion of eyes, head and muscles [37], all of which, except eye blink, are either without trough-shaped EEG pattern or generally non-repetitive.

The proposed algorithm has some characteristics. Firstly, just only one EEG channel's data is required which helps to achieve the low-cost and ergonomics human-robot communication channel; secondly, just four parameters are needed (two thresholds related to the time-domain feature, and two thresholds related to the waveform feature), and these parameters are easy to be fine-tuned; thirdly, the targeted voluntary eye blinks are easy to be performed by users and thus the users are away from any training requirements.

Algorithm Explanation: (1) after collecting the raw EEG data **E,** the first procedure is pre-processing which is to filter the high-frequent component of the raw data; (2) The first step of the algorithm is to detect the peaks of the filtered data (see Fig.3), which are the local minimums with a threshold *delta* (=80μA, initially); (3) according to the intervals of the detected peaks, classify the detected peaks into double-peak and signal-peak with a threshold *interval*=600ms which is determined in Section 4.1; (4) input the classified double-peaks and signal-peaks into subroutine *detect_blink* respectively to obtain the final result of the double-blinks detection $\mathbf{T}_{double}$ and single-blinks detection $\mathbf{T}_{singal}$ which include long blinks and normal blinks; (5) classify the long blinks $\mathbf{T}_{long}$ from $\mathbf{T}_{singal}$ with the a threshold *duration*=600ms which is determined in Section 4.1. So far, the $\mathbf{T}_{double}$ and $\mathbf{T}_{double}$ are the final output the algorithm.

Subroutine 1: subroutine *detect_blink* for blinks detection based on feature extraction and cluster analysis

Parameter : *corr_th*1, *corr_th*2: the thresholds in the first and second pass.
Input : **E**: EEG data; $\mathbf{t}_{min}$: time of all detected local minimums.
Output : $\mathbf{T} = [\mathbf{t}_{start} \quad \mathbf{t}_{end}]$: the start and end time of all detected blinks.

1. Find all extreme points near the input $\mathbf{t}_{min}$:
   $$\mathbf{T}_{blink} = \left[\mathbf{t}_{start} \quad \mathbf{t}_{end}\right] \leftarrow \text{find\_stable\_points}\left(\mathbf{X},\ \mathbf{t}_{min}\right)$$
2. **for** $i = 1, 2, ..., len\left(\mathbf{T}_{blink}\right)$ **do**
3. **for** $j = i+1, ..., len\left(\mathbf{T}_{blink}\right)$ **do**
4. $\mathbf{sig}_i = \mathbf{E}\left[\mathbf{t}_{start}^{(i)} : \mathbf{t}_{end}^{(i)}\right]$; $\mathbf{sig}_j = \mathbf{E}\left[\mathbf{t}_{start}^{(j)} : \mathbf{t}_{end}^{(j)}\right]$
5. $\mathbf{Corr}^{(ij)} = \mathbf{Corr}^{(ji)} \leftarrow \text{corr\_coef}\left(\mathbf{sig}_i, \mathbf{sig}_j\right)$;
6. $\mathbf{Power}^{(ij)} = \mathbf{Power}^{(ji)} \leftarrow \max\left(\text{std}(\mathbf{sig}_i)/\text{std}(\mathbf{sig}_j), \text{std}(\mathbf{sig}_i)/\text{std}(\mathbf{sig}_j)\right)$.
7. **end**
8. **end**
9. Compute the index of matrix **Corr** that is larger than *threshold* (=*corr_th*1, relatively low to ensure detection robust) using Eq.(1).
10. Cluster and select the group with larger amplitude:
    $$\mathbf{index} \leftarrow \text{clustering}\left(\frac{\mathbf{Corr}^{(\mathbf{index\_corr})}}{\mathbf{Power}^{(\mathbf{index\_corr})}}\right)$$
11. Compute new *delta* from the selected group;
12. Detect peaks: $\mathbf{t}_{min} \leftarrow \text{detect\_peaks}\left(\mathbf{E}, delta\right)$
13. Repeat *steps 1 to 11* (in s*tep 10*, let threshold = *corr_th*2 which is relatively high to ensure detection correct)
14. **return** $\mathbf{T} \leftarrow \mathbf{T}_{blink}^{(\mathbf{index})}$

Subroutine *detect_blink*, following [25], detects the blinks based on time-domain feature and cluster analysis. Firstly, the extreme points (see Fig.3) are found by forward and backward scanning the signal power in the vicinity of each local minimum to identify the closest stable points (line 1). In the limited time window near the *i*th local minimum $\mathbf{t}_{\min}^{(i)}$, only when both stable points (the forward one $\mathbf{t}_{start}^{(i)}$ and the backward one $\mathbf{t}_{end}^{(i)}$) are found, its corresponding extreme points $\mathbf{T}_{blink}^{(i)}$ can be returned. Secondly, the correlation coefficient (time-domain similarity) matrix **Corr** and the power ratio (voltage-domain similarity) matrix **Power** of each two signal segments corresponding to the found $\mathbf{T}_{blink}$ are computed (line 2 to 8). Thirdly, the indexes of $\mathbf{T}_{blink}$ with the highly correlation **index_corr** are computed by Eq.(1), using a correlation threshold to find the time-domain similar patterns (line 9). Fourthly, combining the time-domain and voltage-domain features, the **index_corr** are clustered into two groups and the lager average power group is selected as the index of detected blinks (line 10). Finally, in order to detect the missing eye blink patterns, a new *delta* for detecting local minimums of the EEG signal is computed and the second pass of eye blinks pattern detection is done (line 11 to 13). Then, the start and end time of all detected eye blinks will be returned.

$$\begin{cases} i = \arg\max\left(\mathrm{sum}\left(\mathbf{Corr}\right)\right) \\ \mathbf{index\_corr} = \left[(i,j) \text{ if } \mathbf{Corr}^{(ij)} > threshold,\ \text{for } j=1,\cdots,\mathrm{len}\left(\mathbf{Corr}\right)\right] \end{cases} \tag{1}$$

4.3 Online detection approach

The above algorithm is based on time-domain and voltage-domain feature extraction and cluster analysis, by which eye blinks pattern can be detected in an unsupervised and training-free way. However, due to the characteristic of cluster analysis, it needs to collect EEG data over a relatively long period of time, in which multiple eye blinks of various types should occur. Thus, it is hard to perform in real time. To address the problem of that, we propose an online detection approach to make full use of the above algorithm, as shown in Fig.7.

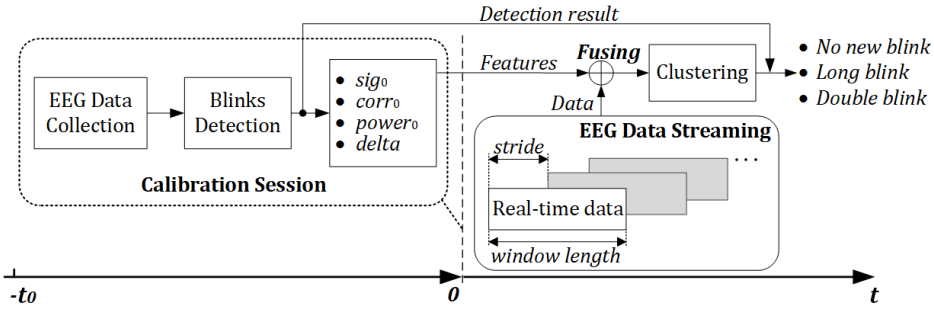

Fig.7. Multiple voluntary eye blinks online detection approach.

The approach includes a calibration session lasting a period of t_0 which is done before the online detection. In the calibration session, an EEG dataset with repeated patterns of three types of eye blinks (long blink, double blink and normal blink) is collected. The dataset then is input to the eye blink detector based on the proposed algorithm, after which the detection n times eye blinks and the features (including signal segments $\mathbf{Sig}_0 = [\mathbf{sig}_1, \mathbf{sig}_2, \ldots, \mathbf{sig}_n]$, correlation coefficient matrix

$\mathbf{Corr}_0 \in \mathbb{R}^{n\times n}$ and power ratio matrix $\mathbf{Power}_0 \in \mathbb{R}^{n\times n}$ of all detected eye blinks, and the computed threshold *delta*) are obtained. After calibration session, an EEG data streaming channel is established and a fixed-length time window is used to capture the streaming EEG data with a fixed stride. In the time step of *t*, the captured EEG data is to detect its local minimum with threshold *delta*, and then find its extreme points based on the local minimum (if detected). If the extreme points can be found, the signal segment between them $\mathbf{sig}_t$ is obtained. With the $\mathbf{Corr}_0$ from the calibration session, an extended correlation coefficient matrix $\mathbf{Corr}_t$ can be computed by Eq.(2).

$$\begin{cases} \mathbf{Corr}_t = \begin{bmatrix} \mathbf{Corr}_0 & \mathbf{corr}_t^T \\ \mathbf{corr}_t & 1 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)} \\ \mathbf{corr}_t \in \mathbb{R}^n \\ \mathbf{corr}_t^{(i)} = \text{corr\_coef}\left(\mathbf{sig}_i, \mathbf{sig}_t\right),\ i = 1,\ldots,n \end{cases} \tag{2}$$

Similarly, an extended power ratio matrix $\mathbf{Power}_t$ is obtained by Eq.(3).

$$\begin{cases} \mathbf{Power}_t = \begin{bmatrix} \mathbf{Power}_0 & \mathbf{power}_t^T \\ \mathbf{power}_t & 1 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)} \\ \mathbf{power}_t \in \mathbb{R}^n \\ \mathbf{power}_t^{(i)} = \max\left({}^{\text{std}(\mathbf{sig}_i)}\!\!\diagup\!\!{}_{\text{std}(\mathbf{sig}_t)}, {}^{\text{std}(\mathbf{sig}_t)}\!\!\diagup\!\!{}_{\text{std}(\mathbf{sig}_i)} \right), i = 1,\ldots,n \end{cases} \tag{3}$$

After that, the concatenated patterns $\mathbf{Sig}_t = [\mathbf{Sig}_0\ \mathbf{sig}_t]$ are clustered based on the feature ${}^{\mathbf{Corr}_t}\!\!\diagup\!\!{}_{\mathbf{Power}_t}$. Finally, the comparison of the clustering result and the result of calibration session is done and the online detection result (idle, long blinks detected, or double blinks detected) can be obtained.

5. Augmented reality-enabled feedback interface

The above voluntary blinks online detector can monitor the human intentional input. However, only two patterns, long blink and double blink, can be recognized which make it almost impossible to accomplish a complex robot control purely using the limited types of EEG patterns. To this gap, we are inspired by the graphical user interface (GUI) of the today's computer which presenting users suggestive or feedback visual contents on the monitor so that users can use the simple inputs (like clicking the mouse) to achieve the conveying of their intention. To transfer the interaction scenario from the 2D monitor to the 3D real world, we introduce a wearable augmented reality headset to provide human some augmented visual feedbacks in the corresponding to their inputs, making the above-proposed brain-computer interface a close-loop one. The rest of this section will present how to implement the four components of the proposed AR-enabled feedback interface: the control logic, the interactive path planning, and the command handling.

5.1 Control logic

In order to incorporate the eyeblinks into the human-robot interaction, we design a control logic to process the human's input. As shown in Fig.10, the designed control logic is based on the finite state machine (FSM). In the FSM, four states are defined: (1) *Standing-by (StdB)*, (2) *Trajectory*

modification (TrajM), (3) *Motion preview (MPrev)*, and (4) *Command handling (CmdH)*; and in any state, the human can input two types of action, *long blink* (in solid arrow) and *double blink* (in dotted arrow), to control the transition of the current state. To facilitate memory, the *long blink* is defined as the action that can change the current state forward; and the *double blink* is defined as the action that can either change the current state backward (in state 1 and state 3) or perform operations within the current state (in state 2 and state 4).

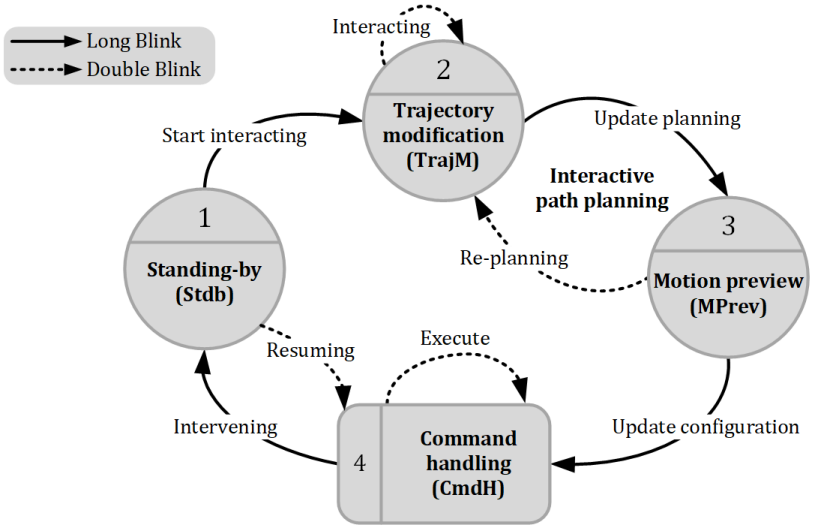

Fig.10. The schematic diagram of the control logic.

In the *StdB* state, the robot stops and is awaiting to the human's inputs. If the human performs a *double blink*, the robot would keep the original configuration and resume to execution state. If the human performs a *long blink*, the robot would enter the *TrajM* state and start listening to the interactive information conveyed from human.

The combination of the *TrajM* state and the *MPrev* state could enable human to perform an interactive robotic path planning with a modify-and-preview process. In the *TrajM* state, the human could modify the robot's planning by visualizing and editing the trajectory using the BCI and the AR headset, which will introduce in the following section. In the *MPrev* state, the modified planning would drive a motion simulation with a 3D virtual robot as a preview, which can give human an intuitive feedback corresponding to their intentional inputs.

After the interactive robotic path planning, the confirm-updated configuration will be used to generate the robotic executable command in the *CmdH* state after a *long blink* by human. And the robot executes the generated command after a *double blink*. During the robot operating, the human can impose an intervention to make robot into the *StdB* state.

5.2 AR-enabled interactive robotic path planning

In our previous work [8], we have developed an interactive path planning using the hand gestures with an AR feedback. In this work, we combine the voluntary eye blinks and the head motion into AR-enabled interaction to achieve a hand-free robotic path planning. The last section has introduced how the voluntary eye blink change the system state. This section will introduce the

interactive method to perform path planning by using head motion capture and AR visual feedback, which lays on the *TrajM* and *MPrev* states.

The AR-enabled interactive robotic path planning consists of (1) trajectory specification with head motion, and (2) AR-based motion preview. In the trajectory specification, the human's head motion is tracked and is mapped into the change of the trajectory of the robot's end-effector (EE). In the motion preview, a virtual robot is built in an AR scenario and simulated the motion with the modification trajectory.

5.2.1 Trajectory specification based on head motion capture

The coordinate systems of the AR headset, robot, and the HRC workplace are shown in Fig.11. In this setting, the robot's motion target is specified according to the instant pose of human head, and is visualized though the AR headset.

At the time step t, the system enters the *TrajeM* state with a human head pose which is represented with a coordinate system **H**. The robot's target appointed by human can be initialized as $\mathbf{P_t}$ by Eq.(7), where **Reach** is the space of the robot's reach, ${}^{w}\mathbf{P}_t = [{}^{w}x \quad {}^{w}y \quad {}^{w}z]$ and ${}^{h}\mathbf{P}_t = [{}^{h}x \quad {}^{h}y \quad {}^{h}z]$ are the coordinates of $\mathbf{P_t}$ relative to coordinate system **W** and **H** respectively, and ${}^{h}_{w}\mathbf{T}$ is the transformation matrix from coordinate system **W** to **H** which consists of a 3X3 rotation matrix ${}^{h}_{w}\mathbf{rot}$ and a 3-dimention translation vector ${}^{h}_{w}\mathbf{trans}$. The ${}^{w}_{h}\mathbf{T}$ is obtained with PTC's Vuforia™ SDK [38] by dynamically tracking a 2D marker attached on the workplace.

$$\begin{cases} {}^{w}\mathbf{P}_t = [{}^{w}x \quad {}^{w}y \quad {}^{w}z] \in \mathbf{Reach} \\ {}^{h}\mathbf{P}_t = \left({}^{w}_{h}\mathbf{T}\right)^{-1} * [{}^{w}\mathbf{P}_t \quad 1]^T = [{}^{w}_{h}\mathbf{rot} \quad {}^{w}_{h}\mathbf{trans}]^{-1} * [{}^{w}\mathbf{P}_t \quad 1]^T \end{cases} \tag{7}$$

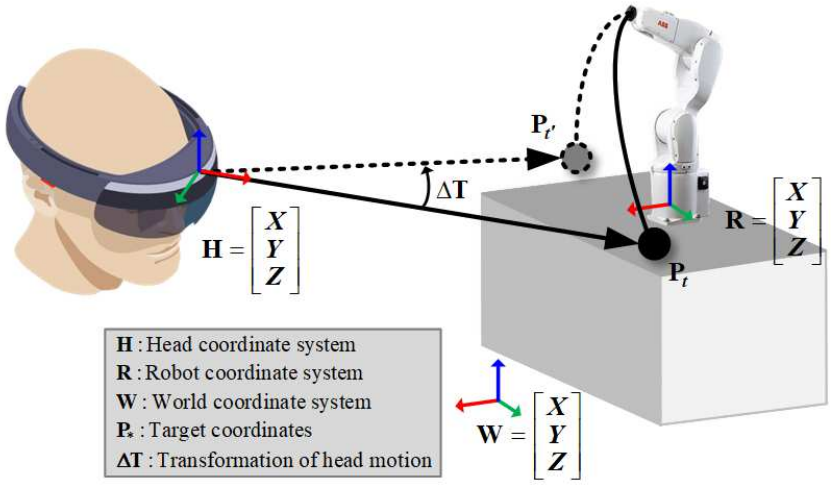

Fig.11. The coordinate systems of the AR headset, robot and HRC workplace.

Then at every updated moment t`, the AR headset can capture the motion of the human head represented by the transformation matrix of head motion $\mathbf{\Delta T}$ which consists of a 3X3 rotation matrix $\mathbf{\Delta rot}$ and a 3-dimention translation vector $\mathbf{\Delta trans}$. With the motion of human head, the appointed robot's motion target $\mathbf{P}_{t'}$ will change simultaneously based on Eq.(8) where the ${}^{r}_{h}\mathbf{T}$ is the

transformation matrix from coordinate system **H** to **R**, which consists of a 3X3 rotation matrix ${}^{r}_{h}\mathbf{rot}$ and a 3-dimention translation vector ${}^{r}_{h}\mathbf{trans}$. Since the robot is stationary relative to the workplace, the ${}^{r}_{h}\mathbf{T}$ can be measured and calculated in advance.

$$\begin{cases} {}^{h}\mathbf{P}_{t'} = \mathbf{\Delta T} * [{}^{h}\mathbf{P}_{t} \quad 1]^{T} = [\mathbf{\Delta rot} \quad \mathbf{\Delta trans}] * [{}^{h}\mathbf{P}_{t} \quad 1]^{T} \\ {}^{r}\mathbf{P}_{t'} = {}^{r}_{h}\mathbf{T} * [{}^{h}\mathbf{P}_{t'} \quad 1]^{T} = [{}^{r}_{h}\mathbf{rot} \quad {}^{r}_{h}\mathbf{trans}] * [{}^{h}\mathbf{P}'_{t} \quad 1]^{T} \\ {}^{r}_{h}\mathbf{T} = {}^{r}_{w}\mathbf{T}\,{}^{w}_{h}\mathbf{T} \end{cases} \tag{8}$$

The $\mathbf{P}_{t'}$ will be determined after human performing a *long blink*, and it is used to calculate the motion trajectory, as well as driving the AR-based motion preview.

5.2.2 AR-based motion preview as feedback

The AR scenario for motion preview is built in the Unity3D. As shown in Fig.12, a virtual robot has been built in Unity3D which has the same size, shape, axis types and hierarchy structure as the real robot ABB IRB1200.

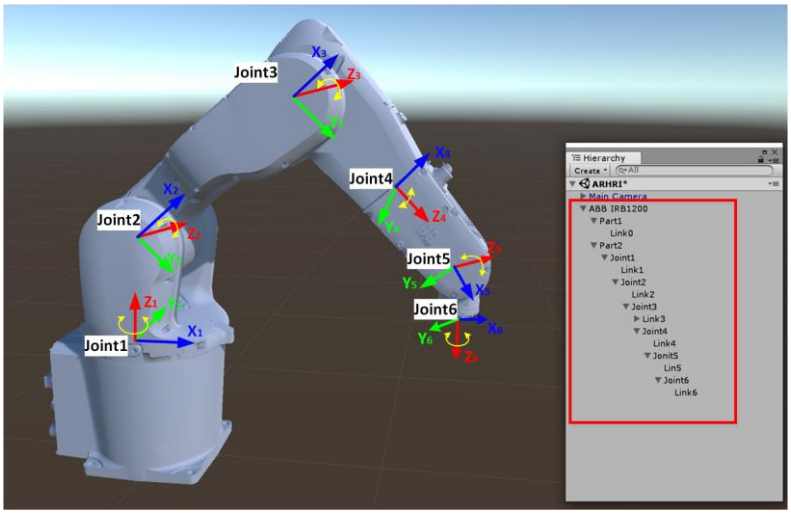

Fig.12. The virtual model and the hierarchy structure of the robot.

After the robot motion target determined and the system entering the *MPrev* state, the determined $\mathbf{P}_{t'}$ is used to computed the robot's joints position $\mathbf{q}_{t'}$ by the inverse kinematic solver. The computed $\mathbf{q}_{t'}$ then drives the built virtual robot built in the AR scenario to run a motion simulation as a preview corresponding to the specified robot configuration. The robotic motion preview animates in the developed AR scenario with a with a refresh rate *f* in frames per second (fps). In every frame of the AR scenario rendering, the refreshed joints position of the virtual robot $\mathbf{q} = [q_1, q_2, q_3, q_4, q_5, q_6]$ is computed by Eq.(9):

$$q_i = \begin{cases} \min\left(\mathbf{q}_0^{(i)} + \dot{\mathbf{q}}_t^{(i)} / f, \mathbf{q}_{t'}^{(i)}\right), \mathbf{q}_{t'}^{(i)} > \mathbf{q}_0^{(i)} \\ \max\left(\mathbf{q}_0^{(i)} - \dot{\mathbf{q}}_t^{(i)} / f, \mathbf{q}_{t'}^{(i)}\right), \mathbf{q}_{t'}^{(i)} < \mathbf{q}_0^{(i)} \end{cases}, i = 1, \ldots, 6 \tag{9}$$

where $\mathbf{q}_0^{(i)}$ and $\mathbf{q}_{t'}^{(i)}$ is the initial and the target position of the *i*th joint respectively, and

$\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6]$ is the joints' speed consistent with real robot settings. At the same time of motion simulation, the EE's trajectory is also rendered though a visual curve and some key points.

According to the motion simulation, the human can either confirm to the current configuration or re-plan the presenting trajectory. After confirming, the final configuration will be handled to generate the command for robot execution.

5.3 Command handling

The input to the command handling module is the trajectory of the robot's EE, namely the time series of the points of robotic planned paths, which is defined by human though the BCI-AR interface. This module is to translate the input information into the robotic executable command which can reconfigure the control program on-the-fly. To realize that, we designed a program for the robot controller with a structure as shown in Fig.13. The program has three key variables which are the velocity of EE *vel*, point series of EE *point* and the orientation of EE *ori*. The *point* is the defined by human though the interactive path planning presenting in the above section, which contains Cartesian coordinates of the path. The *vel* can be adjusted according to the safety status of the HRC workplace (e.g., *vel* reduce with the human-robot distance being closer [8]). The *ori* is defined by the task specification (e.g., in pick-and-place tasks, *ori* should be well-defined to achieve stable object grasping) which could be retrieved in a pre-defined knowledge library.

To make the I/O operation of the robot controller, we implemented this module based on the ABB RobotStudio [39] which can establish a connection to the controller and a developed C# software with ABB PC SDK[40] which can stablish a connection with RobotStudio to read the robot running data and to modify the variable in the controller program.

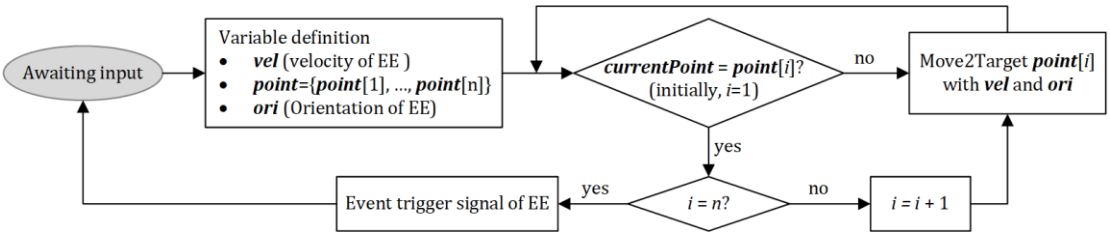

Fig.13. The structure of the robot program.

6. Case study

In this section, we firstly evaluate the performance of the algorithm we proposed in Section 4 with the comparison with the related work; then we develop an industrial HRC assembly case to show the effectiveness of the proposed BCI-AR interface with the comparison with a hand gesture-based AR interface.

6.1 Performance of the voluntary eye blink detection algorithm

In order to verify the effectiveness of the proposed algorithm, we collected a dataset to test its performance of multiple voluntary blinks detection. The dataset is collected from the same 12 subjects and in the same setup as the previous test presented in Section 4.1. The subjects performed

the action with the guidance of a visual clue showing on the monitor while recording EEG data. As shown in Fig.8, each subject was guided to perform 40 times long blink, double blinks and idle status (the subject could perform normal blink or stay still).

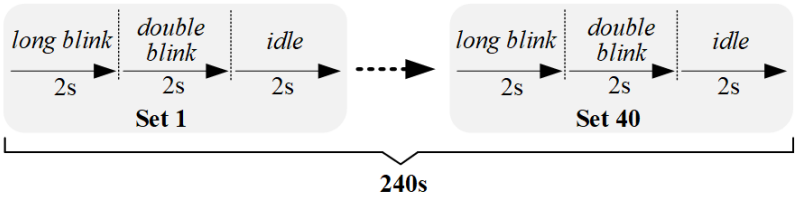

Fig.8. The protocol of EEG data collection with visual clue for each subject.

The collected dataset with the size of 1,440 then is processed by the proposed algorithm implemented in Python. The detection result is presented as a confusion matrix shown in Fig.9. The accuracy, precision and sensitivity of the detection is defined by Eq.(4) ~ Eq.(6), where the *TP* and *TF* are the number of the true positive and true negative; the *FP* and *FN* is the number of the false positive and the false negative.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$precision = \frac{TP}{TP + FP} \tag{5}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{6}$$

Based on that, the performance of algorithm for each subject is also calculated, as shown in Table.1. From that, we found that an average accuracy of 94.31% was achieved. Among all subjects, the best performance occurred in the subject 4 and 12 with an accuracy of 99.17%, while the worst performance occurred in subject 8 with an accuracy of 82.33%. Among all classes, the *double blink* class holds a biggest precision and the *idle class* holds a biggest sensitivity.

In addition, we have done some performance comparison with three related work. The first related work is based on Dynamic Positional Warping (DPW) [22], which is in the template matching style. The second and third related algorithms are both based on the supervised learning, which are based on SVM [23] and RBF [23] respectively. In this comparison experiment, we reused the 1,440-sized data set mentioned above. Due to all the related work needs a training phase, we used 60% (864) of the data set for training and 40% (576) for testing. For the DPW algorithm, we set the length and the stride of the slide windows to 1000ms (50% of the signal length) and 100ms (5% of the signal length), respectively. For the SVM-based algorithm, we optimized the parameters (*C* and *gamma*) by the comparing the performance of the different values of parameters. Similarly, for the RBF-based algorithm, we selected the optimal parameter (*gamma*) according to the same searching technique. The results are shown in Table.2. Our proposed algorithm has a higher average accuracy and precision. Moreover, compared with the related work in Table.2, our proposed algorithm, which needs a calibration session of only 45 seconds, does not need any training phase

or any labeled data.

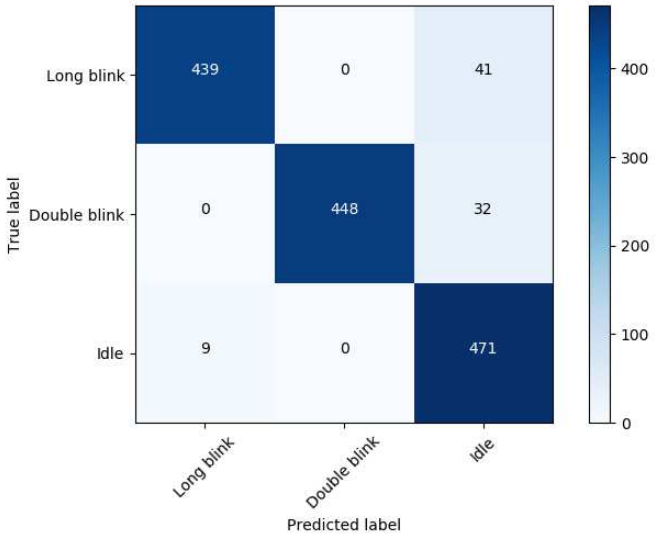

Fig.9. Confusion matrix of multiple voluntary detection for all subjects.

Table.1. Algorithm performance for each subject. ***Prec_ **** and ***Sens_ **** are for the detection precision and the sensitivity of each class (***L***: long blink, ***D***: double blink, and ***Idle***: the idle status).

Subject	***Accuracy***	***Prec_L***	***Prec_D***	***Prec_Idle***	***Sens_L***	***Sens_D***	***Sens_Idle***
1	93.33%	97.22%	100%	84.78%	87.5%	95%	97.5%
2	91.67%	92.31%	100%	84.09%	90%	92.5%	92.5%
3	95.83%	97.37%	100%	92.5%	97.5%	97.5%	92.5%
4	99.17%	100%	100%	97.56%	97.5%	100%	100%
5	95%	100%	100%	86.96%	90%	95%	100%
6	92.5%	92.5%	100%	86.05%	92.5%	92.5%	92.5%
7	97.5%	100%	100%	93.02%	95%	97.5%	100%
8	82.33%	94.97%	100%	75.00%	80%	87.5%	92.5%
9	94.17%	100%	100%	85.11%	87.5%	95%	100%
10	88.33%	100%	100%	74.07%	90%	75%	100%
11	96.67%	100%	100%	90.91%	95%	95%	100%
12	99.17%	100%	100%	97.56%	100%	97.5%	100%
Total	**94.31%**	**97.99%**	**100%**	**86.58%**	**91.46%**	**93.33%**	**98.13%**

Table.2. Performance comparison with the related work. ***Precision (L; D; I)*** is for the precision of the long blink, double blinks and the idle status, respectively.

Algorithm	***Accuracy***	***Precision (L; D; I)***	**Limitations**
Ours	94.31%	97.99%; 100%; 86.58%	Requires a calibration session (45 seconds in this case)
DPW [22]	84.54%	81.62%; 100%; 74.65%	Requires training phase for templates extraction
SVM [23]	76.74%	79.88%; 95.78%; 61.79%	Requires training phase which needs labeled data
RBF [23]	75.52%	82.89%; 97.16%; 60.35%	Requires training phase which needs labeled data

6.2 Case study of an industrial HRC assembly task

In this part, we develop a case study in an industrial HRC assembly task where human and robot, an ABB IRB1200, jointly perform a batch assembly tasks, as shown in Fig.14, to demonstrate the effectiveness of the proposed closed-loop interface combining BCI and AR.

The assembly task consists of two parts to be assembled. Human could use the BCI-AR interface to program the robot motion trajectory on-the-fly to assign which part is the robot to approach and pick. The robot is mastered by the robot controller ABB IRC5 deploying ABB RAPID robot programming. The controller and a PC are connected via Ethernet by which the running data and the control command transfer. The PC deploys a robot controller interface built upon the ABB PC SDK with C#, which can 1) monitor the robot status and modify its controller configuration, 2) establish the communication channel with the HoloLens via LAN, and 3) receive user input from the voluntary eyeblinks detector implemented by Python via TCP/IP protocol. The detector and the OpenBCI GUI are connected via the UDP protocol where an EEG data streaming is established and the online eyeblinks detection is realized. The AR headset achieved the tracking of a marker attached on the workplace to obtain the transformation between the AR headset and workplace.

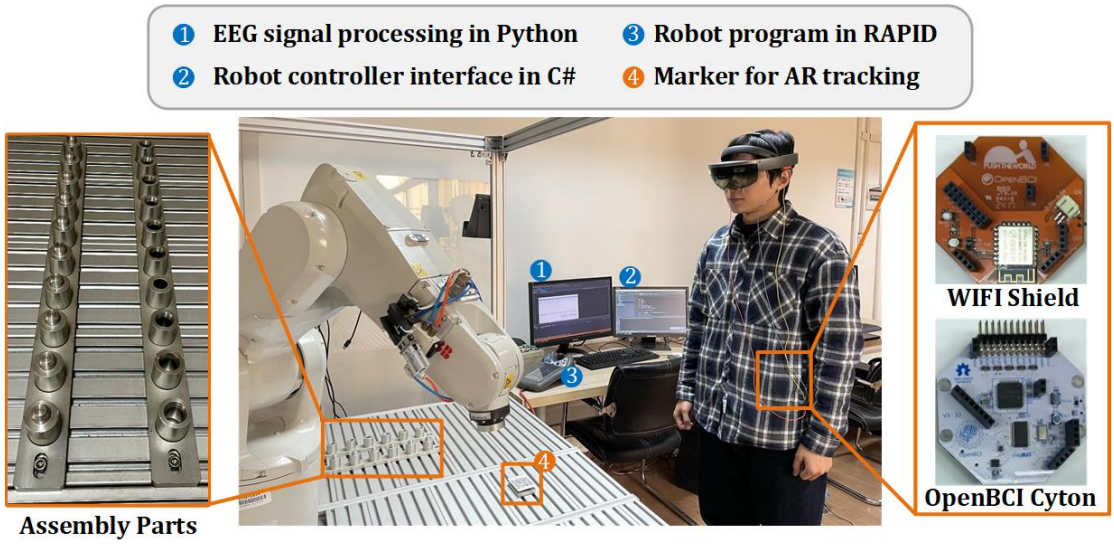

Fig.14. Case study setup.

Fig.15 and Fig.16 show the snapshots of a user's point of view from HoloLens and the collected EEG data during a single interaction. At the moment t=0s, the user performed a long blink which makes the interaction start. Then, a red sphere, pointing out the robot's target, was rendered in the center of the user's perspective with the same depth as the distance between the user's head and robot's TCP, and it can be move with the transformation of the user's perspective, as shown in Fig.15(2) to Fig.15(4). With a double blink performed at t=2.6s, the target was fixed. After a long blink performed at t=4.8s, a trajectory was computed and visualized as shown in Fig.15(5). The user moved his head and focused on a point in a visual trajectory. With a double blink at t=9.7s, the focused point was turned blue and could be moved with the transformation of head pose, by which could edit the previously computed trajectory, as shown in Fig.15(6) to Fig.15(8). After a double blink at t=12.3s and a long blink at t=15.1 s, the modified trajectory and the motion simulation was rendered as a preview, as shown in Fig.15(9) to Fig.15(12). After another long blink at t=22.3s, the

robot executed the rewritten configuration as shown in Fig.15(13) to Fig.15(16).

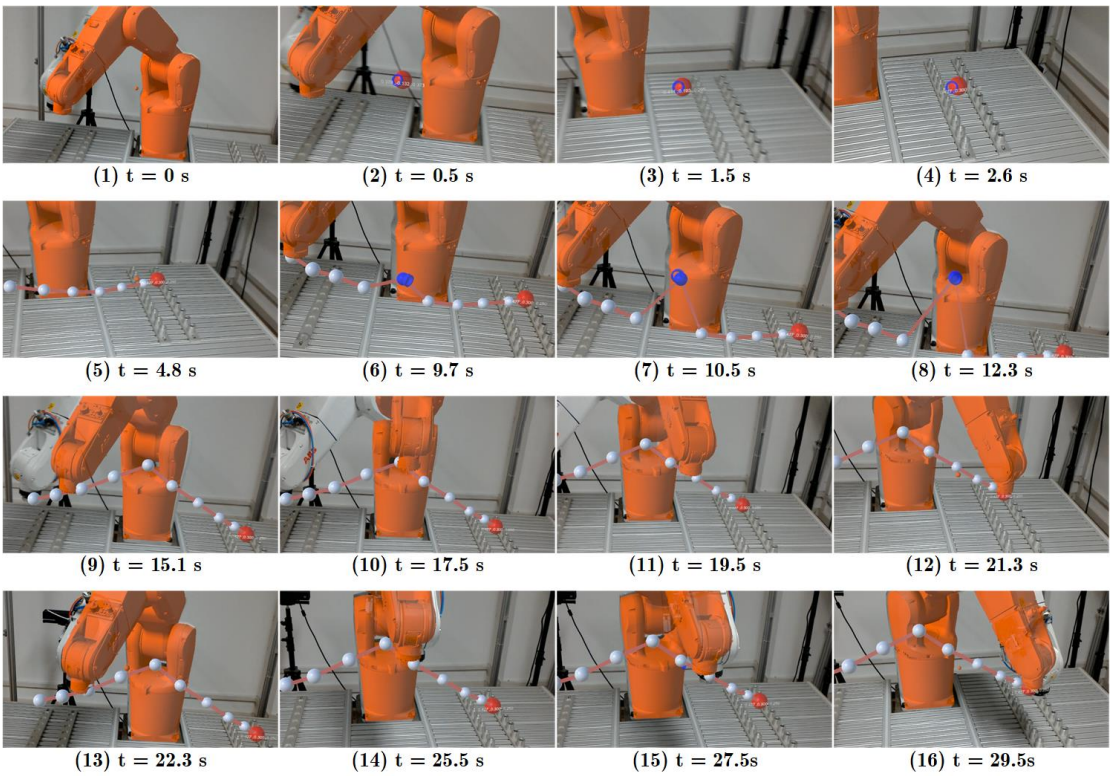

Fig.15. The snapshots of a user's point of view from HoloLens.

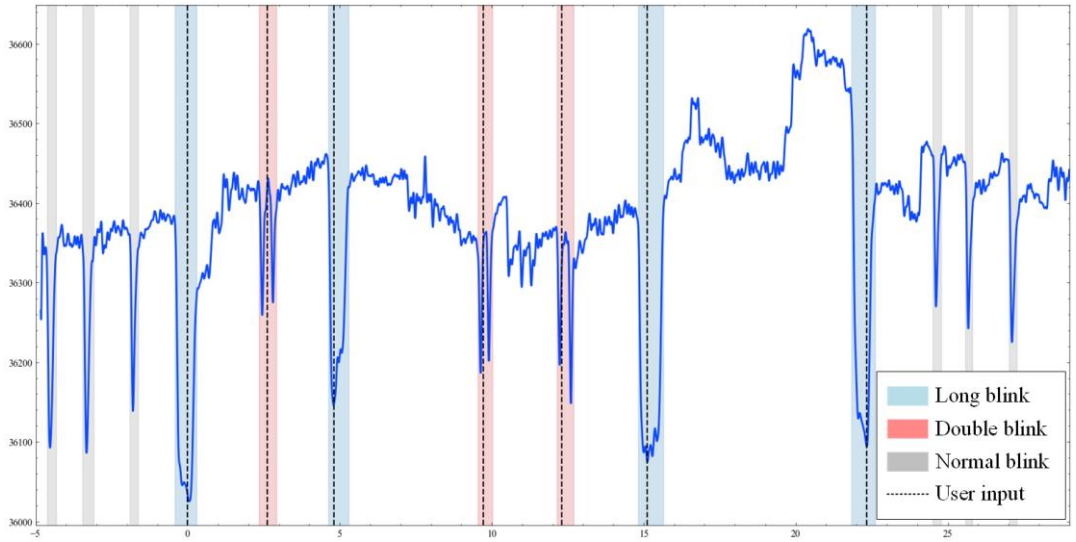

Fig.16. The recording EEG data and the result of eyeblink detection.

In order to better evaluate the performance of the presented system, we compared the performance of this system with [8] on some time indicators. In [8], as shown in Fig.17, the input of the AR-enabled human-robot interaction is based on the hand gestures (HG) recognition of the HoloLens SDK and depends on some AR UIs. In the comparative experiment, 12 users performed the interactive process as shown in Fig.15 with two systems for 5 times each after the user learning sessions, where three indictors were recorded: (1) the total interaction time (TIT) which is defined as the time from the beginning of the interaction to the completion of the robot reconfigured motion, e.g., 29.5s in Fig.15; (2) accumulative input time (AIT) which is defined as the accumulative time

in the interactive path planning, e.g., 22.3s in Fig.15; and (3) the average time per input (AT/I) which is defined as AIT divided by the number of user input, e.g., 22.3/7≈3.19s in Fig.15. Fig.18 shows the performance comparison on the above indicators between two systems. From that, we found that the proposed BCI-AR system has better performance on all indicators with the comparison of the HG-AR system (23.38% shorter in TIT, 25.19% shorter in AIT, and 13.83% shorter in AT/I).

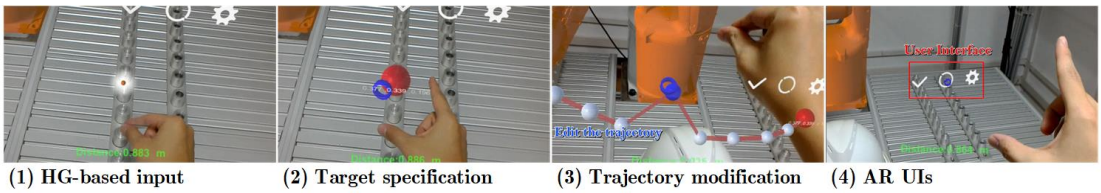

Fig.17. Examples of the user input in [8].

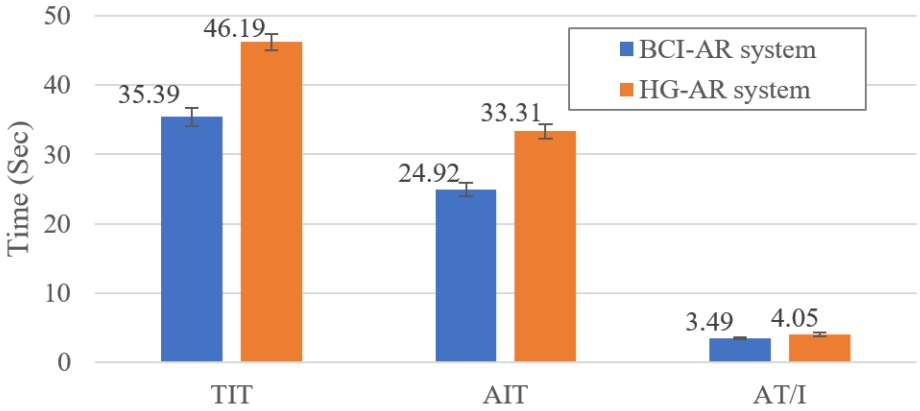

Fig.18. Performance comparison between the BCI-AR system and the HG-AR system. TIT: total interaction time; AIT: accumulative input time; AT/I: the average time per input. The numbers indicate the average values, and the error bars indicate the standard errors of mean (SEM).

7. Conclusion and future work

This paper, aiming to release the better potential of BCI and AR application in industrial HRC, proposed a close-loop interface of human-robot interaction using the human eyeblinks input and the AR feedback. First, we proposed an EEG-based voluntary blinks online detection algorithm which is in an automated unsupervised way and can achieve an average accuracy of 94.31% with stable performance across users. Then, we designed an AR-enabled feedback interface which could switch the interaction state according to the different input and achieve an interactive robotic path planning though the AR-based modify-and-preview process. Finally, we integrated the proposed method into a case study in a HRC assembly task. The case study shows that compared with the hand gesture-based input method, the proposed eye blink-based method can reduce the time of user input, which could improve the effectiveness of the communication between human and robot.

Our future work is listed as followed. Firstly, we will go more deeper into the other EEG patterns recognition (like motion imagery patterns, error-related potential and et al.) to achieve a more natural BCI. Secondly, based on the concept of shared control, the robot autonomy, namely the degree of autonomous decision-making of robot, will be properly improved which can reduce the

requirement of the human input in the BCI-based robotic control tasks. Thirdly, the proposed closed-up BCI could be used as an auxiliary channel for existing interaction solutions, and its integration into the existing interaction system which could build up a more robust multi-modal human-robot interaction will be further study.

Ethical Approval

The subjects involved in this research are all volunteers, and the authors warrant that the paper fulfills the ethical standards of the journal. No conflict of interest exists in the submission of this paper, and paper is approved by all authors for publication. This paper has not been published or presented elsewhere in part or in entirety and is not submitted to another journal.

Consent to Participate

The subjects involved in this research are all volunteers, the authors warrant that the paper fulfills the ethical standards of the journal.

Consent to Publish

All authors are consent to publish this paper in The International Journal of Advanced Manufacturing Technology.

Authors Contributions

Zhenrui Ji designed and conducted the case study, analyzed the results, and wrote the paper. Quan Liu proposed the basic idea and contributed the materials. Wenjun Xu proposed the method design and experimental idea and also modified this paper. Bitao Yao proposed the idea of EEG signal processing. Jiayi Liu analyzed the data and conducted the case study. Zude Zhou modified the structure of the paper and contributed the method development.

Funding

This research is supported by National Natural Science Foundation of China (Grant No. 51775399) and the Fundamental Research Funds for the Central Universities (WUT: 2020III047).

Competing Interests

The authors declare that no conflict of interest exists in the submission of this paper.

Availability of data and materials

The data and materials of this paper are available from the corresponding author on reasonable request.

Reference

1. Liu H, Wang L (2018) Gesture recognition for human-robot collaboration: A review. International Journal of Industrial Ergonomics 68:355-367
2. Cirillo A, Ficuciello F, Natale C, Pirozzi S, Villani L (2015) A conformable force/tactile skin for physical human–robot interaction. IEEE Robotics and Automation Letters 1 (1):41-48
3. Hollmann R, Hägele M The use of voice control for industrial robots in noisy manufacturing

environments. In: 39th International Symposium on Robotics, ISR 2008, 2008. pp 14-18
4. Mohammed A, Wang L (2018) Brainwaves driven human-robot collaborative assembly. CIRP Annals 67 (1):13-16. doi:https://doi.org/10.1016/j.cirp.2018.04.048
5. Mohammed A, Wang L (2020) Advanced Human-Robot Collaborative Assembly Using Electroencephalogram Signals of Human Brains. Procedia CIRP 93:1200-1205
6. Wolpaw JR, Birbaumer N, Heetderks WJ, McFarland DJ, Peckham PH, Schalk G, Donchin E, Quatrano LA, Robinson CJ, Vaughan TM (2000) Brain-computer interface technology: a review of the first international meeting. IEEE transactions on rehabilitation engineering 8 (2):164-173
7. Nijholt A, Tan D, Pfurtscheller G, Brunner C, Millán JdR, Allison B, Graimann B, Popescu F, Blankertz B, Müller K-R (2008) Brain-computer interfacing for intelligent systems. IEEE intelligent systems 23 (3):72-79
8. Ji Z, Liu Q, Xu W, Yao B, Hu Y, Feng H, Zhou Z Augmented reality-enabled intuitive interaction for industrial human-robot collaboration. In: 49th International Conference on Computers and Industrial Engineering (CIE 2019), 2019.
9. Carlson T, Millan JdR (2013) Brain-Controlled Wheelchairs: A Robotic Architecture. IEEE Robotics & Automation Magazine 20 (1):65-73. doi:10.1109/MRA.2012.2229936
10. Carabalona R, Grossi F, Tessadri A, Castiglioni P, Caracciolo A, de Munari I (2012) Light on! Real world evaluation of a P300-based brain–computer interface (BCI) for environment control in a smart home. Ergonomics 55 (5):552-563
11. Akram F, Han SM, Kim T-S (2015) An efficient word typing P300-BCI system using a modified T9 interface and random forest classifier. Computers in biology and medicine 56:30-36
12. Vialatte FB, Maurice M, Dauwels J, Cichocki A (2010) Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives. Prog Neurobiol 90 (4):418-438. doi:10.1016/j.pneurobio.2009.11.005
13. Donchin E, Spencer KM, Wijesinghe R (2000) The mental prosthesis: Assessing the speed of a P300-based brain-computer interface. Ieee Transactions on Rehabilitation Engineering 8 (2):174-179. doi:10.1109/86.847808
14. Pfurtscheller G, Neuper C (2001) Motor imagery and direct brain-computer communication. Proceedings of the IEEE 89 (7):1123-1134
15. Stern JA, Walrath LC, Goldstein R (1984) The Endogenous Eyeblink. Psychophysiology 21 (1):22-33. doi:https://doi.org/10.1111/j.1469-8986.1984.tb02312.x
16. Li Y, He S, Huang Q, Gu Z, Yu ZL (2018) A EOG-based switch and its application for "start/stop" control of a wheelchair. Neurocomputing 275:1350-1357. doi:https://doi.org/10.1016/j.neucom.2017.09.085
17. Molina-Cantero AJ, Lebrato-Vázquez C, Merino-Monge M, Quesada-Tabares R, Castro-García JA, Gómez-González IM (2019) Communication technologies based on voluntary blinks: Assessment and design. IEEE Access 7:70770-70798
18. Hosni SM, Shedeed HA, Mabrouk MS, Tolba MF (2019) EEG-EOG based virtual keyboard: Toward hybrid brain computer interface. Neuroinformatics:1-19
19. Chang W-D, Cha H-S, Kim K, Im C-H (2016) Detection of eye blink artifacts from single prefrontal channel electroencephalogram. Computer methods and programs in biomedicine 124:19-30
20. b Abd Rani MS Detection of eye blinks from EEG signals for home lighting system activation. In: 2009 6th International Symposium on Mechatronics and its Applications, 2009. IEEE, pp 1-4
21. Klein A, Skrandies W (2013) A reliable statistical method to detect eyeblink-artefacts from

electroencephalogram data only. Brain topography 26 (4):558-568
22. Chang W-D, Im C-H (2014) Enhanced template matching using dynamic positional warping for identification of specific patterns in electroencephalogram. Journal of Applied Mathematics 2014
23. Ghosh R, Sinha N, Biswas SK (2018) Automated eye blink artefact removal from EEG using support vector machine and autoencoder. IET Signal Processing 13 (2):141-148
24. Rihana S, Damien P, Moujaess T (2013) EEG-eye blink detection system for brain computer interface. In: Converging Clinical and Engineering Research on Neurorehabilitation. Springer, pp 603-608
25. Agarwal M, Sivakumar R (2019) Blink: A Fully Automated Unsupervised Algorithm for Eye-Blink Detection in EEG Signals. 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton):1113-1121
26. Si-Mohammed H, Sanz FA, Casiez G, Roussel N, Lécuyer A Brain-computer interfaces and augmented reality: A state of the art. In: Graz Brain-Computer Interface Conference, 2017.
27. Angrisani L, Arpaia P, Esposito A, Moccaldi N (2019) A Wearable Brain–Computer Interface Instrument for Augmented Reality-Based Inspection in Industry 4.0. IEEE Transactions on Instrumentation and Measurement 69 (4):1530-1539
28. Escolano C, Antelis JM, Minguez J (2011) A telepresence mobile robot controlled with a noninvasive brain–computer interface. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (3):793-804
29. Lampe T, Fiederer LD, Voelker M, Knorr A, Riedmiller M, Ball T A brain-computer interface for high-level remote control of an autonomous, reinforcement-learning-based robotic system for reaching and grasping. In: Proceedings of the 19th international conference on Intelligent User Interfaces, 2014. pp 83-88
30. Fang H, Ong S, Nee A (2014) A novel augmented reality-based interface for robot path planning. International Journal on Interactive Design and Manufacturing (IJIDeM) 8 (1):33-42
31. Lambrecht J, Krüger J Spatial programming for industrial robots based on gestures and Augmented Reality. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 7-12 Oct. 2012 2012. pp 466-472. doi:10.1109/IROS.2012.6385900
32. Quintero CP, Li S, Pan MK, Chan WP, Loos HFMVd, Croft E Robot Programming Through Augmented Trajectories in Augmented Reality. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1-5 Oct. 2018 2018. pp 1838-1844. doi:10.1109/IROS.2018.8593700
33. Rosen E, Whitney D, Phillips E, Chien G, Tompkin J, Konidaris G, Tellex S (2020) Communicating robot arm motion intent through mixed reality head-mounted displays. In: Robotics Research. Springer, pp 301-316
34. OpenBCI. https://openbci.com/.
35. Khazi M, Kumar A, Vidya M (2012) Analysis of EEG using 10: 20 electrode system. International Journal of Innovative Research in Science, Engineering and Technology 1 (2):185-191
36. OpenBCI GUI. https://github.com/OpenBCI/OpenBCI_GUI/.
37. Islam MK, Rastegarnia A, Yang Z (2016) Methods for artifact detection and removal from scalp EEG: A review. Neurophysiologie Clinique/Clinical Neurophysiology 46 (4):287-305. doi:https://doi.org/10.1016/j.neucli.2016.07.002
38. PTC Vuforia Engine. https://developer.vuforia.com/.
39. Connolly C (2009) Technology and applications of ABB RobotStudio. Industrial Robot: An

International Journal
40. Liu Q, Liu Z, Xu W, Tang Q, Zhou Z, Pham DT (2019) Human-robot collaboration in disassembly for sustainable manufacturing. International Journal of Production Research 57 (12):4027-4044

Figures

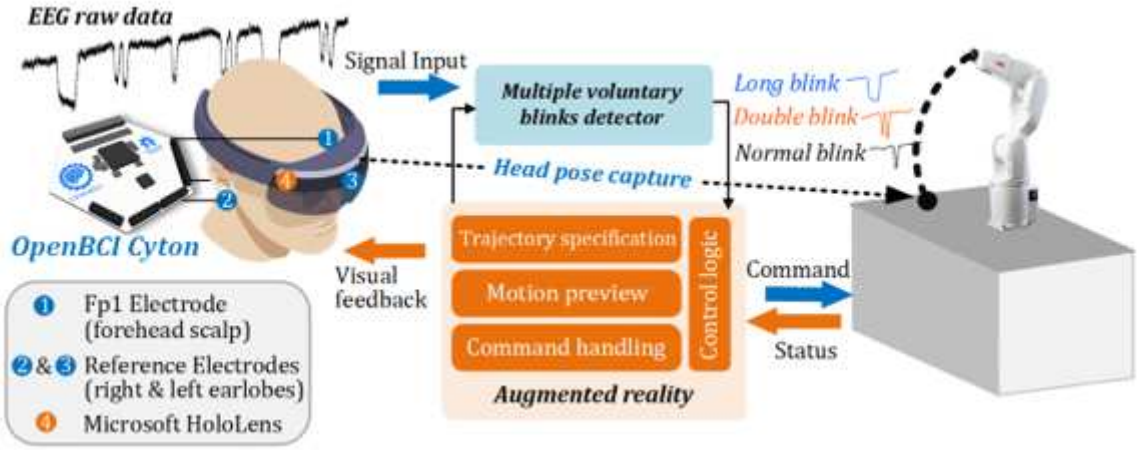

Figure 1

System overview.

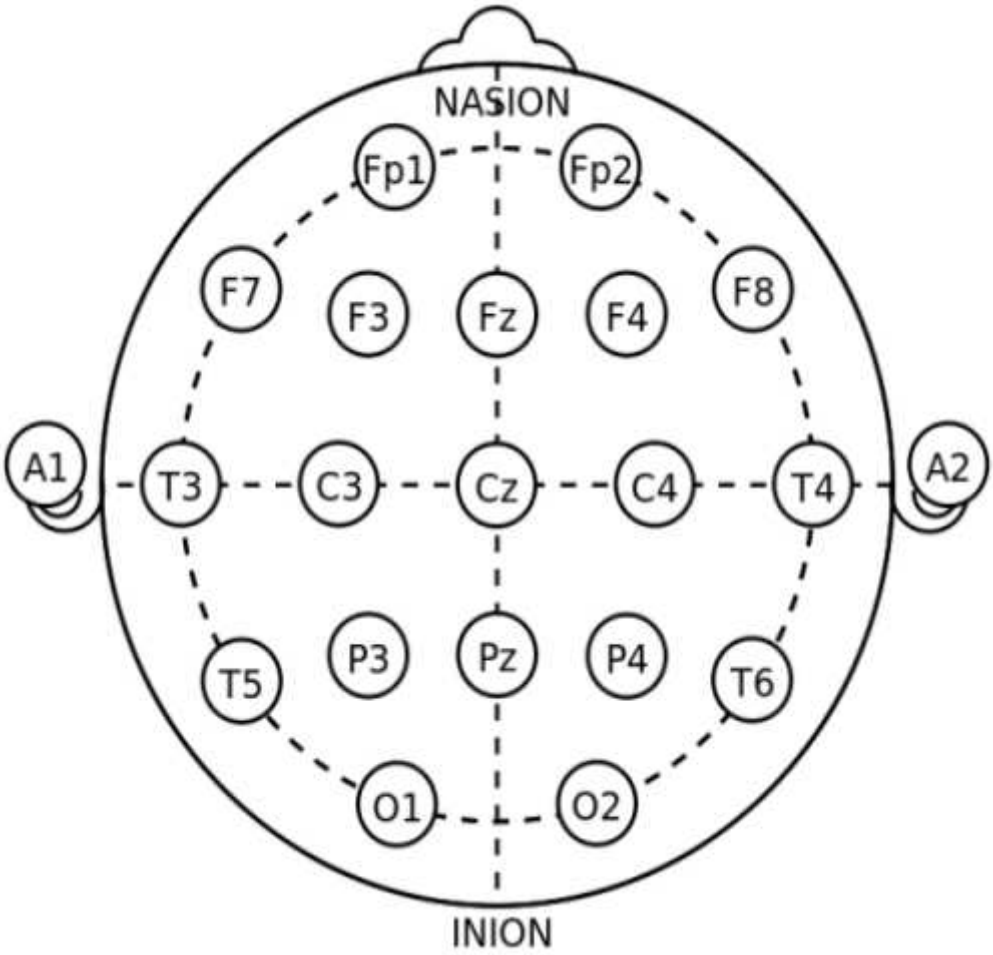

Figure 2

The international 10-20 EEG electrodes placement system [35].

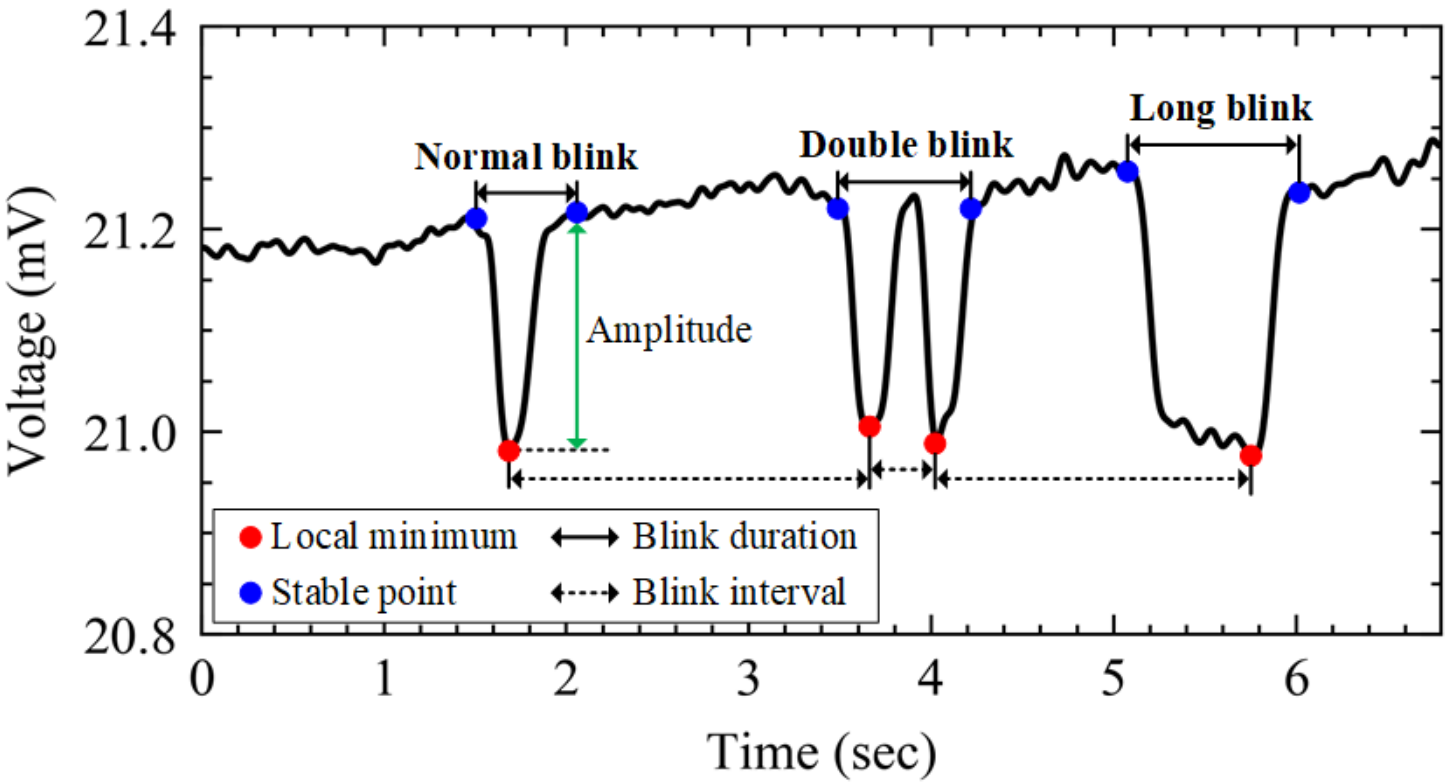

Figure 3

The typical EEG waveforms of normal blink, double blink and long blink.

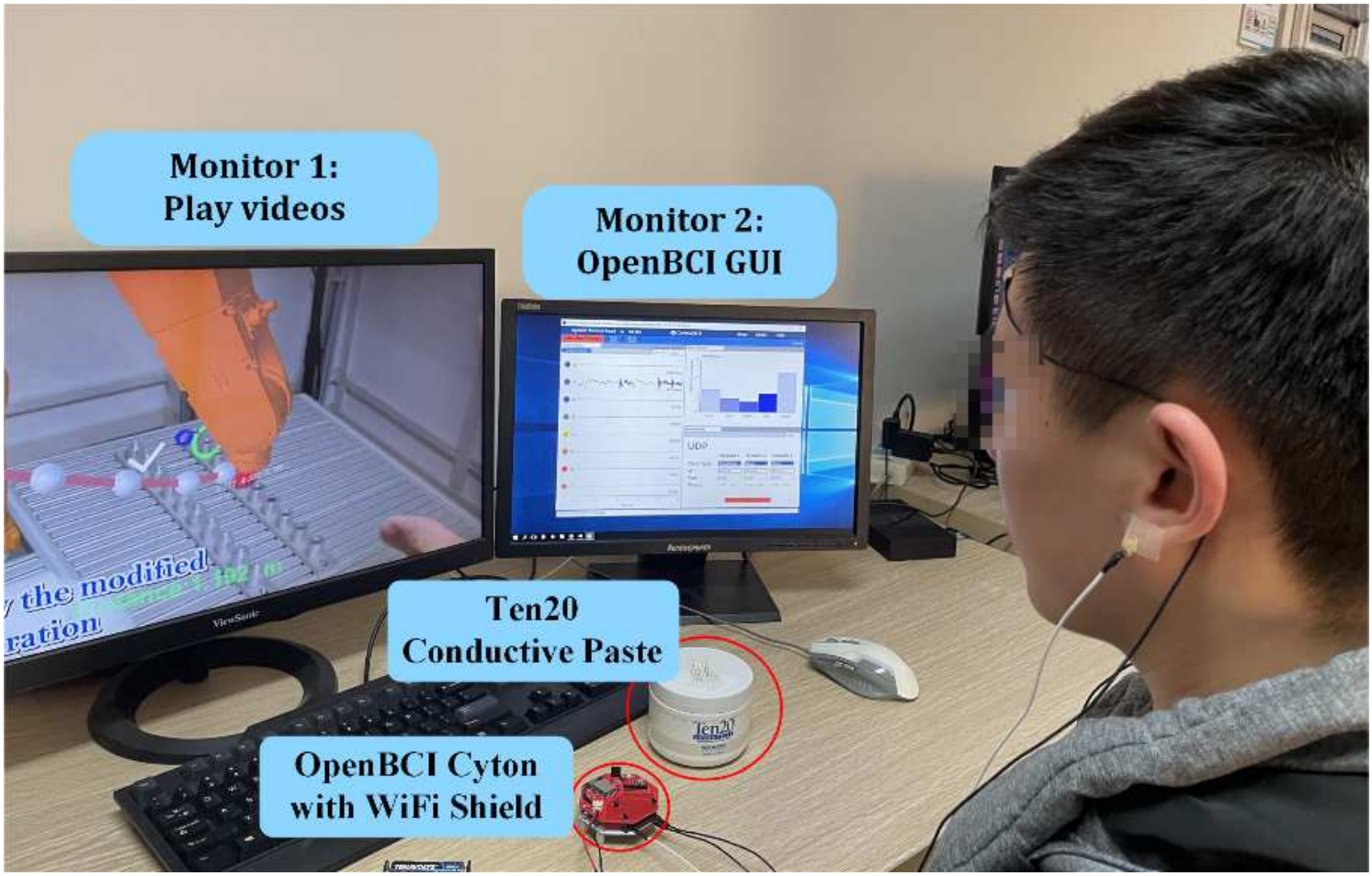

Figure 4

The setup of EEG data collection.

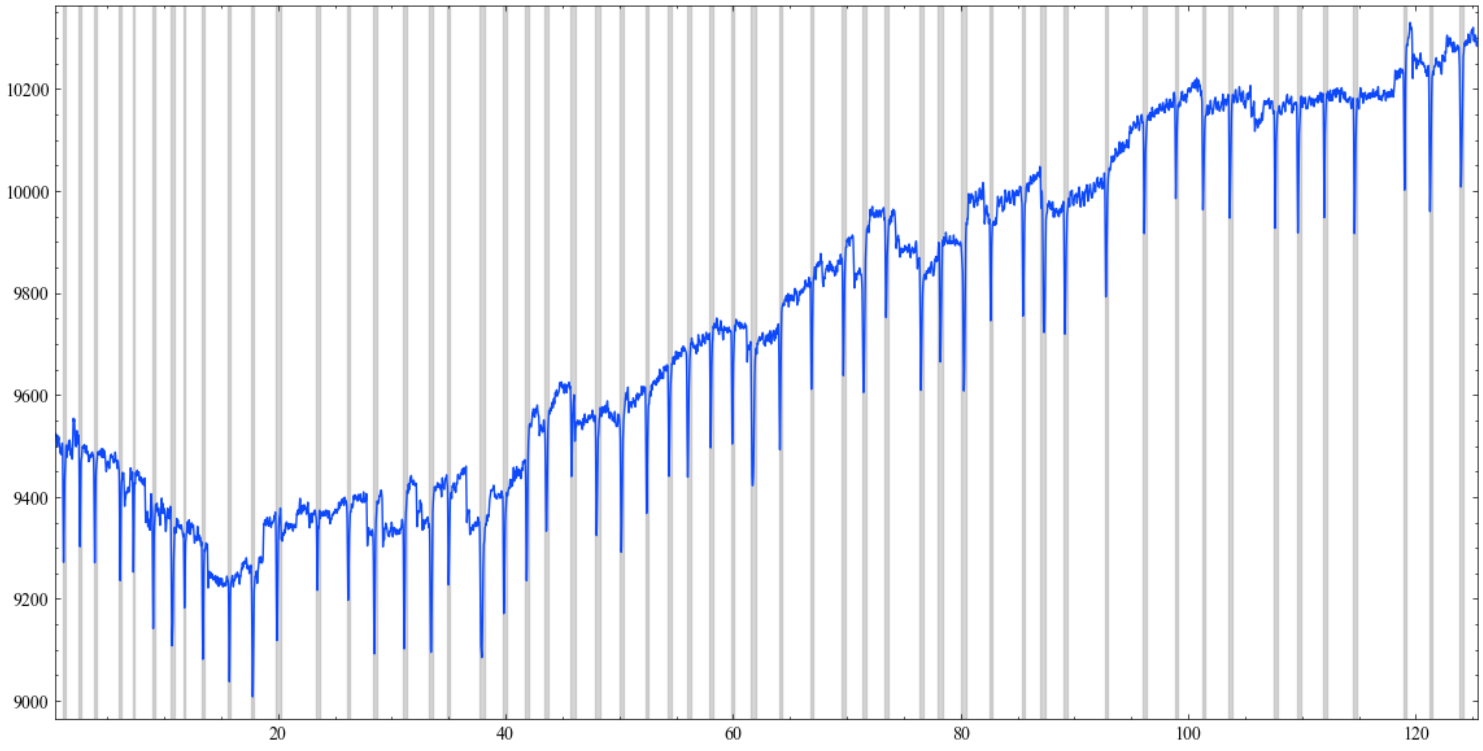

Figure 5

The result of normal blinks detection of the subject 1. Gray shades indicate all detected eye blinks and their widths represent the duration of each blink.

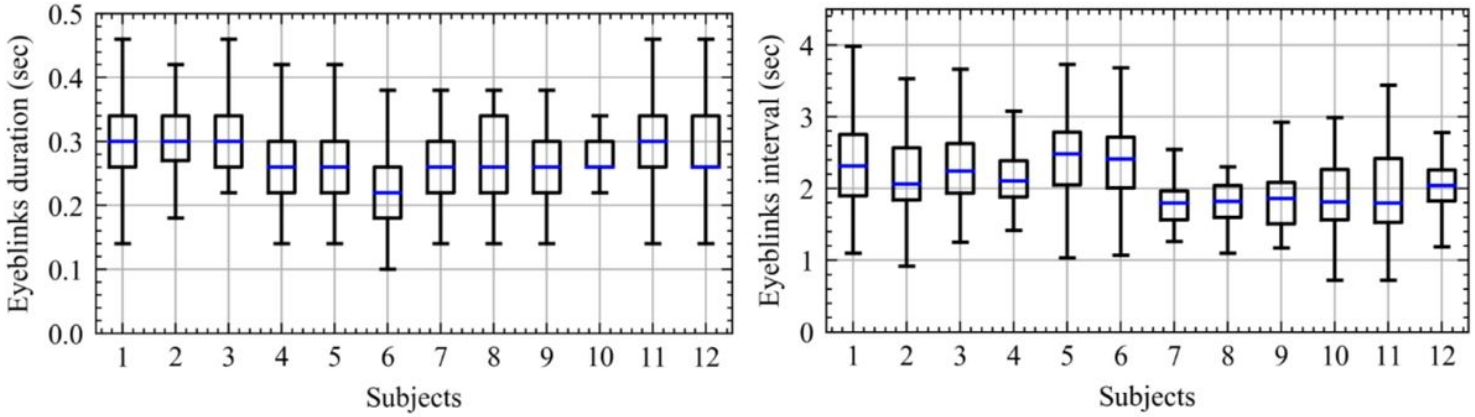

Figure 6

The duration (left) and interval (right) of the normal eye blinks.

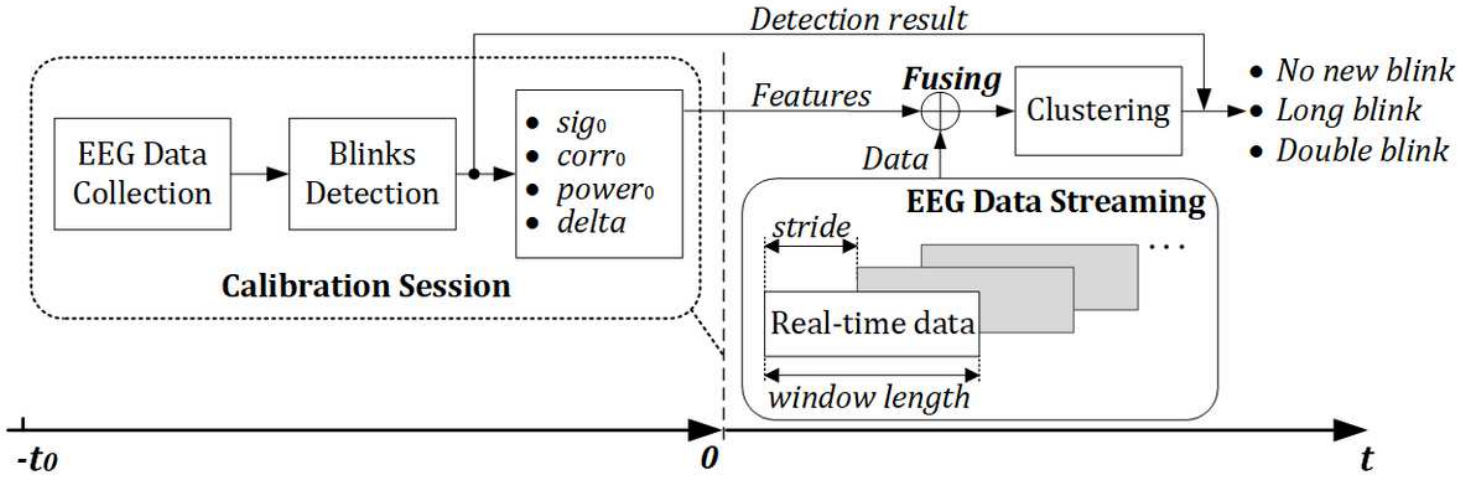

Figure 7

Multiple voluntary eye blinks online detection approach.

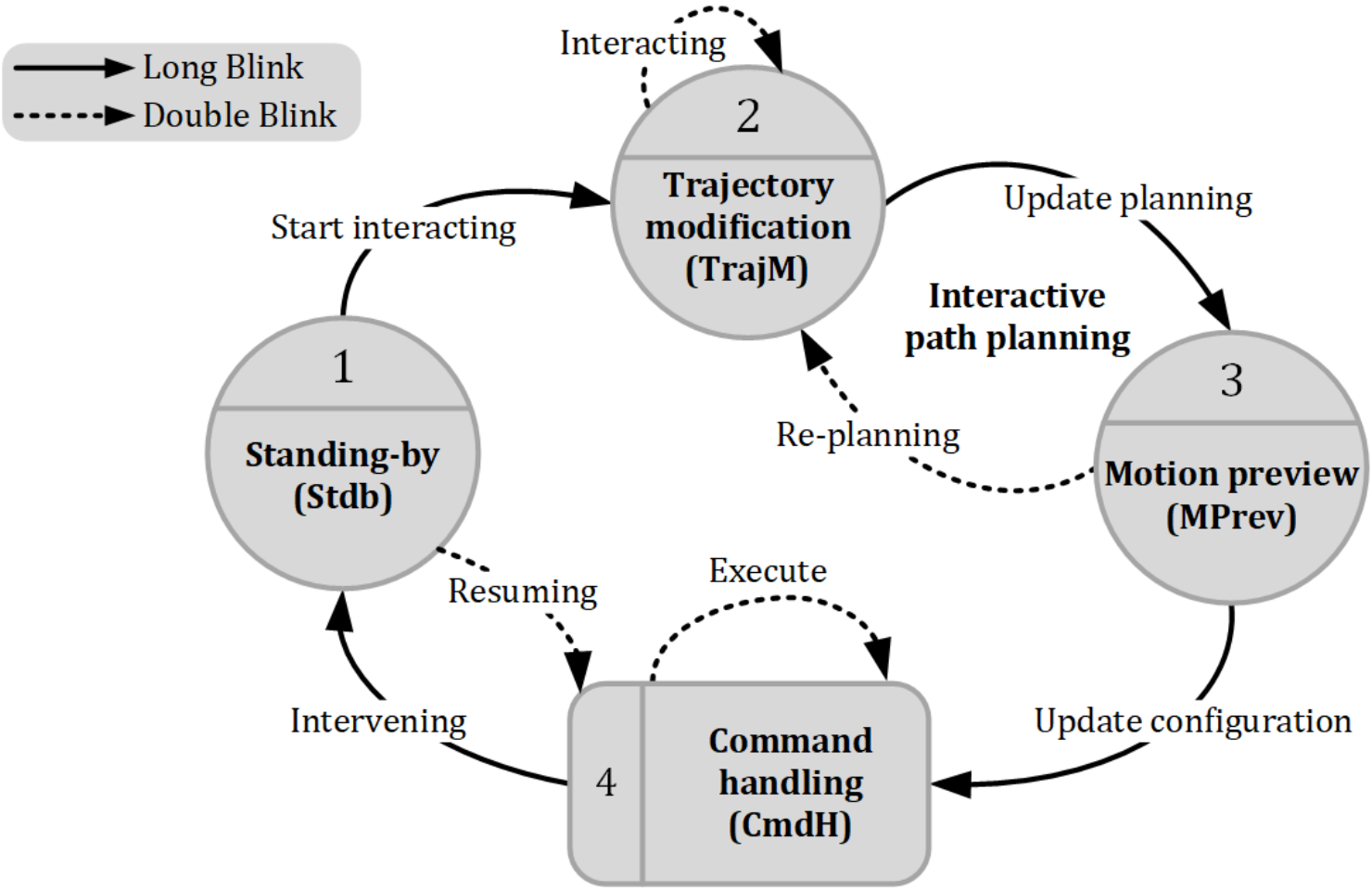

Figure 8

The schematic diagram of the control logic.

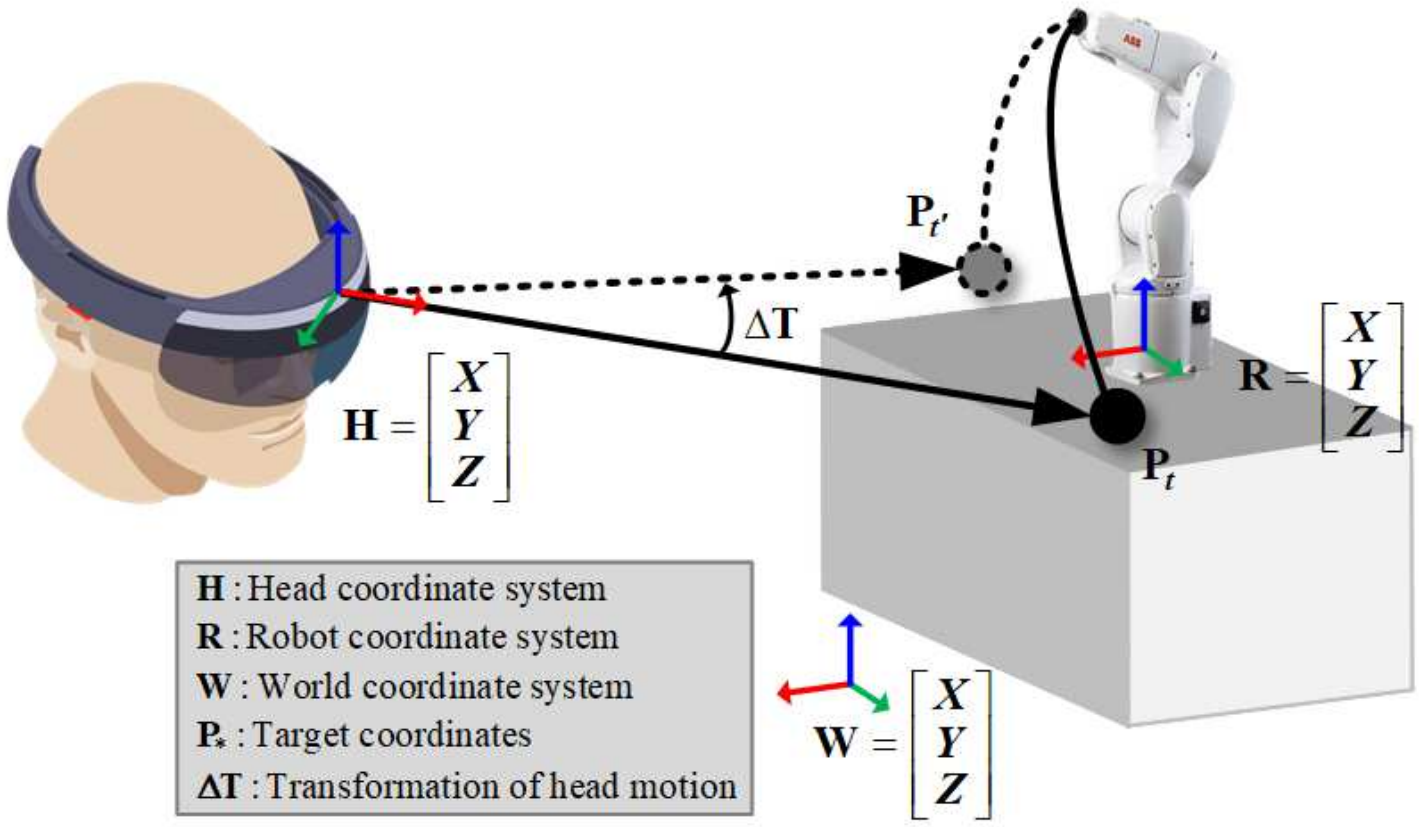

Figure 9

The coordinate systems of the AR headset, robot and HRC workplace.

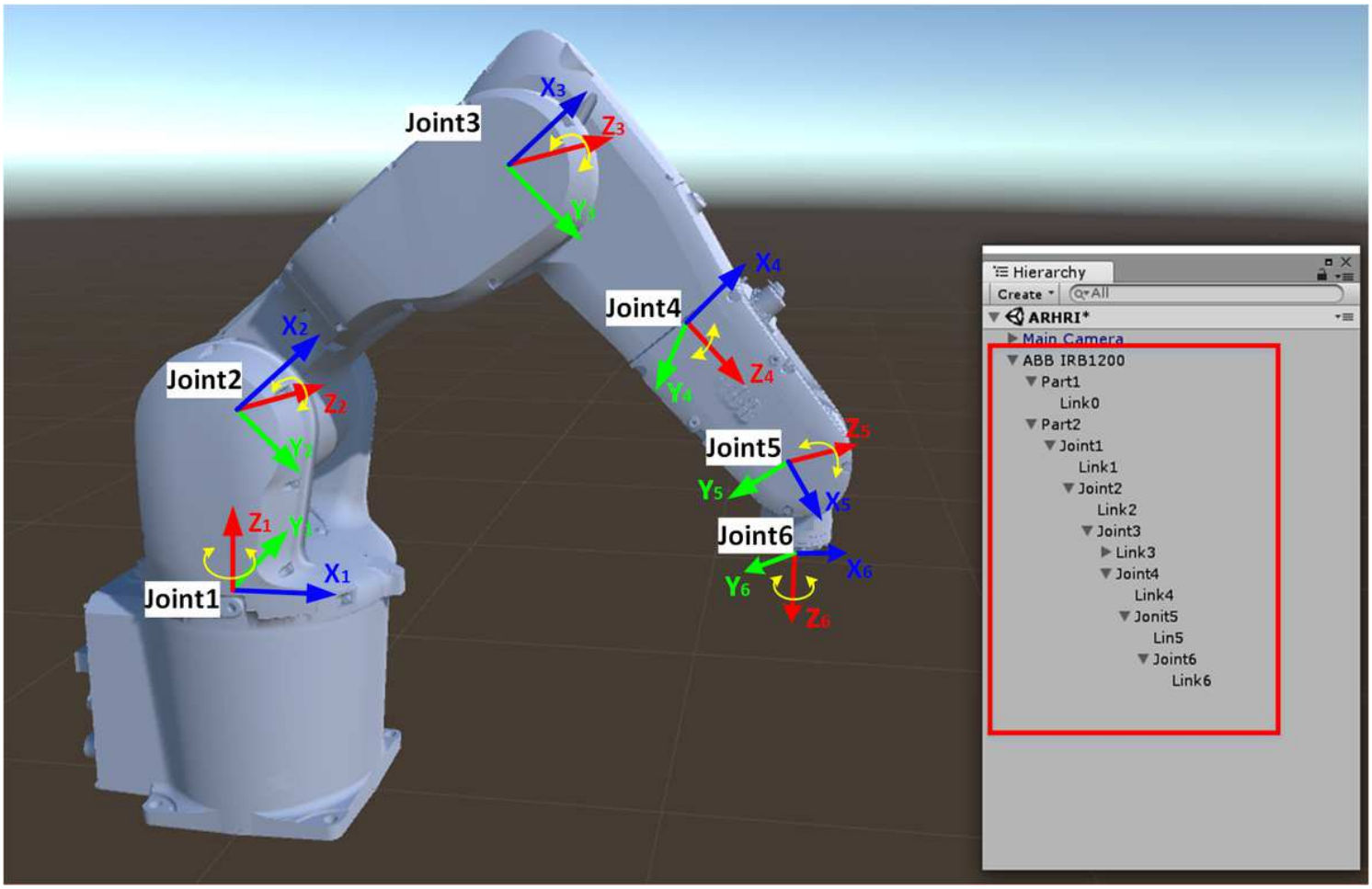

Figure 10

The virtual model and the hierarchy structure of the robot.

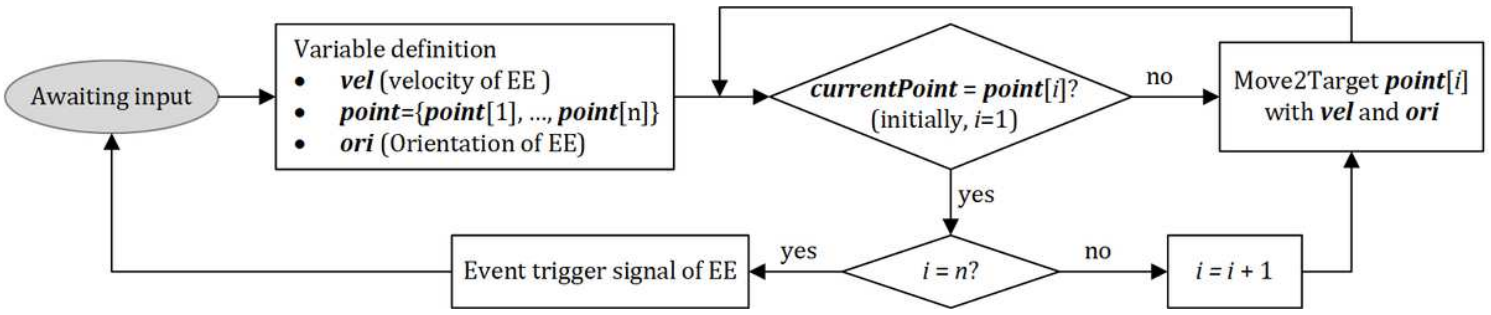

Figure 11

The structure of the robot program.

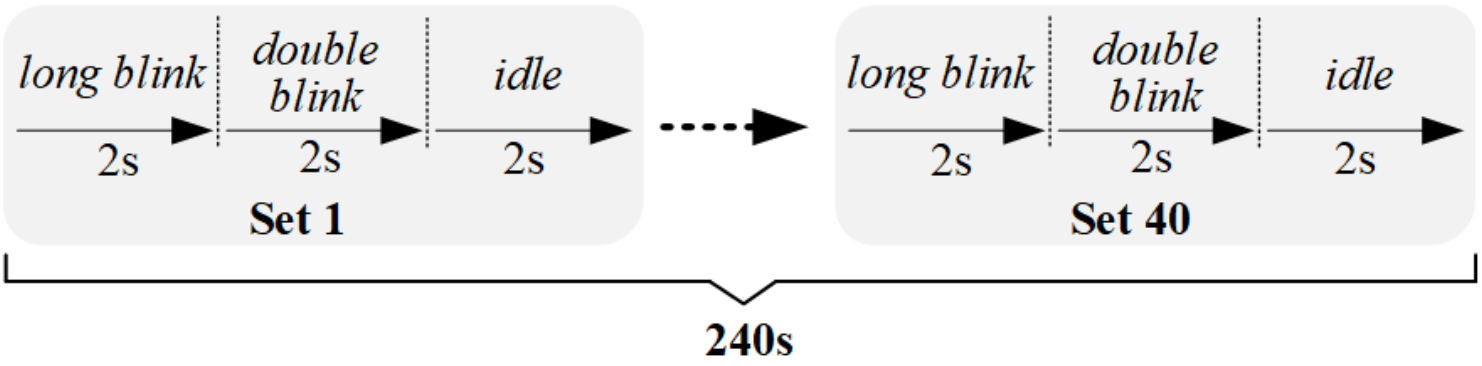

Figure 12

The protocol of EEG data collection with visual clue for each subject.

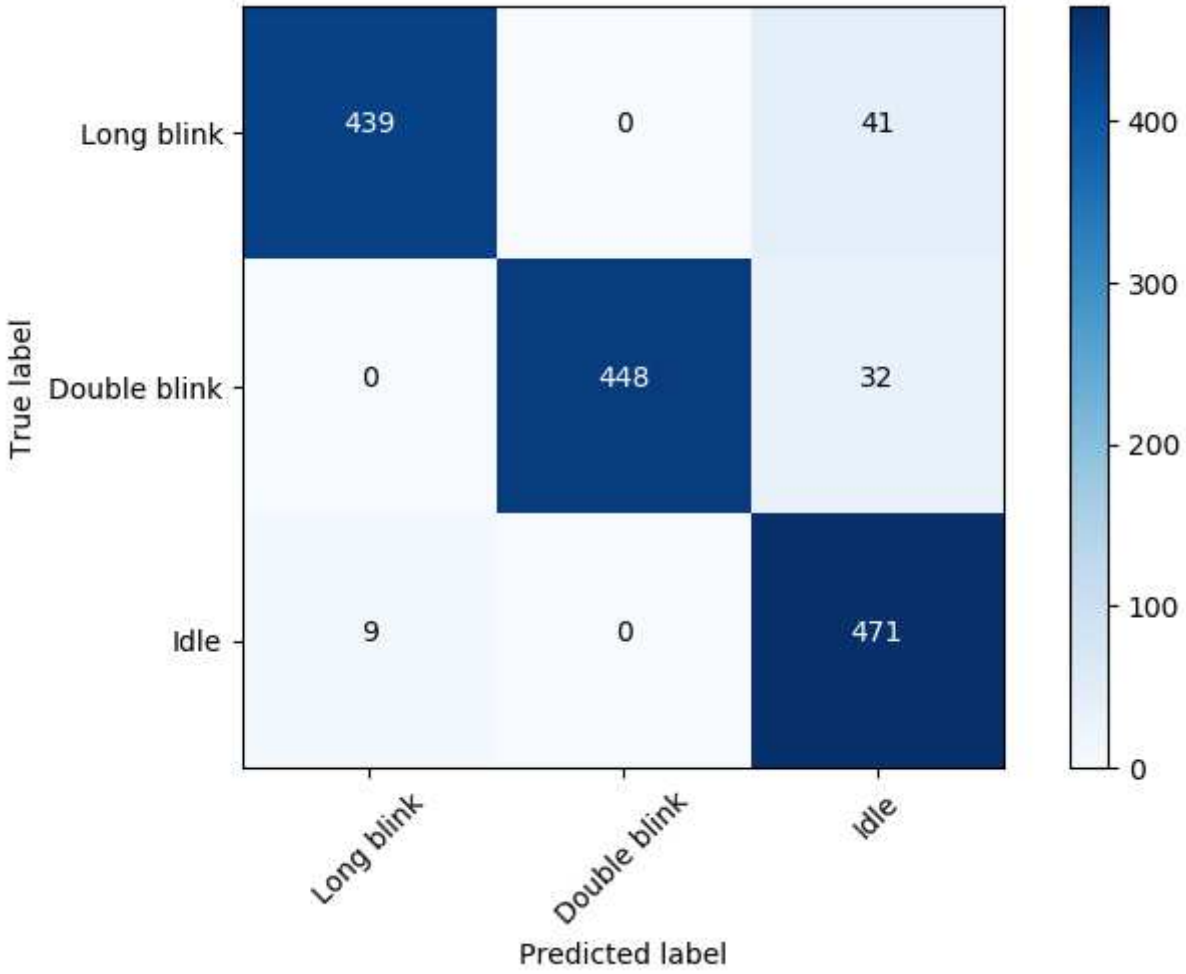

Figure 13

Confusion matrix of multiple voluntary detection for all subjects.

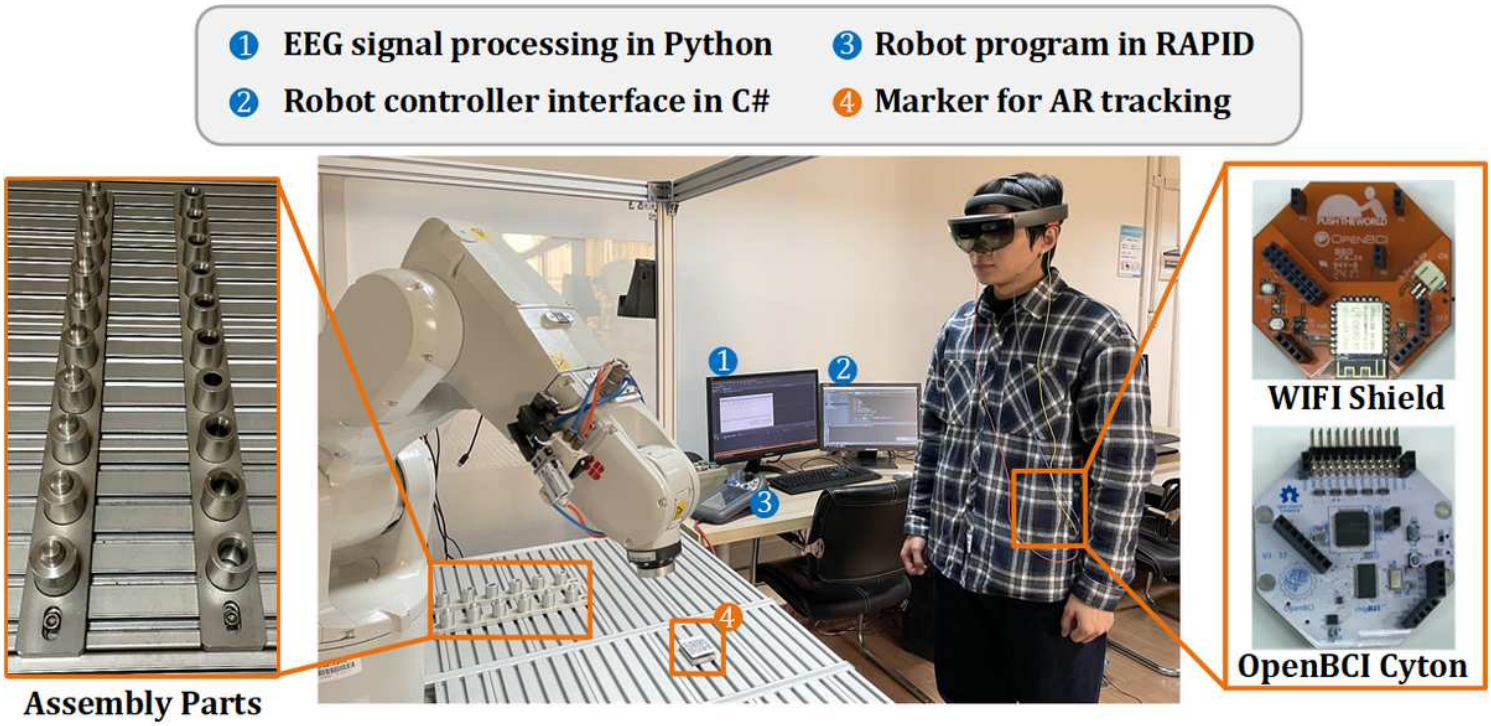

Figure 14

Case study setup.

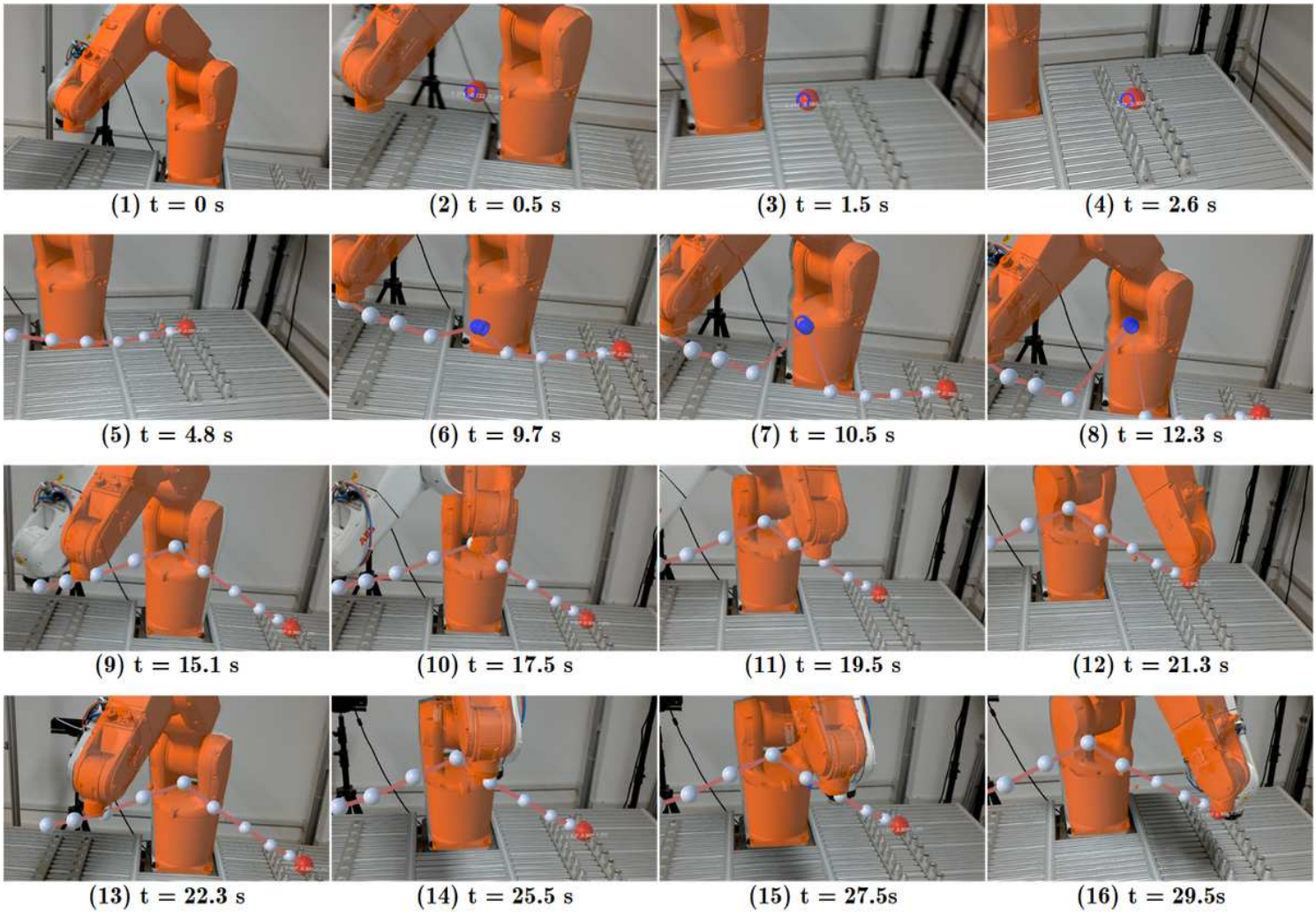

Figure 15

The snapshots of a user's point of view from HoloLens.

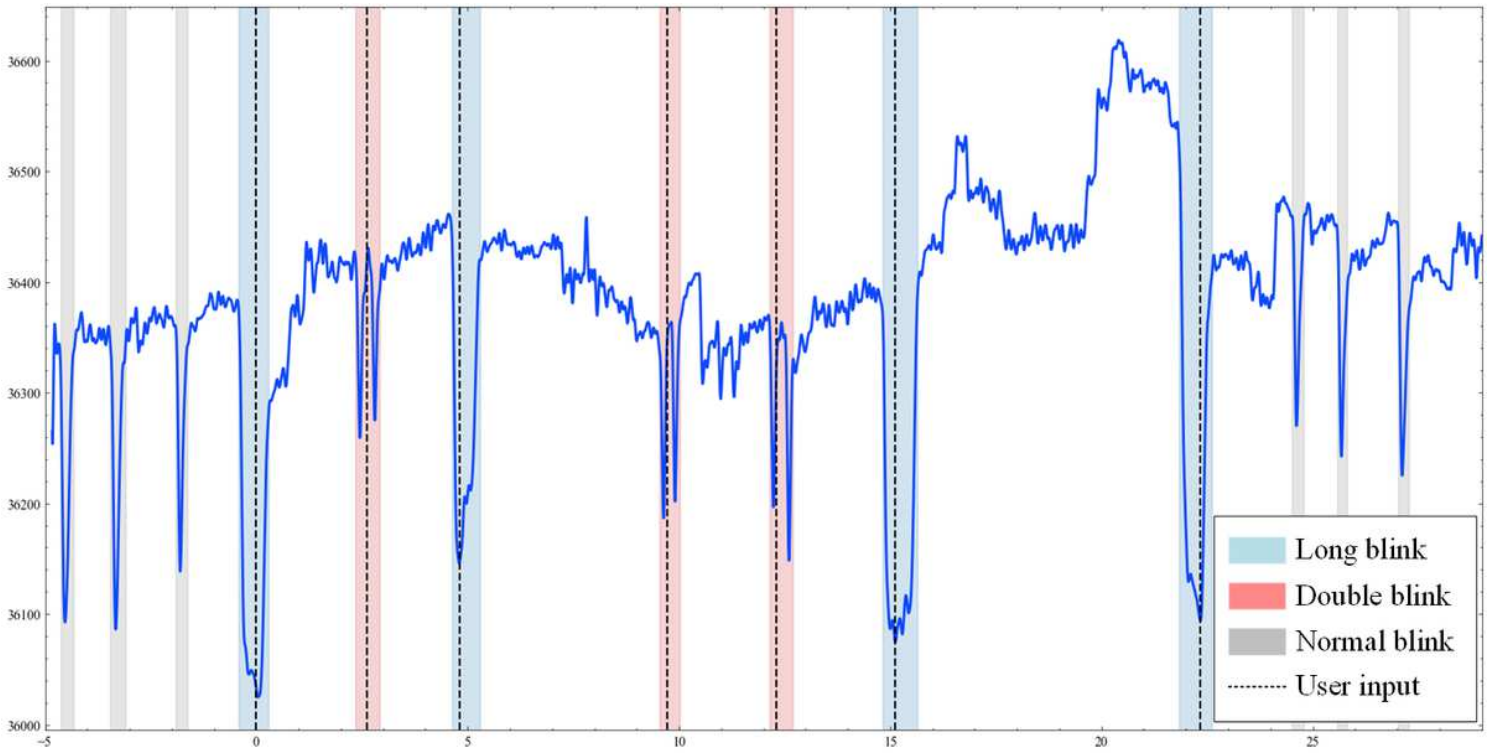

Figure 16

The recording EEG data and the result of eyeblink detection.

(1) HG-based input (2) Target specification (3) Trajectory modification (4) AR UIs

Figure 17

Examples of the user input in [8].

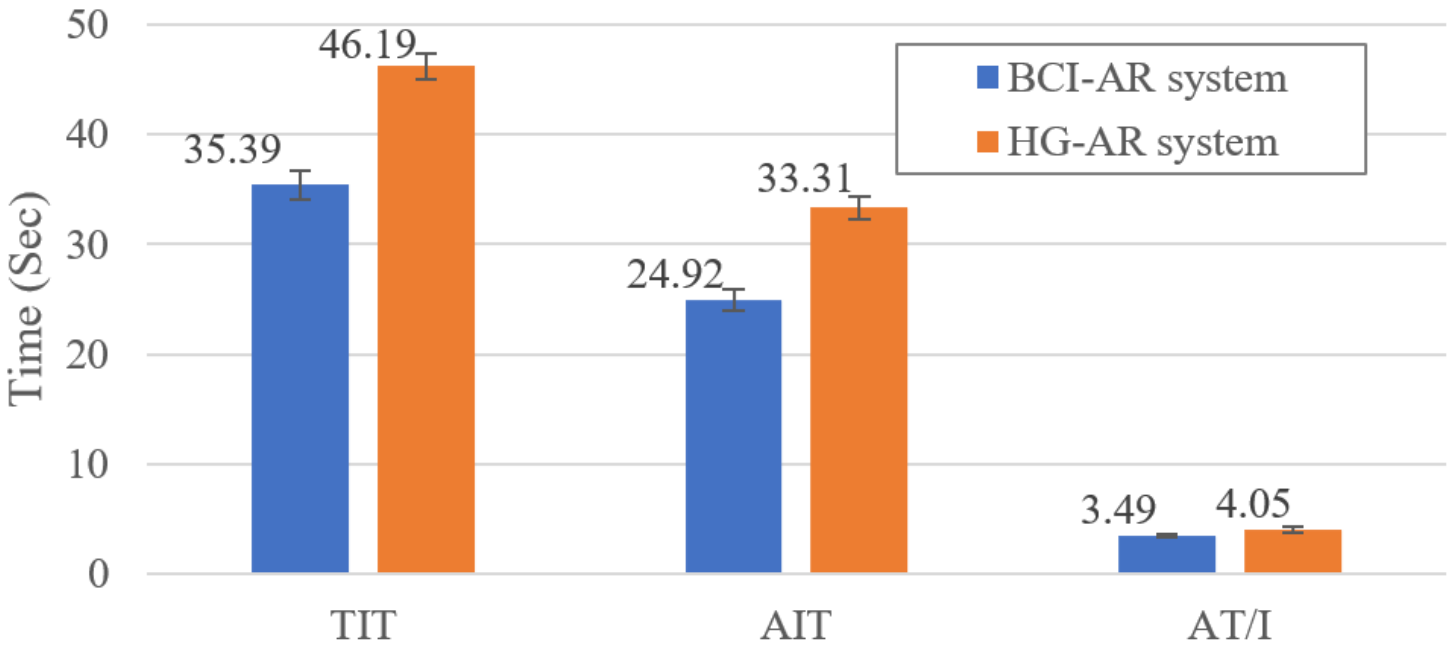

Figure 18

Performance comparison between the BCI-AR system and the HG-AR system. TIT: total interaction time; AIT: accumulative input time; AT/I: the average time per input. The numbers indicate the average values, and the error bars indicate the standard errors of mean (SEM).