Applied Soft Computing Journal 159 (2024) 111648

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Development of real-time brain-computer interface control system for robot

Yang An [a], Johnny Wong [b], Sai Ho Ling [c,*]

[a] College of Artificial Intelligence and Big Data for Medical Science, Shandong First Medical University & Shandong Academy of Medical Sciences, Jinan 250117, China
[b] School of Building Environment, Faculty of Design, Architecture and Building, University of Technology Sydney, Ultimo, NSW 2007, Australia
[c] School of Electrical and Data Engineering, Faculty of Engineering and IT, University of Technology Sydney, Ultimo, NSW 2007, Australia

HIGHLIGHTS

- A hybrid BCI real-time control system is developed to process BCI commands and drive an intelligent robot in real time.
- A actor-critic based decision-making model is introduced to mitigate unconscious brain activities and minimize action errors.
- The performance of the proposed BCI real-time control system is better than that of the state-of-the-art BCI systems.

ARTICLE INFO

Keywords:
Brain computer interface
Electroencephalogram
EEG robot

ABSTRACT

Electroencephalogram (EEG)-based brain-computer interfaces (BCI) have been considered a prevailing non-invasive method for collecting human biomedical signals by attaching electrodes to the scalp. However, it is difficult to detect and use these signals to control an online BCI robot in a real environment owing to environmental noise. In this study, a novel state recognition model is proposed to determine and improve EEG signal states. First, a Long Short-Term Memory Convolutional Neural Network (LSTM-CNN) was designed to extract EEG features along the time sequence. During this process, errors caused by the randomness of the mind or external environmental factors may be generated. Thus, an actor-critic based decision-making model was proposed to correct these errors. The model consists of two networks that can be used to predict the final signal state based on both the current signal state probability and past signal state probabilities. Subsequently, a hybrid BCI real-time control system application is proposed to control a BCI robot. The Unicorn Hybrid Black EEG device was used to acquire brain signals. A data transmission system was constructed using OpenViBE to transfer data. An EEG classification system was built to classify the BCI commands. In this experiment, EEG data from five subjects were collected to train and test the performance and reliability of the proposed control system. The system records the time spent by the robot and the moving distance. Experimental results were provided to demonstrate the feasibility of the real-time control system. Compared to similar BCI studies, the proposed hybrid BCI real-time control system can accurately classify seven BCI commands in a more reliable and precise manner. Overall, the offline testing accuracy was 87.20%. When we apply the proposed system to control a BCI robot in a real environment, the average online control accuracy is 93.12%, and the mean information transmission rate is 67.07 bits/min, which is better than those of some state-of-the-art control systems. This shows that the proposed hybrid BCI real-time control system demonstrated higher reliability, which can be used in practical BCI control applications.

1. Introduction

1.1. Background

The number of people with disabilities in the world exceeds 1.5 billion due to stroke, car accidents, work accidents, and so on [1]. With the need to enhance the quality of life and promote mobility for disability, along with advancements in neuroscientific technologies, the development of auto-controlled wheelchairs based on brain–computer interfaces (BCI) has attracted growing attention. Brain control is an ideal method because it helps control machines anywhere and anytime by imagining mental activities. As a result, BCI has become an important

* Corresponding author.
E-mail address: Steve.Ling@uts.edu.au (S.H. Ling).

https://doi.org/10.1016/j.asoc.2024.111648
Received 28 February 2023; Received in revised form 13 December 2023; Accepted 12 April 2024
Available online 21 April 2024

technology for helping people with disabilities.

Electroencephalography (EEG) and electromyography (EMG) are two types of bio-signals that can be used for robot control. Electroencephalography (EEG) is one of the main non-implanted BCIs techniques. EEG measures the electrical activity of the brain, which can be collected from a wearable EEG device [2], whereas EMG measures the electrical activity of muscles [3]. These signals can be used to control robots, allowing for more intuitive and natural interaction with machines. Compared with implantable BCI, EEG wearable devices are ethical and less invasive [4]. The EEG-based control of robots has been extensively researched in recent years. By measuring the electrical signals produced by the brain, researchers can decode the intended movement of a person and use them to control a robot [5]. This technology has the potential to improve the quality of life of people with disabilities by allowing them to control robotic devices based on their thoughts. EMG-based control of robots is another promising area of research. By measuring the electrical signals produced by muscles, researchers can extract the features of the signal and use them to control a robot [6]. This technology has applications in fields such as rehabilitation, in which patients can use their muscle signals to control robotic devices and improve their motor skills.

In recent years, there have been remarkable developments in EEG-EMG-based BCI applications, such as brain-muscle-controlled wheelchairs and EEG-EMG-based auxiliary machines [7]. With advancements in artificial intelligence and biosignal classification and analysis methods, it is possible to develop more related applications that can help disabled people complete simple daily tasks [8]. For example, implementing robot-assisted rehabilitation can serve as a pragmatic and effective solution for enhancing the well-being and overall quality of life of patients and families engaged with Alzheimer's disease (AD) [9]. In addition to helping disabled people, some BCI-related games can be developed as an exergame to increase people's activity or enhance children's brain imagination abilities [10].

1.2. Literature review

1.2.1. Compare offline and online EEG classification methods

In recent years, there has been a surge in interest in the development of offline EEG classification methods. For example, [11] tested the public dataset, BCI4–2a, to build an offline classification model. They used Joint Approximate Diagonalization (JAD) to extend the traditional Common Spatial Pattern (CSP) into a multiclass CSP. This method can reduce the uncertainties caused by artifacts. To avoid EEG information loss, they also proposed a self-regulated supervised Gaussian fuzzy adaptive system as a classifier. The learning rate can also be automatically adjusted using a coefficient. [12] used offline methods to train a long short-term memory network (LSTM), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) and tested the model on the public dataset BCI competition4–2b. In addition, they collected their own data and increased the amount of training data using an augmentation method. When they controlled the robotic arm in real time, they set a threshold for the classifier that determined the final command based on the action probability. Although they can achieve real-time classification, their device transfer rate is not very high and they have a delay of 1.4–2.55 seconds, so their method cannot achieve efficient real-time control.

Owing to the low signal-to-noise ratio of EEG signals, it is difficult to accurately recognize EEG signals, particularly for motor imagery. [13] presented a winning solution to the supervised motor imagery task in the BCI Controlled Robot contest in World Robot Contest 2021. The authors proposed a solution that included data augmentation, pre-processing, feature extraction, and model training. The model is based on EEGNet, which is a popular convolutional neural network model for classifying EEG data. Despite the lack of stability, this solution was the most successful for this task. The channels closest to the vertex are the most helpful for feature extraction. The authors concluded that this solution is suitable for supervised motor imagery tasks not only in this competition, but also in future scenarios.

Over the past few years, substantial progress has been made in the development of EEG-based online classification systems to construct real-time BCI control applications. EEG-based real-time control applications were developed in [14] and [15]. [14] used four channels and employed a combined CNN and LSTM as the classifier to classify two tasks, namely, motor imagery of making a fist and motor imagery of opening a fist. The acquisition time for each experiment was 3 s. They used a hard processor system that included efficient data processing hardware and memory. The transfer rate is high; therefore, their method can limit the delay time to within ten milliseconds. Thus, their systems can achieve efficient real-time classification. [15] constructed a signal processing model and CSP feature extraction model using the OpenViBE software. the spatially transformed signal energy was then calculated, and the average value of the signal was input into the LDA for classification. They also used a Virtual Reality Peripheral Network (VRPN) that can efficiently transmit data and does not require extra support from other software. Compared with [16] and [17], the data transmission technique developed by [14] and [15] is more efficient.

1.2.2. Real time multi-task classification models

While the methods discussed above focus on EEG binary classification, there has also been a growing number of studies focusing on multiclass tasks in real time. For example, [18] and [19] implemented three classes of motor imagery-based control methods. [18] used the Filter Bank Common Spatial Pattern (FBCSP) as the feature extraction method, mutual information-based best individual feature (MIBIF) as the feature selection method, and support vector machine (SVM) as the classifier to control the lower-limb exoskeleton. They achieved real-time control by performing classification every 0.5 seconds, and the data size for each classification was 2 s. Because motor imagery requires a certain amount of imaging time, this sliding-window-based segmentation method is more suitable for analyzing EEG signals.

In addition to the above traditional classification models, the convolutional neural network can accurately extract more motor information from EEG signals. [20] proposed a real-time decoding system for motor imagery movements based on a convolutional network. The EEG signals were first converted into spectrogram images using three secondary tables that captured the time-frequency features of the signals. Subsequently, they used a pre-trained GoogLeNet model to classify the spectrogram images into four motor imagery tasks: up, down, left, and right movements. The classified motor imagery tasks were then used to control the movement of the robot manipulator in real-time. Additionally, [21] presented a novel sequential learning model based on Graph Neural Networks (GNN) for EEG-EMG-based stroke rehabilitation BCI. The model processes a sequence of graph-structured data derived from EEG and EMG signals, which capture both the motor intention and movement execution features of patients. The model divides the movement data into sub-actions and predicts them separately, generating sequential motor encoding that reflects the temporal dynamics of the movements. The model also uses time-based ensemble learning to improve the prediction accuracy and provide execution quality scores for each movement.

1.2.3. Development of BCI real-time control applications

The past few years have also produced a significant body of research on other real-time control applications. Some studies have highlighted the potential of BCIs in controlling robots and assistive devices. [22] presented a brain-computer interface (BCI) system that allows a human operator to control end effectors using only mental commands. The authors proposed a novel approach to BCI design by modeling BCIs as a communication system and deploying a human-implementable interaction algorithm for the non-invasive control of a high-complexity robot swarm. They demonstrated the feasibility of this approach through a large-scale user study, which involved a user test of a full BCI system on robot swarms and simulations. However, a single use of mental

commands may lead to the instability of the robotic system. Thus, [23] proposed a continuous shared control strategy that combines continuous BCI and autonomous navigation for mobile robot systems. This can improve the stability of the control systems. The weight of the shared control was designed to dynamically adjust the fusion of the continuous BCI control and autonomous navigation. The results of the online shared control experiments showed that all subjects could complete navigation tasks in an unknown corridor with continuous shared control.

In addition, some researchers wanted to prevent subjects from issuing incorrect commands before controlling the robot. [24] investigated the application of an error-related potential (ErrP)-based BCI paradigm to control robot movements with implicit commands. The authors proposed a novel robotic design for an ErrP-based BCI system that allows humans to continuously evaluate a robot's intentions and intervene earlier, if necessary, before the robot commits an error. The high classification accuracy demonstrated that the proposed ErrP-based BCI system is feasible for human–robot intention communication before the robot commits an error. In addition, owing to the instability of EEG signals, [25] presented a novel BCI system for controlling an assistive robot with user eye artifacts. The authors considered eye artifacts that contaminate electroencephalogram (EEG) signals as a valuable source of information because of their high signal-to-noise ratio and intentional generation. The proposed methodology detects eye artifacts from EEG signals using characteristic shapes that occur during events.

Some studies have focused on the application of BCI technologies to motor rehabilitation or assistance. Both [26] and [27] aimed to provide a natural and reliable method of interacting with an exoskeleton. [26] presented a novel multimodal human-machine interface system (mHMI) that uses EOG, EEG, and EMG signals to control a soft robot hand in real time, while [27] presented a real-time control system for a lower-limb exoskeleton that uses EEG and EMG signals. [26] showed the potential of BCI technology for motor rehabilitation therapies that use neural plasticity to restore motor function and improve the quality of life of stroke survivors. [27] developed a real-time control system that can initiate and differentiate movements of the right and left legs with a high degree of reliability.

In addition, [28] proposed a dual-stage deep learning framework that can classify high-level motor imagery tasks related to natural hand grasp motions from EEG signals. The framework consists of two stages: feature extraction and classification. The feature extraction stage uses a CNN to extract spatial and temporal features from EEG signals, and an RNN to capture the sequential dynamics of motor imagery tasks. The classification stage uses an attention mechanism to weigh the importance of the different features. The framework also uses an EMG-based learning strategy that leverages the EMG signals of the healthy arm as a guidance signal during model training to improve the accuracy and reliability of EEG-based inference.

Similarly, [29] presented a hybrid EEG-EMG-based BCI system for real-time robotic-arm control. The system uses non-invasive EEG and EMG biosensors to design a wireless hybrid BCI system that can detect the motor intention and wrist muscle movements of the healthy arm of subjects with above-hand amputees. The system can control the robotic arm within three Degrees of Freedom (DOF), which corresponds to the movement of the shoulder, elbow, and wrist joints. The system was experimentally tested on four subjects with upper limb amputees after a training period of one day and achieved an accuracy of 70–90%.

1.3. Challenges and contributions

Despite these research efforts, there is a dearth of studies aimed at achieving a high-performance BCI control system. While existing literature and studies have developed mechanisms to control robots using EEG signals, accurate and seamless classification and execution of multiple EEG commands are lacking. Many existing BCI control approaches can only recognize less than four BCI commands. In addition, most methods can be used only for offline control. Thus, these systems cannot be applied to any real environment. Additionally, when people perform brain tasks, there are energy changes; however, these changes are weak and difficult to detect. Performing brain tasks is a process that cannot be immediately completed. During this process, the brain states keep changing, and changes are gradually generated. For example, when a subject performs motor imagery tasks, the brain signals change from a resting state to a motor imagery state, and the energy in the related frequency bands decreases gradually. The control method proposed in most papers is direct control, which means that the result of each recognition is directly used as the final control command. However, in practical applications, the users may be affected by the external environment. Thus, they may generate a control command unconsciously, which is not the desired result for the user. Therefore, using this control approach, the output commands do not follow the user's expectations. As a result, detecting the signal state and predicting the current robotic action are difficult tasks.

Therefore, a hybrid BCI real-time control system estimation technique was proposed in this study to address the problems discussed above. This study has three main motivations. First, we established a complete brain-controlled robot system that enables the robot to obtain real-time EEG commands and operate stably in unfamiliar environments. In addition, this system introduces actors and discriminators to make judgments on the current robot state based on the output commands of the robot's current and previous times, thereby optimizing the robot's output actions to improve the robot's lagging situation. Second, we embedded our proposed feature extraction and classification algorithm into this real-time control system, enabling the robot to accurately recognize multiple biological signals, which significantly improves the real-time recognition accuracy and information transmission rate of the robot. Third, in this experiment, we combined three commonly used physiological signals: EEG, EMG, and EOG signals. Compared with using a single physiological signal, this combined approach helps improve the information transmission rate of the robot and makes its movements more flexible. The contributions of this study are twofold:

Contribution 1: The hybrid BCI real-time control system developed in this study helps process multiple BCI commands in real time, promoting higher robotic precision and more reliable control. To implement a hybrid BCI real-time control system, two components were developed in this study. First, an EEG-based dynamic classification system was proposed to address EEG multi-classification tasks. Second, a data transmission system is proposed to achieve data communication. By applying the proposed control system, the BCI robot could be controlled using seven BCI commands in real time. The performance of the proposed control system is better than that of the state-of-the-art BCI systems.

Contribution 2: The actor-critic based decision-making model proposed in this study can learn the user's control habits. The model considers not only the current signal state but also the previous signal states. By continuing to learn the changes in the user's brain states and corresponding labels, the model can update the parameters and correct the final control command. By using the proposed method, unexpected actions caused by unconscious human brain activity can be avoided. It can also reduce the errors caused by external influences. Compared with traditional classifiers, the proposed method can better predict reasonable robotic actions.

This paper is organized into five sections. A literature review of EEG classification models and EEG-based control systems is provided in the introduction section. The challenges and contributions of this study are discussed in this section. The Methodology section presents an overview of the proposed hybrid BCI real-time control system. It introduces two subsystems, including a data transmission system and an EEG-based dynamic classification system. The Experiments and Results section presents an EEG-controlled robot built using the proposed hybrid BCI real-time control system. The EEG robot will be built using well-trained models and its performance will be tested in a real environment. Important performance indicators are recorded to evaluate the performance and reliability of the robot. The benefits of the proposed decision-

making model, the different structures of the classification network, and the comparison of performance with other BCI models are explained in the discussion section. Finally, the conclusion section provides a conclusion regarding the materials covered in this study. Directions for further research are discussed.

2. Methodology

2.1. Block diagram of system

This section describes the proposed hybrid BCI real-time control system. In this study, we used a data transmission system and an EEG dynamic classification system to construct a hybrid BCI real-time control system that is used to control a BCI robot car. The block diagram of the system is shown in Fig. 1. The first step was to build a connection between the EEG acquisition device and the computer using a USB Bluetooth adapter. A computer requires device configuration that includes setting specific device ports and connection names. The data transmission system then obtains the raw data from the computer through its I/O port. After processing the signals, the online data were sent to the EEG-based dynamic classification system to classify the commands. Subsequently, the output commands were sent to the BCI robot via WebSocket. Finally, the commands were executed by the BCI robot.

2.2. Data transmission system

A data transmission system was proposed to obtain EEG signals from the hardware and send the processed data to other software or external devices. The structure of the data-transmission system is shown in Fig. 2. It connects the EEG device according to the device port and connection name and obtains data via the Lab Streaming Layer (LSL). Subsequently, the online data were saved in the acquisition server. If other software or devices want to access data, they must send a request to the server. The online data are then processed and sent by the data processing and reception model.

OpenViBE is free open-source software for real-time neuroscience. It can be used to acquire, filter, process, classify, and visualize brain signals in real-time. The package includes a designer tool to create and run custom applications, along with several preconfigured and demo programs that are ready for use. A few functional boxes were designed. The EEG data processing and reception models are shown in Fig. 3. An acquisition client box was used to receive EEG data. A channel selector box was used to select effective channels. Bandpass and band stop boxes were used to filter EEG signals. The GDF file-writer box was used to save the EEG signals in the GDF format.

A Lua stimulator box was used to guide the subjects to take corresponding actions. There were 11 stimulation labels: 1) left-hand motor imagery, 2) right-hand motor imagery, 3) eyeball move to the left, 4) eyeball moves to the right, 5) eyeball moves down, 6) bite teeth, 7) experiment start, 8) experiment end, 9) break, 10) prepare to perform the task, and 11) start a trial. The number and time of the task prompt labels were preset. All the prompt labels were randomly presented. The subjects performed the corresponding tasks according to prompts.

In this system, the lab streaming layer and OpenViBE acquisition server are connected to ensure the acquisition of high-quality signals in the signal-detection scene. The lab streaming layer is a system for the unified collection of time series measurements in research experiments. It handles networking, time synchronization, and real-time access, as well as centralized collection, viewing, and disk recording of the data. After the signal was processed by the OpenViBE designer, the data were exported by the LSL data export box.

The EEG device transmitted the collected EEG data to a laptop through a Bluetooth adapter. First, it matches the device port and name on the laptop and creates a lab-streaming layer. The EEG data were sent to the OpenViBE software through the lab streaming layer. OpenViBE received data through an acquisition server. Subsequently, the data processing and reception model was established in OpenViBE designer, including the data visualization module and MATLAB connection module. The visualization module can directly plot the EEG signals in real time. The MATLAB connection module was used to send the processed data to the MATLAB software. An EEG-based dynamic classification system was established in MATLAB to analyze and classify EEG signals.

2.3. EEG based dynamic classification system

The EEG-based dynamic classification system is the main component of the hybrid BCI control system, as shown in Fig. 4. EEG data were obtained from the data transmission system. The proposed classification system was used to classify BCI commands, which were subsequently used to control the BCI robot. The proposed system recognizes seven different commands. The system includes five models: pre-processing, denoising, EEG state recognition, motor imagery classification, and eyeball movement classification models. In Fig. 4, when the EEG signal arrives, it is pre-processed and denoised by the pre-processing and denoising models. Some noise and high-frequency components were removed by these two models. The processed signal is then input into the EEG state recognition model to calculate the probabilities of motor imagery, eyeball movement, EMG, and rest state signals, which are further used to determine the robotic action states. Subsequently, an actor-critic-based decision-making model is developed to integrate all the probabilities of the commands and determine the exact signal state. Eventually, if the signal state is motor imagery, the motor imagery classification model is used to classify the signal to obtain the final commands. If the signal state is an eyeball movement, the eyeball movement classification model is used to classify the signal and obtain

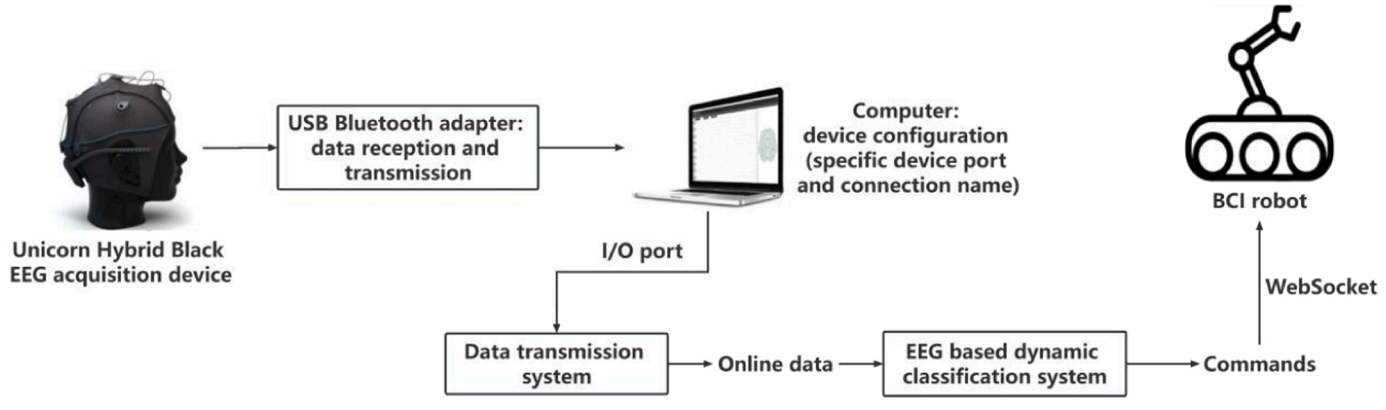

Fig. 1. Block diagram of hybrid BCI real time control system.

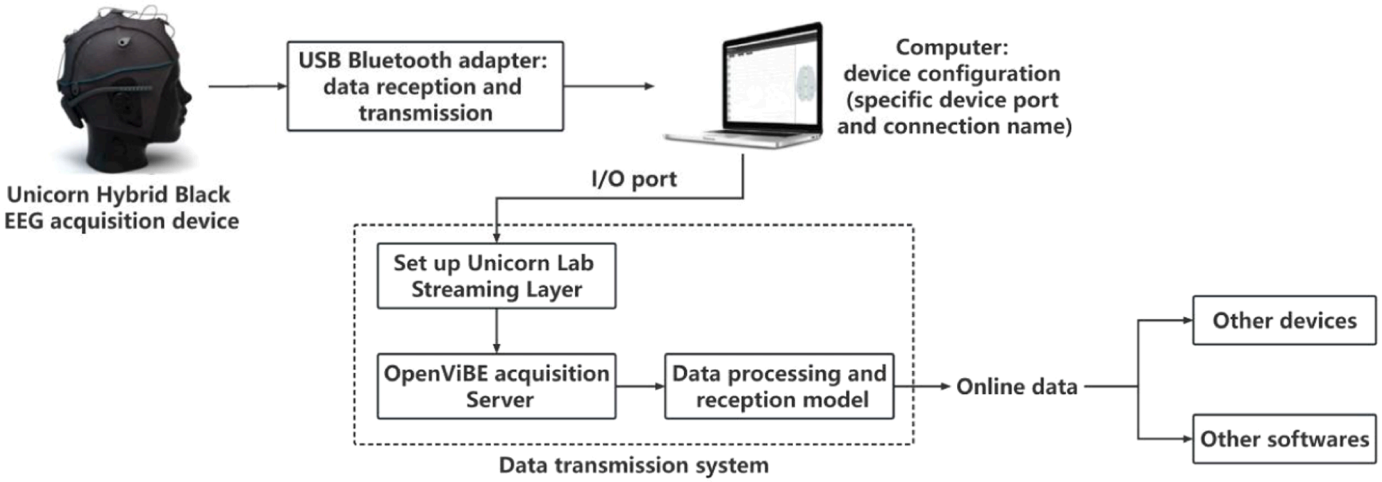

Fig. 2. Structure of data transmission system.

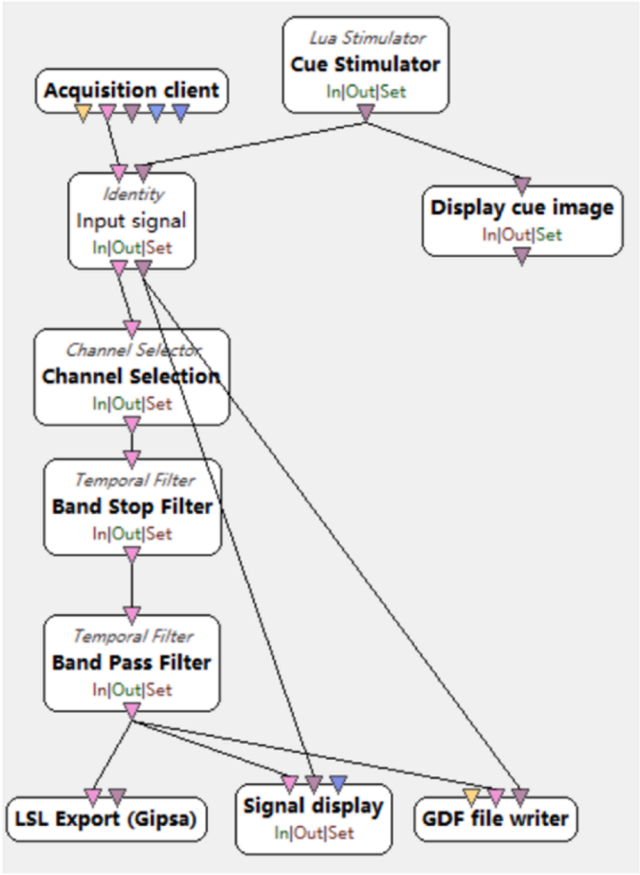

Fig. 3. EEG data processing and reception model in OpenViBE.

the final commands. If the signal state is EMG, it is directly used to control the robot. Otherwise, if the signal is in the resting state, the robot will do nothing.

2.4. Pre-processing and denoising model

In the pre-processing model, the raw EEG data are filtered by a 1–40 Hz bandpass filter because the main features of EEG exist in this frequency band. This can also remove high-frequency noise, such as 50 Hz or 60 Hz linear noise. The second step is to re-reference the raw EEG data minus the mean value of all channels. The function of this step is to correct the reference electrode to a value close to zero. Finally, bad signal segments were removed. The signal energy fluctuates significantly because the equipment may be affected by noise. We defined such data as low-quality data and removed these data by setting the energy threshold. The output of this model is a pre-processed EEG signal.

The pre-processed signal was input into the denoising model for noise removal. EEG signals can be easily affected by unexpected noises, such as eye blinking and heartbeat. These noisy signals may generate higher energies than the original EEG signal. This artifact may affect the performance of the EEG signal classification. Thus, we constructed an automatic denoising method to remove noise from pre-processed EEG signals. Details can be found in [30].

2.4.1. EEG state recognition model

2.4.1.1. Overview of model. The denoised signal is input into the EEG state recognition model, which can identify four signal states: i) rest, ii) EMG, iii) motor imagery, and iv) eyeball movement. The block diagram of this model is shown in Fig. 5. A hybrid long short-term memory convolutional neural network was introduced to extract the time-domain features of EEG signals and classify the signals along the time sequence. Suppose that the output of the EEG state-recognition model is the probability of each type of signal. Subsequently, these probabilities were used to determine the robotic actions.

2.4.1.2. Structure of network. The network structure and parameters are shown in Fig. 6 and Table 1, respectively. First, the signal is input into the network and the input signal is converted into an image through the folding sequence layer. Through three sets of convolution and pooling operations, temporal features were extracted from the signal, and the features were compressed. Subsequently, the acquired feature maps are rearranged into time sequences through the sequence unfolding layer. The output of this layer can be regarded as the features extracted from the EEG signals of each period and then rearranged according to the time sequence. Finally, the extracted compressed feature signal is input into the LSTM layer to calculate the signal state probabilities.

2.4.2. Actor-critic based decision making model

2.4.2.1. Overview of model. A block diagram of the decision-making model is shown in Fig. 7. The purpose of this model is to predict the states based on previous recognition probabilities. Recognition probabilities were input into the two models. The first is to make a decision based on maximum probability. The second method is to use an actor

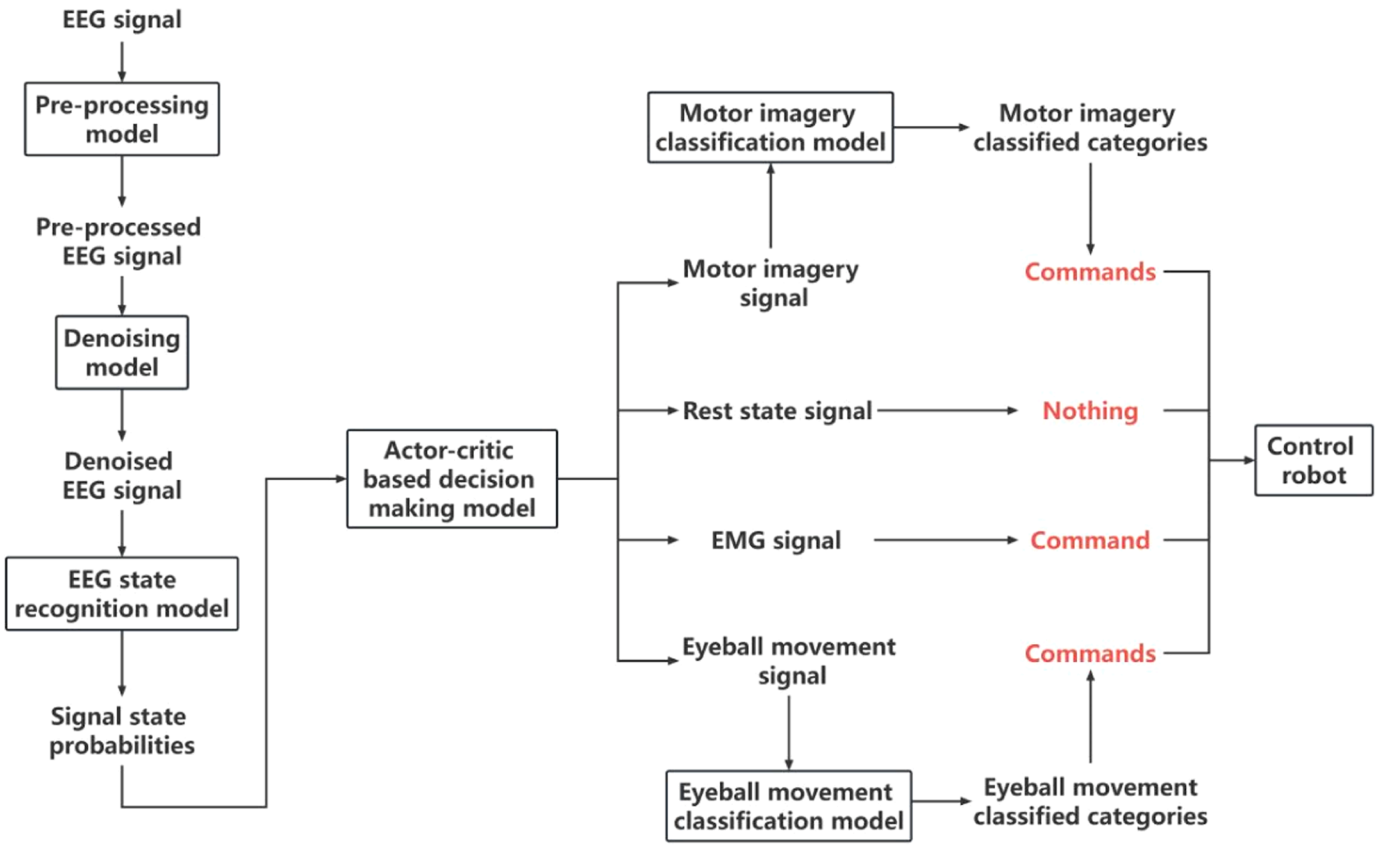

Fig. 4. Structure of EEG-based dynamic classification system.

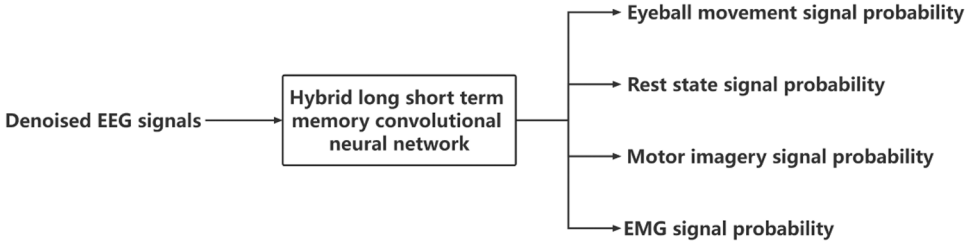

Fig. 5. Block diagram of EEG state recognition model.

network to predict the desired action. The critic network is used to evaluate whether the output from the actor network is good, responding to EEG recognition probabilities. Thus, a critic network can be used to evaluate the outputs of the two models. Finally, the action with the highest score was used as the final action.

2.4.2.2. Structure of actor and critic networks. The structure of the actor network is illustrated in Fig. 8. The input is the probability of the current signal state and past signal states. The output was the predicted action. There are three hidden layers in the network. After each fully connected layer, a ReLU layer was used as the activation function.

The structure of the critic network is shown in Fig. 9. It has two inputs. The first input is the probability of combining the current and past signal states. The number of states depends on the data transmission rate. The second input is the prediction action. The relation features among the probabilities were learned by the fully connected layer. The extracted features are then combined with the predicted action. Two additional fully connected layers were used to extract further features among the probabilities and action. Finally, an evaluation score is output.

2.4.2.3. Training actor and critic networks. The structures of the two networks need not be complicated. The key is to design the loss functions. The training process is illustrated in Fig. 10. For the critic network, our purpose was to judge whether the output of the actor was good. Therefore, we used the mean square error as the loss function. We input signal state probabilities P_s and correct actions a_g into the critic network to calculate the score and then calculate the mean square error between the score and 1. Then, we input the signal state probabilities P_s and random actions a_b into the critic network to calculate the score and then calculate the mean square error between the score and 0.

For the actor network, our goal was to predict the current action

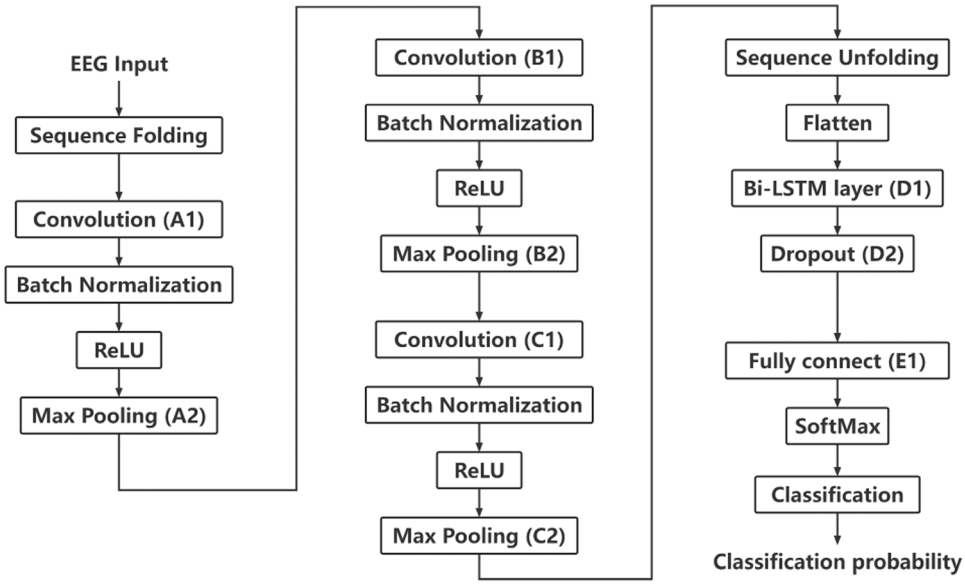

Fig. 6. Hybrid long short-term memory convolutional network (LSTM-CNN) structure used in EEG state recognition model.

Table 1
Long short term memory convolutional network parameters (The first column shows the layer names. In the second column, it shows the filter window size. The values in parentheses represent the filter's length and width respectively. The third column shows the moving steps of the filter window. These two parameters represent the number of steps the filter takes to move along the length direction of the image and the number of steps it takes to move along the width direction of the image, respectively. The forth column shows the filter numbers in the convolution layers. If the layer is a pooling layer, there is no need to set the filter number. In layers D1 and E1, the filter number represents the number of neurons used in each layer. D2 is a Dropout layer, thus '0.3 dropout' means there are 30% neurons do not participate in subsequent calculations).

Layer	Filter size	Stride	Filter number
A1	[3,10]	[1,1]	32
A2	[2,5]	[2,5]	-
B1	[2,5]	[1,1]	64
B2	[2,4]	[2,4]	-
C1	[2,3]	[1,1]	128
C2	[2,3]	[2,3]	-
D1	-	-	100
D2	-	-	0.3 dropped
E1	-	-	4

based on the previous and current signal states. Thus, there are two conditions for training the actor network. First, the predicted action should be close to the ideal action; therefore, we use cross-entropy as the first loss term. Second, the output actions continue to be inputted into the critic network to obtain a score. For the actor network, it is expected

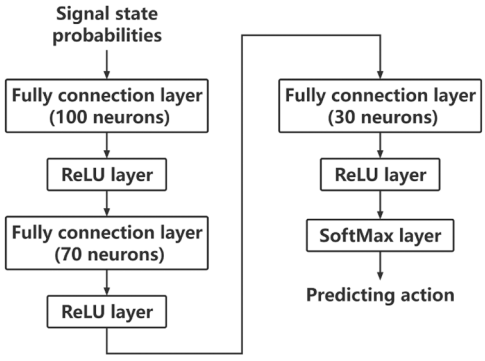

Fig. 8. Structure of actor network.

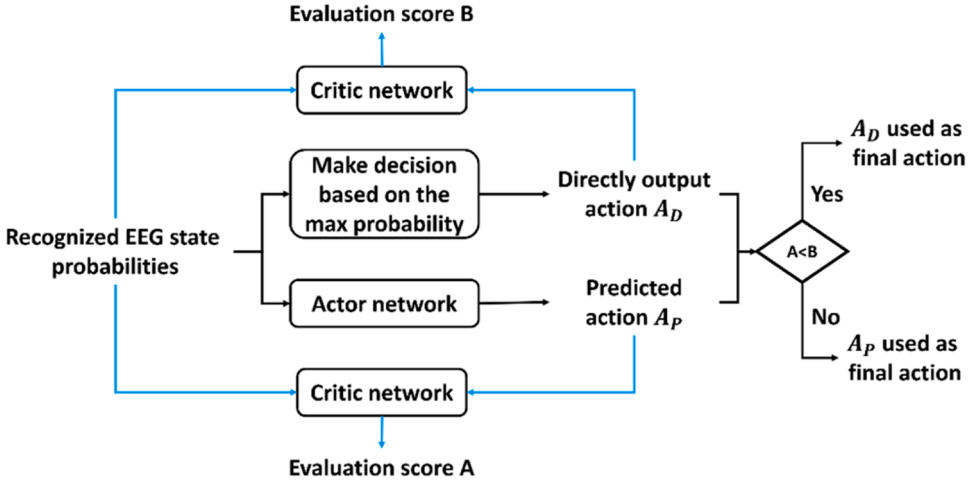

Fig. 7. Block diagram of the decision making model.

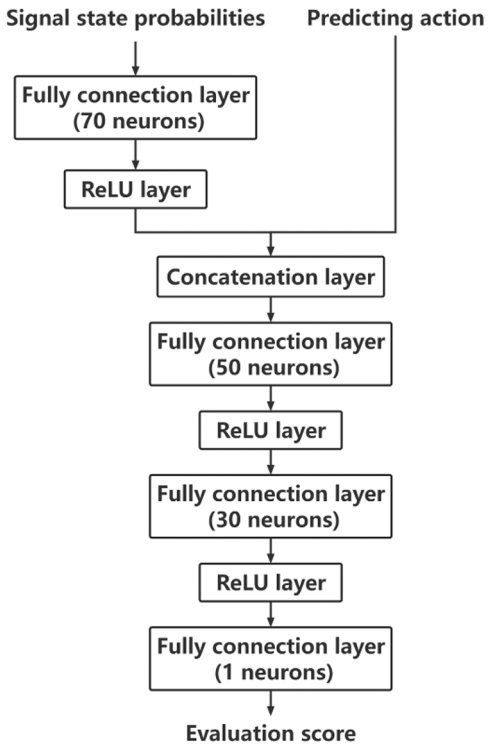

Fig. 9. Structure of critic network (Fully connect layer (70 neurons) means that 70 neurons are placed in this layer and connect each neuron to all neurons in the previous layer separately. The ReLU layer is an activation layer. When the input is less than 0, the output of that layer is 0. When the input is greater than or equal to 0, the output of the layer is the original input value. The function of the concatenation layer is to merge all neurons from both layers. For example, if the first target layer has 70 neurons and the second target layer has only 1 neuron, then the merged layer has a combined total of 71 neurons).

that the predicted action is a good action. Thus, the score should be close to one. We calculated the mean square error between the output of the critic network and 1. These two loss terms were added together as a loss function of the actor network.

$$L_A = -\frac{1}{N_m}\sum_{n=1}^{N_m} A(P_{sn})\log(Y_{sn}) + (1-A(P_{sn}))\log(1-Y_{sn})) + \frac{1}{N_m}\sum_{n=1}^{N_m}(1-C(P_{sn},A(P_{sn})))^2 \quad (1)$$

$$L_C = \frac{1}{N_m}\sum_{n=1}^{N_m}\left(1-C\left(P_{sn},a_g\right)\right)^2 + \frac{1}{N_m}\sum_{n=1}^{N_m}(0-C(P_{sn},a_b))^2 \quad (2)$$

where N_m is batch size; P_{sn} is the n_{th} EEG signal state probability vector; Y_{sn} is the n_{th} EEG signal action label; a_g is correct predicted action; a_b is incorrect predicted action; L_A is the loss of the actor; L_C is the loss of the critic; $A(P_{sn})$ is the output of the actor when inputting the EEG signal state probability vector. $C\left(P_{sn},a_g\right)$ is the output of the critic when the first input is the EEG signal state probability vector and the second input is the correct predicted action. The network is trained using the ADAM updating method. After training, the critic network can recognize good and bad actions. The actor network can predict actions.

2.4.3. EEG task classification model

2.4.3.1. Overview of model. Two classification models were used to classify the specific tasks. The first model was used to classify motor imagery tasks. The second model was used to classify eyeball movement tasks. A block diagram of the classification model is shown in Fig. 11. In this model, the task-state EEG data were first filtered using multiple sets of band-pass filters. In the previous section, we determined the exact signal state. If the signal state is motor imagery, we need to stack all of the frequency-filtered sub-signals to obtain the combined frequency-filtered signals that are used as the target subject EEG data. For example, the size of each filtered subsignal is 4×1000, and we have eight sets of bandpass filters. After we stack the signals, we obtain a signal with a size of 32×1000. If the signal state is an eyeball movement, the frequency-filtered subsignals are directly used as the target subject EEG data. Then, the auto-selected regularized common spatial pattern algorithm is applied to the target EEG data and other subjects' EEG data to obtain spatially transformed data (this algorithm is introduced in 2.3.4.2). Subsequently, if the signal state is motor imagery, mutual information-based best individual feature selection is applied to select the most effective spatial features (this algorithm is introduced in Section 2.3.4.3). Otherwise, if the task is eyeball movement, variance-difference-based best individual feature selection is applied, and the obtained spatial vectors are stacked along the channel dimension. Eventually, the spatial features were input into a convolutional neural network to classify the final task category.

2.4.3.2. Auto-selected regularized common spatial pattern. Consider S_c is the pre-processed target EEG data and $\widehat{S_c}$ is the EEG data of other subjects, we can calculate R_c that is the covariance of S_c and $\widehat{R_c}$ that is the covariance of $\widehat{S_c}$:

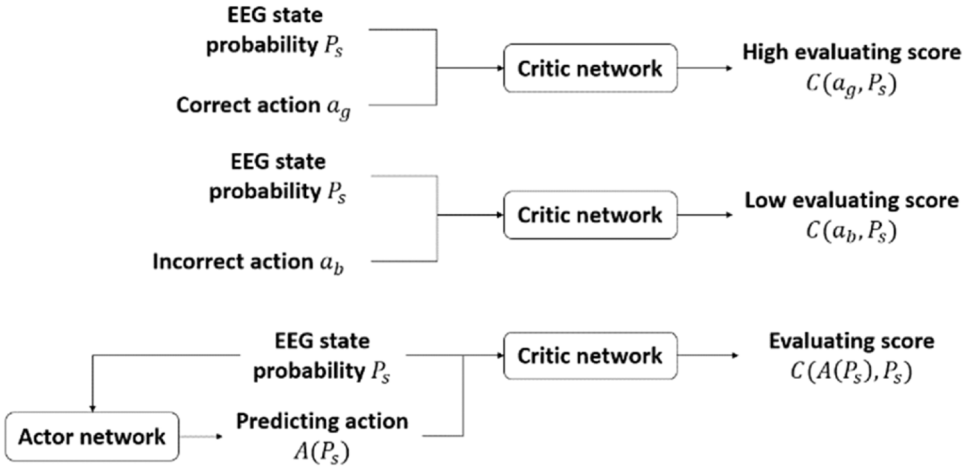

Fig. 10. The process of training the actor network and critic network.

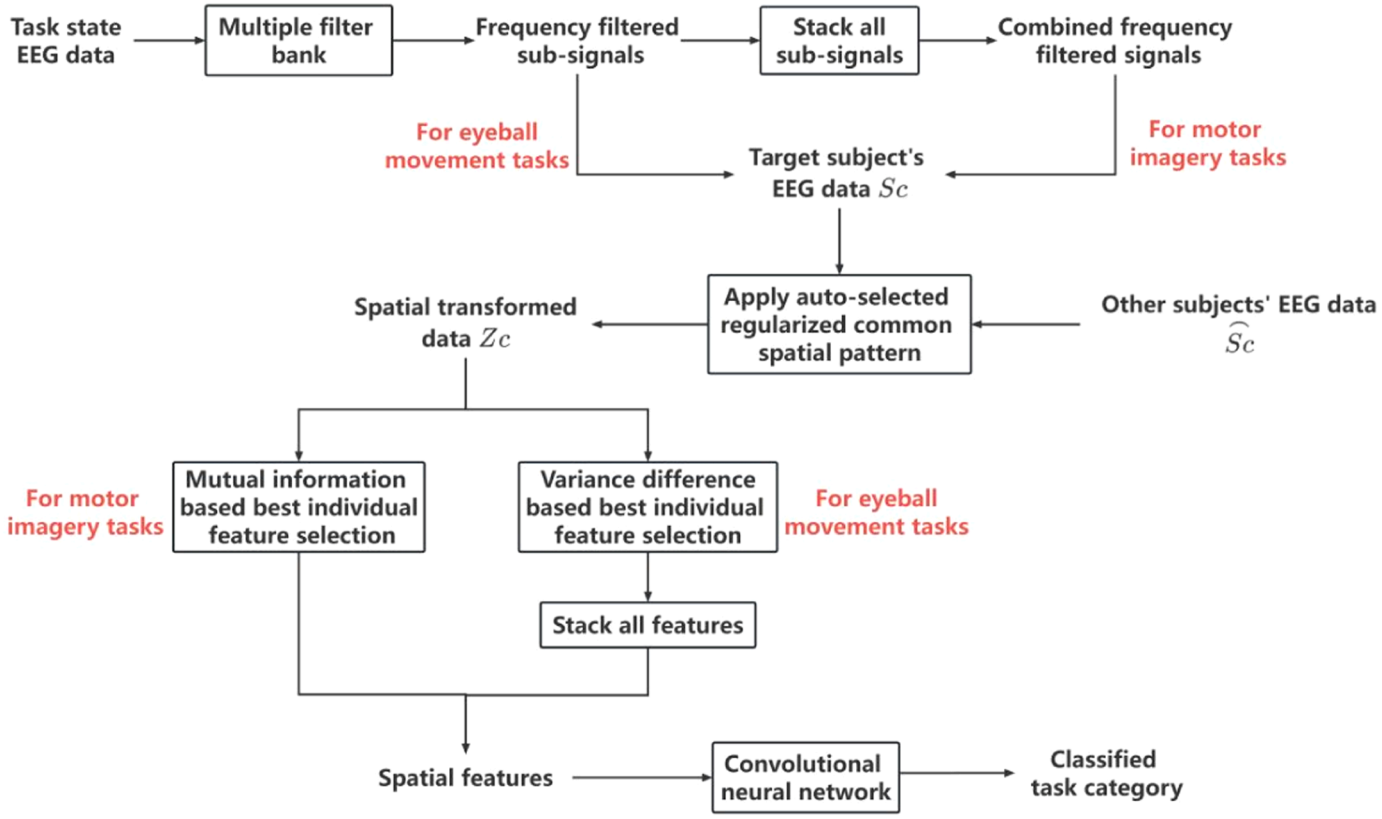

Fig. 11. Block diagram of EEG task classification model.

$$R_c = \sum_{n=1}^{N_e} \frac{S_{cn} S_{cn}^T}{trace(S_{cn} S_{cn}^T)} \tag{3}$$

$$\widehat{R_c} = \sum_{\hat{n}=1}^{\hat{N}_e} \frac{\widehat{S_{c\hat{n}}}\,\widehat{S_{c\hat{n}}}^T}{trace(\widehat{S_{c\hat{n}}}\,\widehat{S_{c\hat{n}}}^T)} \tag{4}$$

where *trace(•)* is the sum of elements on the diagonal of the matrix; N_e is the number of class of S_c; $\widehat{N_e}$ is the number of class of $\widehat{S_c}$. Then, we can obtain J_c, which is the regularized covariance matrix, and $\Sigma(\beta_c, \gamma_c)$, which is the mixed covariance matrix, by using the regularization parameters β_c and γ_c. We have:

$$J_c(\beta_c) = \frac{(1-\beta_c) \bullet R_c + \beta_c \bullet \widehat{R_c}}{(1-\beta_c) \bullet N_t + \beta_c \bullet \widehat{N_t}} \tag{5}$$

$$\Sigma_c(\beta_c, \gamma_c) = (1-\gamma_c) \bullet J_c(\beta_c) + \frac{\gamma_c}{N_c} trace[J_c(\beta_c)] \bullet I \tag{6}$$

where N_c is the total number of channels, β_c controls the variance of the estimated covariance, and γ_c is the second regularized parameter that can reduce large eigenvalues and increase small eigenvalues. Then, decompose the mixed covariance matrix and obtain the eigenvalue λ_c and eigenvector U_c. Sort eigenvalue U_c in descending order and obtain the whiting matrix P_w.

$$U_c \lambda_c U_c^T = \Sigma_c = \Sigma_{c1} + \Sigma_{c2} \tag{7}$$

$$P_w = \sqrt{\lambda_c^{-1}} U_c^T \tag{8}$$

where Σ_{c1} is the mixed covariance matrix of the first-class data; Σ_{c2} is the mixed covariance matrix of the second-class data. Apply P_w to the mixed matrix of the two classes to obtain the whiten matrix of the first-class data S_{w1} and the whiten matrix of the second-class data S_{w2}. After that, continue to decompose one of the class matrixes S_{w1} to obtain the eigenvalues λ_B and eigenvectors U_b.

$$S_{w1} = P_w \Sigma_{c1} P_w^T \tag{9}$$

$$S_{w2} = P_w \Sigma_{c2} P_w^T \tag{10}$$

$$U_b \lambda_B U_b^T = S_{w1} \tag{11}$$

Eventually, we can obtain the spatial filter W_c.

$$W_c = U_b^T P_w \tag{12}$$

2.4.3.3. Feature selection. Subsequently, we used the feature matrix and mutual information-based regularization parameter selection method [31] to select the regularization parameters and recalculate the final spatial filter. We applied the spatial filter corresponding to these two parameters to the pre-processed signal and obtained the spatially filtered data Z_c.

$$Z_c = W_c S_c \tag{13}$$

The filter is applied to the pre-processed signal to obtain the variance feature matrix of the first class X_{c1} and the variance feature matrix of the second class X_{c2}.

$$X_{c1} = \mathrm{var}(W_c S_{c1}) \tag{14}$$

$$X_{c2} = \mathrm{var}(W_c S_{c2}) \tag{15}$$

where var() is a function for calculating the variance. We can then obtain the variance difference D_v. If the variance-based best individual feature selection method is used, the channel data from the spatially filtered data Z_c with the largest variance difference are used as the final classification spatial feature. The corresponding labels were defined for the two types of variance features. The feature vector is X_c and the label

vector is Y_c. The information entropies $H_I(X_c)$ and $H_I(Y_c)$ can be calculated. $H_I(X_c)$ is the information entropy of the feature vectors X_c, and $H_I(Y_c)$ is the information entropy of the label vector Y_c. Then, the joint probability density function is used to calculate their mutual information $M_I(X_c, Y_c)$.

$$H_I(X_c) = -\sum_{x \in X_c} P(x)\log_2 P(x) \tag{16}$$

$$H_I(Y_c) = -\sum_{y \in Y_c} P(y)\log_2 P(y) \tag{17}$$

$$M_I(X_c, Y_c) = \frac{2\sum_{y \in Y_c}\sum_{x \in X_c} P(x,y)\log\left(\frac{P(x,y)}{P(x)P(y)}\right)}{H_I(X_c) + H_I(Y_c)} \tag{18}$$

where $P(x)$ is the probability of x, $P(y)$ is the probability of y, and $P(x,y)$ is the joint probability of x and y. If the mutual-information-based best individual feature selection method is used, the channel data from the spatially filtered data Z_c with the largest mutual information is used as the final classification spatial feature.

2.4.3.4. Structure of classifier. Finally, a convolutional neural network (CNN) is introduced to classify EEG features. The structure and parameters are presented in Fig. 12 and Table 2, respectively.

Table 2
Convolutional neural network parameters.

Layer	Filter size	Stride	Filter number
A1 Convolution	[1,10]	[1,1]	32
A1 Max Pooling	[1,4]	[1,4]	-
B1 Convolution	[1,8]	[1,1]	32
B1 Max Pooling	[1,3]	[1,3]	-
B2 Convolution	[1,1]	[1,1]	64
B3 Convolution	[1,1]	[1,3]	64
C1 Convolution	[1,5]	[1,1]	64
C1 Max Pooling	[1,2]	[1,2]	-
C2 Convolution	[1,1]	[1,1]	128
C3 Convolution	[1,1]	[1,2]	128
D1 Convolution	[1,3]	[1,1]	128
D1 Max Pooling	[1,2]	[1,2]	-
D2 Convolution	[1,1]	[1,1]	256
D3 Convolution	[1,1]	[1,2]	256
E1 Convolution	[1,2]	[1,1]	256
E1 Max Pooling	[1,2]	[1,2]	-
E2 Convolution	[1,1]	[1,1]	512
E3 Convolution	[1,1]	[1,2]	512
F1 Convolution	[1,2]	[1,1]	1024
G1 Average Pooling	[1,3]	[1,3]	-

3. Experiments and results

3.1. Introduction to hardware

3.1.1. EEG acquisition device

Unicorn Hybrid Black is a consumer-grade bio-signal amplifier kit (as shown in Fig. 13). The device can obtain EEG recordings using Bluetooth. It is a dry device (which does not need a bio-gel) that contains eight DC-coupled analog input channels with a 24 Bit resolution. The sample rate was 250 Hz. The EEG electrodes of this device have the advantage of fast and easy preparation with high-quality EEG signals [32].

3.1.2. BCI robot

The robot used in this experiment is illustrated in Fig. 14. The body and arm of the robot are made of acrylic sheets. There are eight engines

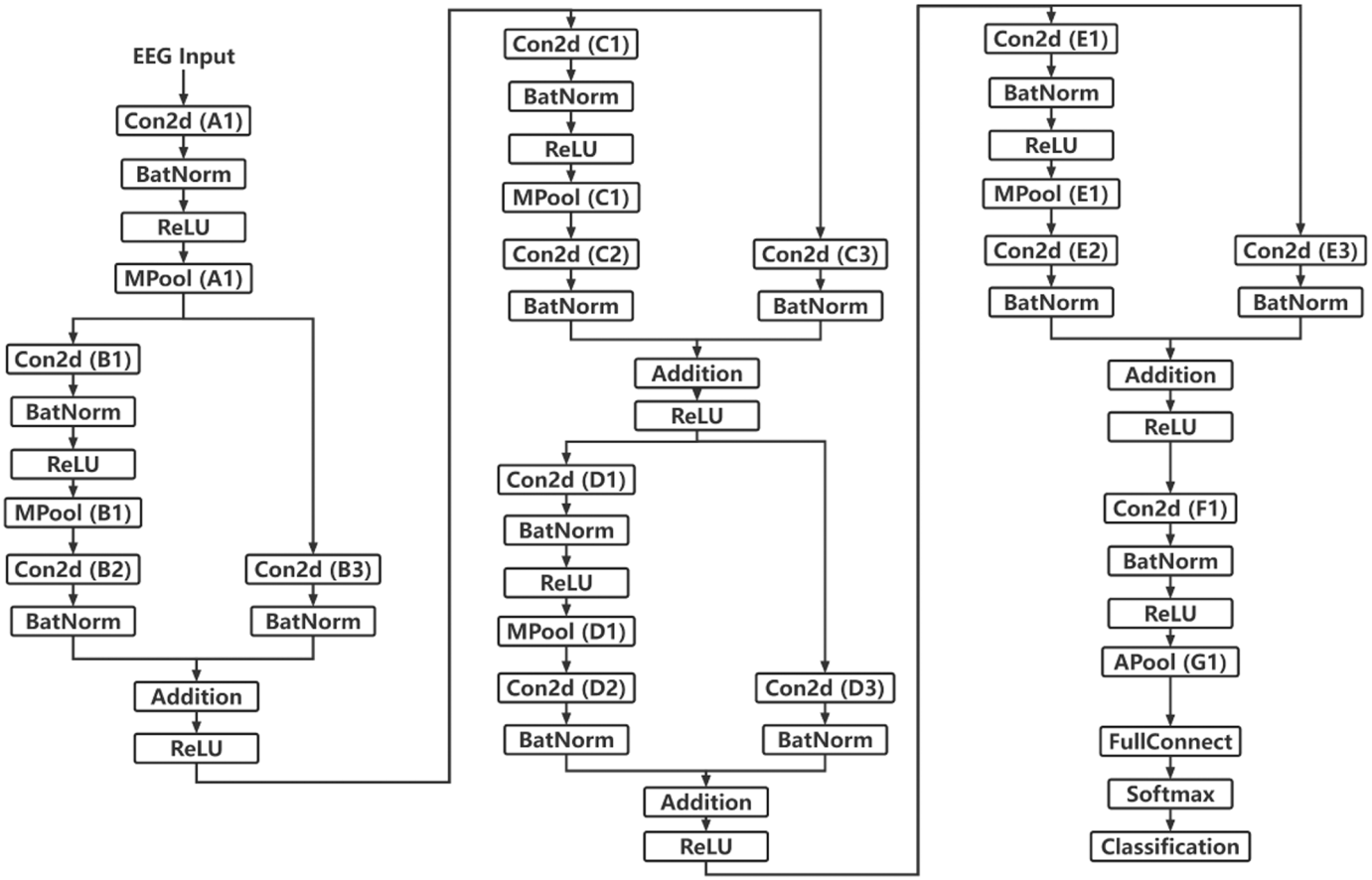

Fig. 12. Convolutional neural network structure used as classifier.

Fig. 13. Unicorn Hybrid Black EEG acquisition device.

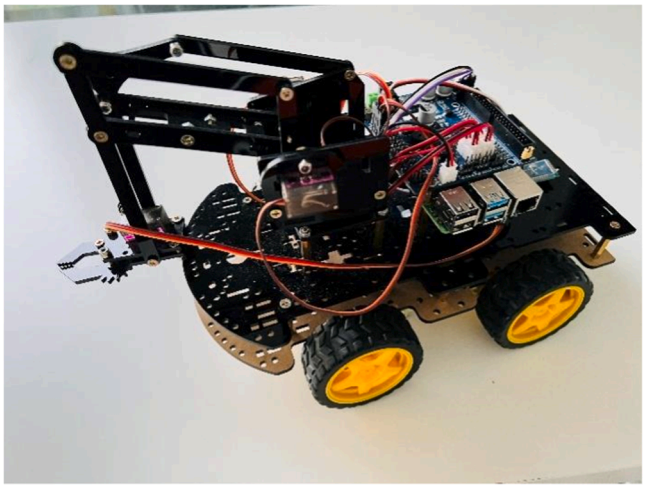

Fig. 14. Raspberry Pi 4 robot.

in which four engines are used to drive the four wheels and another four engines are used to extend the arm, retract the arm, control the direction, and control the grasp or release action. The CUP of the robot is Raspberry Pi 4, which is a single-board computer, and its operating system is Linux. The expanding board is PCA9685, which is a PWM/servo driver used to control the eight engines. Inside this robot, a non-blocking I/O model was built to process the received commands and control the driver. The robot can communicate with other devices or software through WebSocket.

3.2. Experiment preparation

3.2.1. Data collection

We collected EEG data using a Unicorn Hybrid Black EEG acquisition device. This experiment was approved by the University of Technology Sydney (UTS). The ethics approval number was ETH22–7056.

The subjects performed the corresponding tasks according to screen prompts. The timing schemes of the paradigms are shown in Figs. 15 and 16. The first 15 s was the beginning of the experiment, and the screen prompted the start of the experiment. After that, there was a one-second cue time before the start of the new trial. When the trial started, the screen prompted the participants to prepare for the brain tasks. The subject had 2 seconds to prepare. The screen then shows a specific task. The subjects had 4 seconds to perform the task. Finally, the subject had 2 seconds break time. The experiment consisted of two motor imagery tasks, three eyeball movement tasks, and one EMG task, namely imaginary left-hand movement, imaginary right-hand movement, eyeball move to the left, eyeball move to the right, eyeball move down, and bite teeth. One trial of the experiment contained 9 s of data collection. Each experiment consisted of 50 trials for each motor imagery task and 30 trials for the EMG or eyeball movement task. There were 220 trials for each experiment. Each participant completed four experiments. Thus, there were 880 trials for each subject.

Because the motor imagery task is abstract and difficult to complete, we let the subjects imagine several movements to stimulate their imagination ability. Two cups were placed on a table. At the beginning of the experiment, the subjects placed their hands naturally on their legs. When the screen prompted imaginary left or right movements, the subjects first imagined lifting the hand, then holding the cup, and finally placing the cup on the leg.

3.2.2. Online data segmentation strategy

Performing brain tasks requires a certain amount of time. During the execution of these brain tasks, the brain transitions from one state to another. The recognition model calculates the probability for each second. The decision-making model can predict the final action based on the current and past few-state probabilities. EEG signals at successive times were correlated. Therefore, when we trained these two models, we could not only obtain the data of the time when the subjects were performing the task. We should also consider the data every second and use continuous probabilities to train the model.

We defined a window of size 1000. where each entry contains data records for a period of 4 s. A window was set up at the beginning of the filtered signal. The window slides every second along the time sequence of the collected EEG signals. Every second, the window can extract 8 $\times$ 1000 data records. During data collection, when a prompted task appeared on the screen, the subject began to perform the task for 4 s. Thus, every time a motor event occurred, the effective time of the task was 4 seconds after the event marker. The data during this 4 second period are relevant.

As the window moved along the time sequence, it overlapped seven

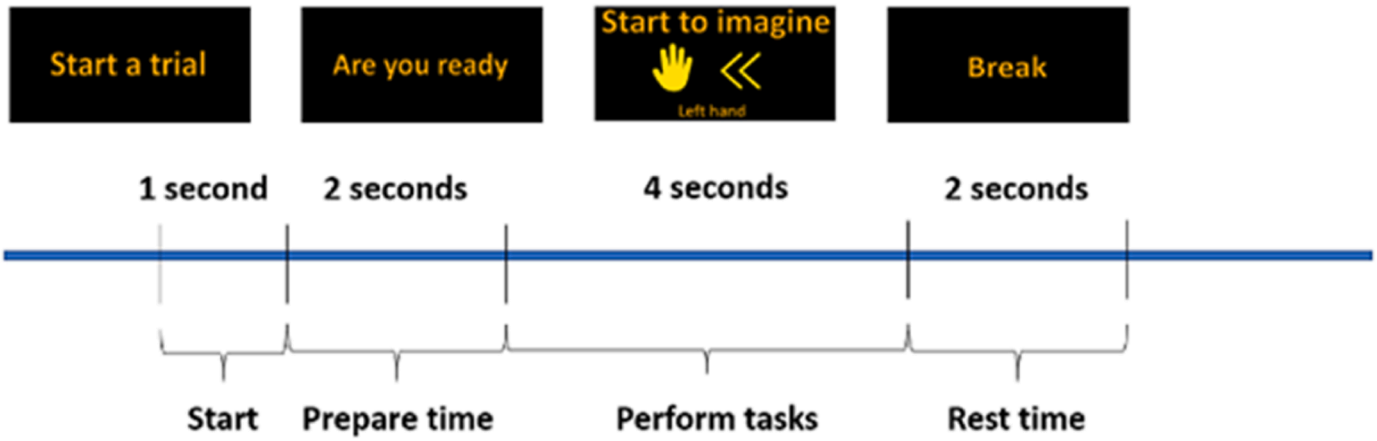

Fig. 15. Timing scheme of the paradigm for one trial.

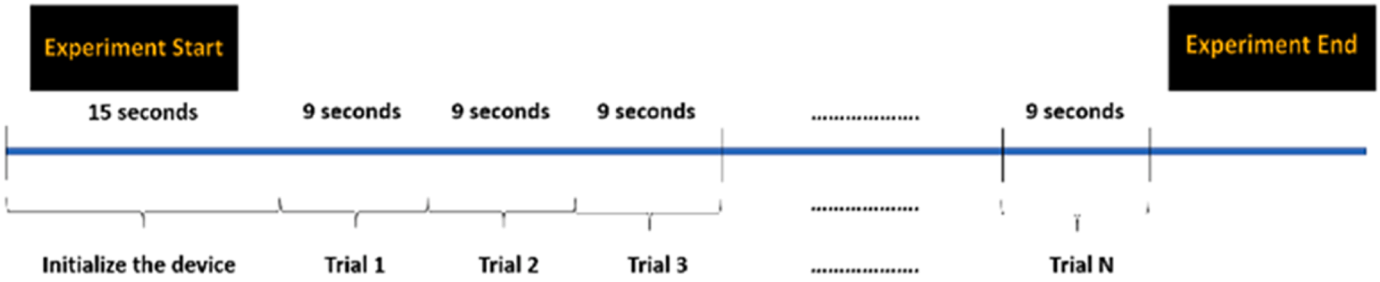

Fig. 16. Timing scheme of the paradigm for one experiment.

times with the effective data in each trial. The maximum overlap time was 4 s. If the window overlapped with the relevant data for 3 s or more, this window data was used as one epoch of the task state data. Otherwise, if the window overlapped with the relevant data for less than 3 s, this window data was used as one epoch of the rest state data. Thus, in each trial, we extracted three task-state training data records and four rest-state training data records. In addition, if the window does not overlap with the relevant data, the data of this window are the pure rest data. Thus, for each subject, there were 2640 trials of task state data (880 data points × 3 windows) when using this epoch extraction strategy. For each subject, there were 3520 trials of resting-state data (880 data × 4 windows) using this strategy.

3.2.3. Train and test the classifier

By employing an online data segmentation strategy, we obtained specific quantities of different types of task state data. This included 400 groups of motor imagery data, 360 groups of eyeball movement data, and 120 groups of EMG data. Each group of task state data consists of three windows, whereas each group of resting-state data consists of four windows.

For training and evaluation purposes, we employed a ten-fold cross-validation method to ensure the robustness of our results. During each training step, we used specific quantities of data for the training set. This included 1080 motor imagery data (360 data × 3 windows), 972 eyeball movement data (324 data × 3 windows), 324 EMG data (108 data × 3 windows), and 3168 rest data (792 data × 4 windows). In addition, we employed 120 motor imagery data (40 data × 3 windows), 108 eyeball movement data (36 data × 3 windows), 36 EMG data (12 data × 3 windows), and 352 rest data (88 data × 4 windows) for the testing sets. The initial learning rate was set at 0.01 and the mini-batch size was 64. The network is trained using the ADAM updating method.

3.2.4. Commands design

As shown in Table 3, several control commands were assigned in the experiment, and each command corresponded to a specific action. Because of the long reaction time of motor imagery, such signals are not suitable for commands such as go forward and stop, which require a fast response. In the experiments, arm-related control commands did not require a fast response. Thus, left-hand imagery movement was used to control the arm in the switch direction. The right hand imagery movement was used to control the forward movement of the arm. During the control process, the most important command is stopped because it needs to be used at any time to terminate the execution of the action. The reaction time of the stop command must be very short, and the machine should be able to recognize this command at any time. From the results of the model construction, the EMG signal can provide feedback immediately and accurately. Therefore, the EMG signal generated by the biting of teeth was used as the stop command. Eyeball movement classification is easier to perform than motor imagery. Thus, leftward and rightward eyeball movements are used to control the robot to turn left and right, respectively. The downward eye movement was used to control the forward movement of the machine.

Table 3
Control commands corresponding to actions.

Action	Commands
Biting teeth	Stop
Left hand motor imagery	Switch direction of arm
Right hand motor imagery	Arm move forward
Leftward eyeball movement	Turn left
Rightward eyeball movement	Turn right
Downward eyeball movement	Go forward

3.2.5. Robotic models connection and data transmission protocol

The hybrid BCI real-time control system comprises three parts: a data transmission system, an EEG-based dynamic classification system, and a real-time BCI robot. The data transmission system includes a lab streaming layer, a data acquisition server, and a data processing and reception model. The EEG-based dynamic classification system includes a pre-processing model, denoising model, EEG state recognition model, actor-critic-based decision-making model, and EEG multitask classification model. The corresponding equipment and environment of the proposed system are shown in Fig. 17.

In the data classification system, we used training data with a length of 1000 points. The collection device operated at a sampling frequency of 250 Hz, resulting in the collection of 250 data points per second. A data transmission diagram is shown in Fig. 18. When transferring the data, we extracted the first four seconds of data from the current time point and input it into the classification system to obtain commands. The data recognition time was set to one second, meaning that the extracted four seconds of data were classified per second. Real-time data are transmitted through the Bluetooth transmitter to the central processing unit (CPU) and then to the EEG-based dynamic classification system via the data transmission system. Finally, the commands generated by the classification system were sent to the robot through WebSocket.

When the robot executes commands, the command execution time must exceed one second. Thus, there may be a situation where the robot does not complete the current command and the system gives one or more different commands in the meantime. When this situation occurs, the robot must execute the previous command before executing the next command. However, if a different command is received as a stop command, it will execute the stop command immediately.

3.3. Experiment setup

3.3.1. Experiment A: EEG state recognition and decision-making model performance evaluation

The EEG state recognition and decision-making models were used to determine the signal state in the proposed EEG-based dynamic classification system. These two models aim to calculate the probabilities of the four EEG signals and predict the final action based on state probability changes at different times. Therefore, these two models require continuous data. The four EEG signals are i) rest state EEG signals, ii) motor imagery EEG signals, iii) eyeball movement EEG signals, and iv) EMG signals.

Using the online epoch extraction strategy, we obtained 400 trials of

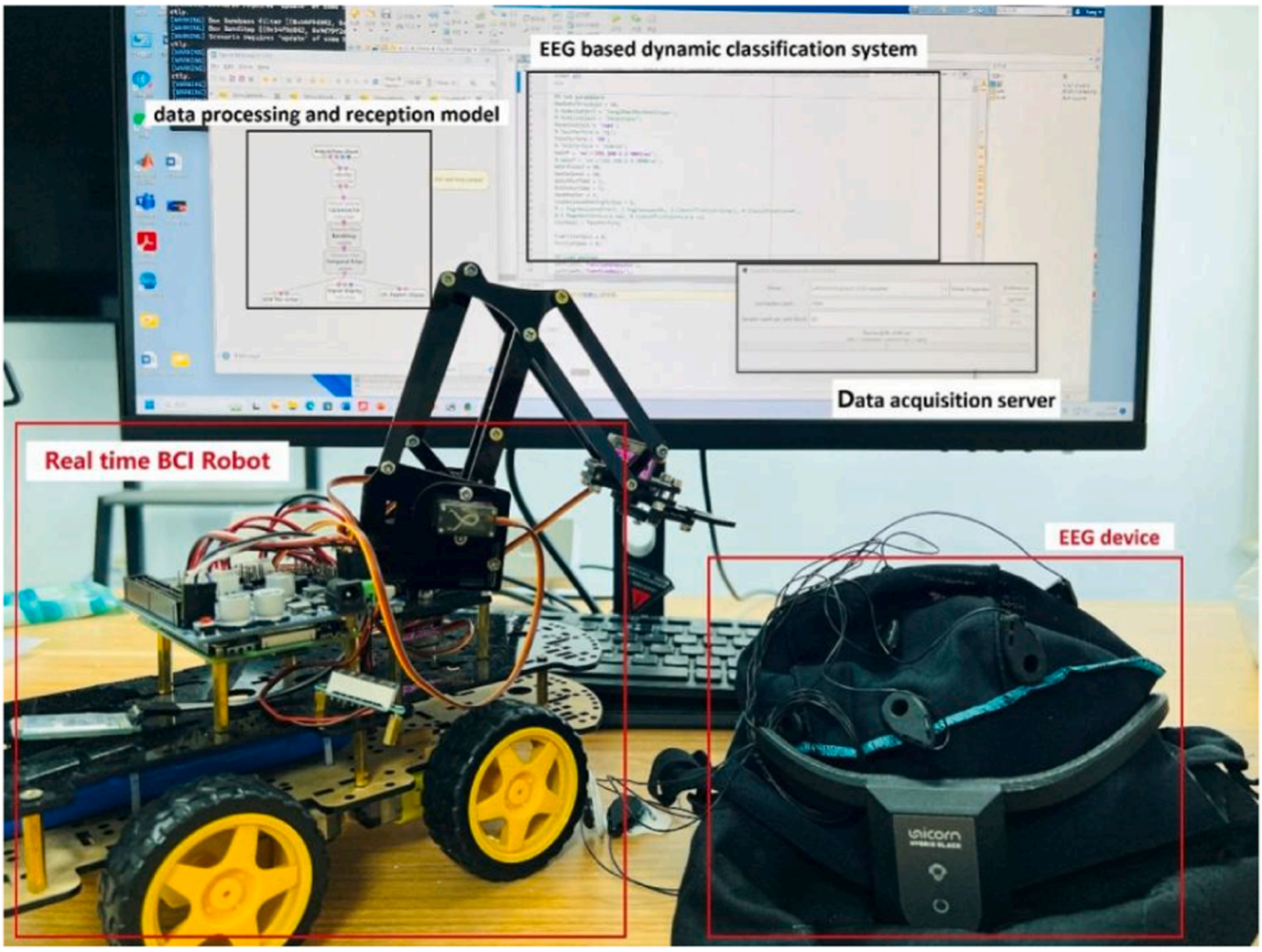

Fig. 17. The equipment and environment for the hybrid BCI real time control system.

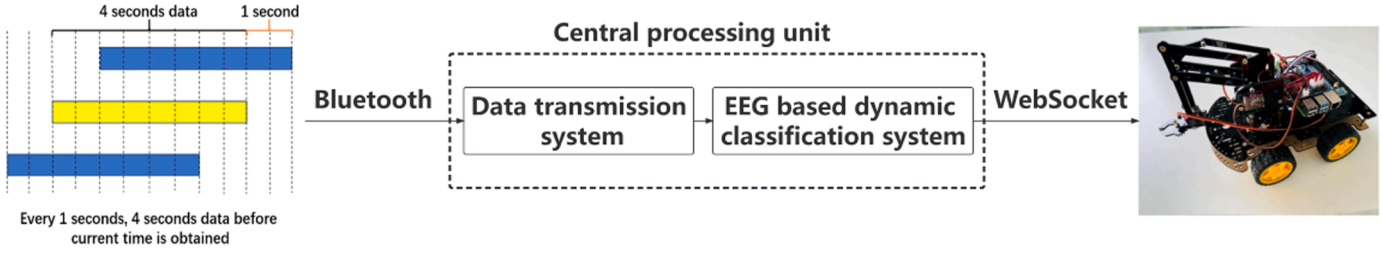

Fig. 18. data transmission diagram.

motor imagery data, 360 trials of eyeball movement data, 120 trials of biting teeth data, and 880 trials of rest state data. Each trial of the task data contained seven segments of data. Segmented data were used to train and test the model. When we tested the model, we input seven segmented data into the model. The model will output seven probability vectors. For each vector, the output with the highest probability is used as the recognized label. Thus, the model generates seven recognized labels. If one or more recognition outputs are the correct label and the other outputs are the rest state labels, then the output of this trial is judged to be the correct output. Otherwise, if the output has two or more different labels or the output category is inconsistent with the target category, the output of this trial is judged to be misclassified.

After training the recognition model, we used the model to calculate the probabilities of the four EEG signals. We then input the current state probability vector and the three most recent state probability vectors into the actor-critic-based decision-making model to predict the final action. We also used traditional methods to predict the action and compared their recognition performance. Table 4 presents the results.

From the results, these four signals were easy to distinguish because the time and spatial domain features of these three signals were distinct. They achieved an accuracy of $> 80\%$. The models with the best performance were the EEG state recognition model and a combination of the recognition and decision-making models. The decision-making model can help the model avoid errors because it considers not only the current signal state probability but also the past signal state probabilities. Thus, the performance of the decision-making model is better

Table 4
Evaluation of the performance of EEG state recognition model and decision-making model.

		Support Vector Machine (SVM)	Traditional Convolutional Neural Network (CNN)	EEG state recognition model	EEG state recognition model + Actor-critic based decision making model
Subject A	Accuracy (%)	87.22	92.05	94.49	96.59
	kappa	0.8059	0.8787	0.9159	0.9479
	Se	0.0250	0.0261	0.0268	0.0272
Subject B	Accuracy (%)	83.58	91.14	92.61	94.26
	kappa	0.7507	0.8646	0.8872	0.9125
	Se	0.0240	0.0259	0.0262	0.0267
Subject C	Accuracy (%)	84.38	90.74	92.22	94.03
	kappa	0.7628	0.8585	0.8807	0.9087
	Se	0.0242	0.0257	0.0261	0.0266
Subject D	Accuracy (%)	88.32	92.16	93.87	96.43
	kappa	0.8143	0.8793	0.9038	0.9437
	Se	0.0247	0.0263	0.0263	0.0269
Subject E	Accuracy (%)	85.48	91.84	93.12	95.41
	kappa	0.7785	0.8681	0.9015	0.9318
	Se	0.0242	0.0261	0.0265	0.0267
Mean	Accuracy (%)	85.80	91.59	93.26	95.34
	kappa	0.7824	0.8698	0.8978	0.9289
	Se	0.0244	0.0260	0.0264	0.0268

than that of the recognition model. Finally, we used the best model for the real-time control experiments.

3.3.2. *Experiment B: offline robotic performance evaluation*

Online segmented data were used to train and build a hybrid BCI real-time control system. There are six control commands and one rest command, which are explained in the command design section. As in the data collection steps, command prompts were randomly generated on the screen. According to the instructions that they were given, subjects had to perform the corresponding actions of the commands within 4 s. There was a 2-second preparation time before the prompt occurred. Each command appeared randomly 50 times, for a total of 300 actions. After that, the data of the previous 4 s were taken every second and input into the hybrid EEG real-time control system. The system outputs classification results. The data length was 1815 s, including 15 s of system initialization time. During the 4 s of action execution, if one or more command outputs are the real label command and the other outputs are the rest state label, then the command is judged to be the correct output. If the output has two or more different commands or the output category is inconsistent with the target category for these four seconds, then the command is judged to be misclassified. This is how the control accuracy of the EEG real-time control system is evaluated. Table 5 lists the real-time classification accuracies for each command.

The results show that for offline testing, the recognition accuracy of the stop command can reach 100%. The classifications of turning left, turning right, and going forward commands are more accurate than arm-related commands controlled by motor imagery. Motor imagery depends on the imagined abilities of the different subjects. Subjects with better imagined abilities may be more likely to perform motor imagery. The overall recognition accuracy exceeded 85%. Therefore, the classification accuracy of this system is sufficient, and it can be used for real-time control experiments.

3.3.3. *Experiment C: Online robotic performance evaluation*

The actual environment used to test the robot is shown in Fig. 19. First, we set the starting and destination points. We then placed obstacles between the two points and placed a target object at the destination. The subjects were required to control the robot car from the starting point while avoiding obstacles and reaching the destination. During this process, there were target objects on the road. The subjects were required to control the robot arm to push the target objects down.

In this process, we recorded the time spent by the robot car to reach its destination from the starting point. At the same time, we also recorded the driving path and the moving distance of the car. Before the experiment, we used a remote controller to control the car to perform the above tasks and recorded the time, route, and driving distance. These were used as references to evaluate the reliability of EEG real-time control systems. The subject was required to control the robot by following the reference route. Using a remote controller, we used a keyboard to manually control the robot and execute one command every three seconds. Table 6 presents the online testing performance of the control system.

Table 5
Offline testing performance of real time control system.

Correct /total number	Backward /stop	Turn left	Turn right	Go forward	Arm direction switch	Arm move forward	Sum
Subject A	50/50	46/50	47/50	49/50	43/50	42/50	277/300
Subject B	50/50	44/50	45/50	48/50	36/50	34/50	257/300
Subject C	50/50	43/50	39/50	41/50	29/50	31/50	233/300
Subject D	50/50	47/50	47/50	48/50	44/50	41/50	277/300
Subject E	50/50	45/50	46/50	47/50	39/50	37/50	264/300
Mean correct number	50.0/50	45.0/50	44.8/50	46.6/50	38.2/50	37.0/50	261.6/300

Fig. 19. The actual environment used to test the robot.

Table 6
Online testing performance of real time control system.

Subject	Spending time (s)	Running distance (m)	Target object (5) push down
Subject A	181.80	6.42	5
Subject B	260.26	6.71	5
Subject C	266.23	6.73	4
Subject D	200.85	6.63	5
Subject E	220.80	6.43	4
Mean	225.99±36.76	6.58±0.15	4.6
Reference (remote controller)	171.20	5.93	5

4. Discussion

4.1. The benefits of combining EEG recognition and actor-critic based decision-making model

LSTM-CNN is a hybrid network that combines CNN and LSTM. A CNN can extract time-domain features of the signal and compress the features. EEG signals have multiple channels; therefore, they can be considered high-dimensional signals. Compared to other classifiers, LSTM-CNN is more suitable for extracting features along the time sequence of high-dimensional data.

The signal was input to the recognition model every second, and the length of each input signal was fixed. When the user performs a brain task, especially a motor imagery task, the task cannot be completed instantaneously. The task execution process required a certain amount of time. Thus, we assumed that the output of our recognition model was accurate and that the probability of the output represented the degree to which the subject completed the task.

For example, a motor imagery task requires 4 s to complete. We input the 4 s signal into the recognition model every second. We then obtain four probabilities for the signal states. Initially, if the user does nothing, then the probability of the rest is 100%, and the probability of motor imagery is 0. When the user performs the action for 2 s, the 2 seconds rest state signal and the 2 seconds motor imagery signal are input into the recognition model. Thus, the probability of motor imagery should be approximately 50%, and the rest state probability should also be approximately 50%. When the user performs the motor imagery action for 4 s, the entire task interval signal is completely inputted into the model. In this case, the probability of motor imagery should be 100%, and the remaining state probability should also be 0. Therefore, when the user performs a task, the output probability of the task state should increase from 0% to 100%, and then return to 0 when the task is completed. The output rest state probability should decrease from 100% to 0 and then increase to 100%. This is an ideal probability changing process. However, we could not ensure that the trained recognition model was perfect. Therefore, in practical applications, the output cannot reach the ideal state, as mentioned above. Therefore, the desired probability cannot be outputted every time. This means that there are some errors if we consider only the current probability output from the recognition model. Thus, the purpose of the proposed decision-making model is to learn the probability-change process. It extracts the relationship between the probabilities at different times.

Two networks are used to predict the action. First, the tasks were performed gradually, and the signal input into the model at each time overlapped with the signal at the previous time. In other words, they shared their own information. Thus, the current signal state is related to past signal states. Consequently, the input of the first network is the state probability vector that contains both the current state probability and past state probabilities. The purpose of this network is to learn the relations between the signal state probabilities. Based on this relationship,

the correct ideal current action can be predicted.

However, action probability is a discrete variable; therefore, we do not know whether the predicted probability is good. Thus, we created another network that evaluates whether the action predicted by the actor network is close to the ideal action. Its inputs are the current and previous state probabilities, and the current predicting action. The output is the evaluation score of this action, where 0 indicates bad and 1 indicates good. Initially, we trained the critic network so that the network could judge whether the action was good or not. Subsequently, we trained both networks simultaneously to reach an equilibrium state. Finally, the actor has the ability to output an ideal action based on the previous few signal state probabilities, and the critic has the ability to judge whether the actor's output is good.

To prove the feasibility of the proposed model, we determined the state changes of the EEG signals according to the task execution by the subjects. Fig. 20 shows the task category that the subject should perform, where the abscissa represents time and the ordinate represents the signal state. When the subject was performing the tasks, the recognition window was moving, and the window overlapped with the valid data segment three times. The ideal outputs of these three windows of data are the executed actions when the subject performs tasks according to the instructions, and the types of these three actions should be the same because they are continuous.

Fig. 21 shows the actions predicted using only the EEG recognition model. The EEG state recognition model can accurately identify the target signal state within the effective task execution time, because the characteristics of the signal state can be easily classified using the proposed EEG state recognition model. However, it can be seen that the output of this model does not fully conform to the expected action, and the action sometimes has a delay. In addition, owing to external influences, when the subject is resting, the subject may also generate unexpected brain activities. Fig. 22 shows the predicted actions using the actor-critic-based decision-making model. When we apply this model to the experiment, we can see that most of the delay problems have been resolved, and the randomly generated actions owing to external noise have also been corrected.

The output obtained from the recognition model only considers the signal state at the current time. Therefore, in this case, the target task can only be recognized when the action features contained in the signal are sufficiently distinct. However, because of external factors, it is difficult to extract significant features from the EEG signal; therefore, the recognition accuracy of the target task is very low. At the beginning, the action just starts to be executed, and the features of the signal state are not distinct enough; therefore, it is difficult to recognize. This can lead to delays. When we use the decision-making model, it considers the changes in both the previous and current signal states. This model can predict a subject's intentions based on this change. Therefore, the model can provide the correct output before the action is executed and avoids the occurrence of vacation delay. In addition, EEG users cannot concentrate for long periods of time and may sometimes take unconscious actions. If the recognition model is used alone, it can only determine the current action and execute the current action. However, this action is sometimes not the expected action of the user. The decision-making model can determine whether the current action is randomly generated by evaluating the signal status at a previous time. Thus, it can determine whether an action is unconsciously generated by the user to correct the signal output. Consequently, this model can make the control system more stable and reliable.

4.2. Compare the classification network using different structures

In the classification system, we used a convolutional neural network as a classifier. The performance of the model can be influenced by several factors such as the number of layers, neurons, and activation functions within the network. To thoroughly explore the effects of these parameters, we conducted experiments in which we tested networks with different structures using the same training and testing sets of task-state data. This dataset comprised 360 trials of eyeball movement data and 400 trials of motor imagery data. To ensure robustness, we employed a ten-fold cross-validation method to evaluate the performance of these diverse network configurations.

The convolution size is unchanged and is built based on the single-channel-based series network proposed in [31]. The reason for using This network is used because a single-channel learning strategy can extract effective information from each independent channel, which prevents the information between adjacent channels from affecting each other. This is better for extracting useful information from spatially transformed EEG signals. We used three convolutional layers, one max pooling layer, two activation layers, three batch normalization layers, and one addition layer as the hidden layer unit. Fig. 23 shows the hidden-layer unit. The structure of the classification network is illustrated in Fig. 24.

Our purpose was to investigate whether network structure affected the performance of the model. First, we examined the impact of changing the number of hidden-layer units. Second, we assessed the effects of modifying the number of convolution filters. (in Fig. 22, the parameters are in each convolution layer and hidden layers by changing L_{Start}, L_{End}, L_A to L_N). Third, we investigated the influence of different

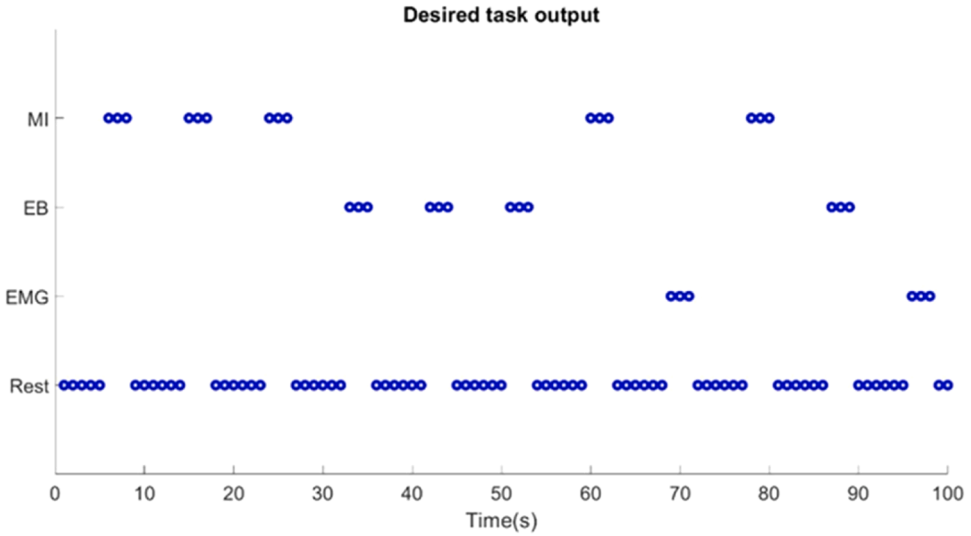

Fig. 20. The desired task labels.

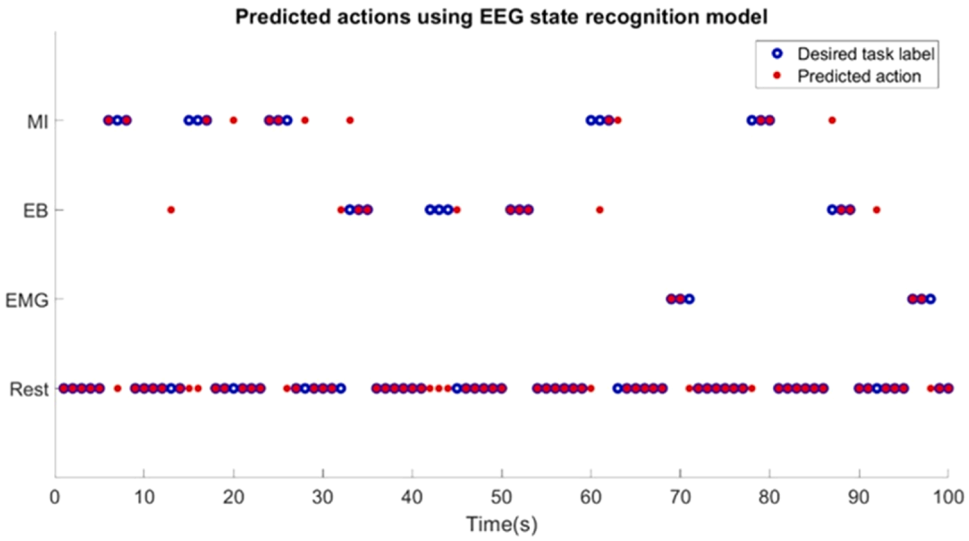

Fig. 21. The predicted actions using EEG recognition model.

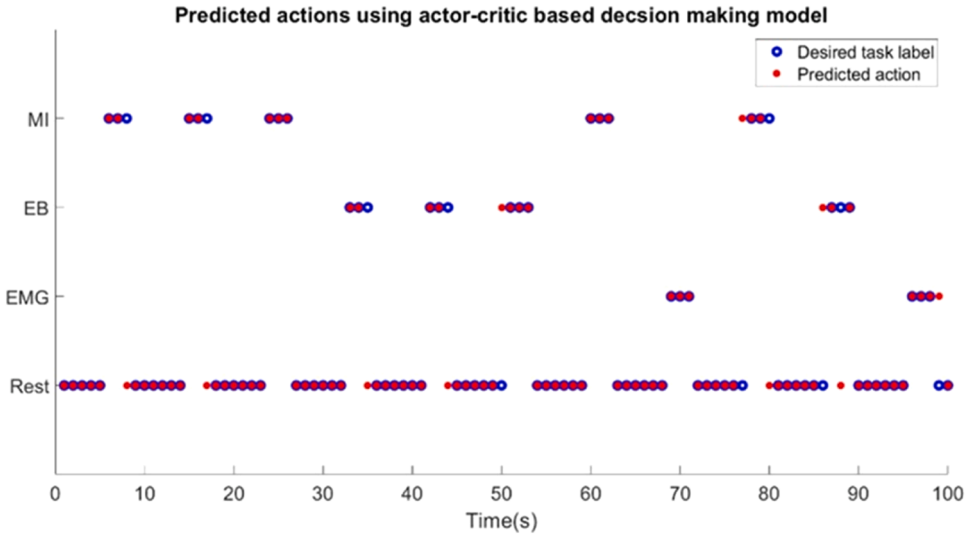

Fig. 22. The predicted actions using actor-critic based decision making model.

activation functions on each hidden layer unit. Tables 7, 8, and 9 present the test results of the network with different structures.

From Table 7, the mean accuracy of using the ReLU activation function was greater than that of using the Tanh or Sigmoid activation functions for both eyeball movement and motor imagery tasks. Through statistical analysis, we calculated the p-values using different activation functions. For the eyeball movement classification task, the p-values of accuracy were 0.000000058 (compare ReLU and Tanh activation function) and 0.000000027 (compare ReLU and Sigmoid activation function), which were all less than 0.05. For the motor imagery classification task, the p-values of accuracy were 0.0078 (compare ReLU and Tanh activation function) and 0.0056 (compare ReLU and Sigmoid activation function), which were all less than 0.05. These statistical results prove that when a single channel-based series network structure is used, the network with the ReLU activation function is better than that with the Tanh or Sigmoid activation function.

From Tables 8 and 9, it can be observed that the accuracy of using the three networks did not change significantly. As shown in Table 8, the highest mean accuracy was obtained using four hidden layers. For the eyeball movement classification task, the p-values of accuracy were 0.7431 (compared using three hidden layers and four hidden layers), 0.8678 (compared using four hidden layers and five hidden layers), and 0.6577 (compared using three hidden layers and five hidden layers). For the motor imagery classification task, the p-values of accuracy were 0.9668 (compared using three and four hidden layers), 0.9197 (compared using four and five hidden layers), and 0.9498 (compared using three and five hidden layers).

As shown in Table 9, the highest mean accuracy was obtained using the convolutional structure R2. For the eyeball movement classification task, the p-values of accuracy were 0.6434 (compared using structures R1 and R2), 0.5181 (compared using structures R1 and R3), and 0.1499 (compared using structures R2 and R3). For the motor imagery classification task, the p-values of accuracy were 0.8323 (compared using structures R1 and R2), 0.8617 (compared using structures R1 and R3), and 0.9673 (compared using structures R2 and R3). From the statistical testing results, the p-values of accuracy were all greater than 0.05, indicating that these accuracy results did not have significant differences. The statistical analysis results demonstrate that the number of hidden layers and convolution filters may not significantly affect the overall performance of the network.

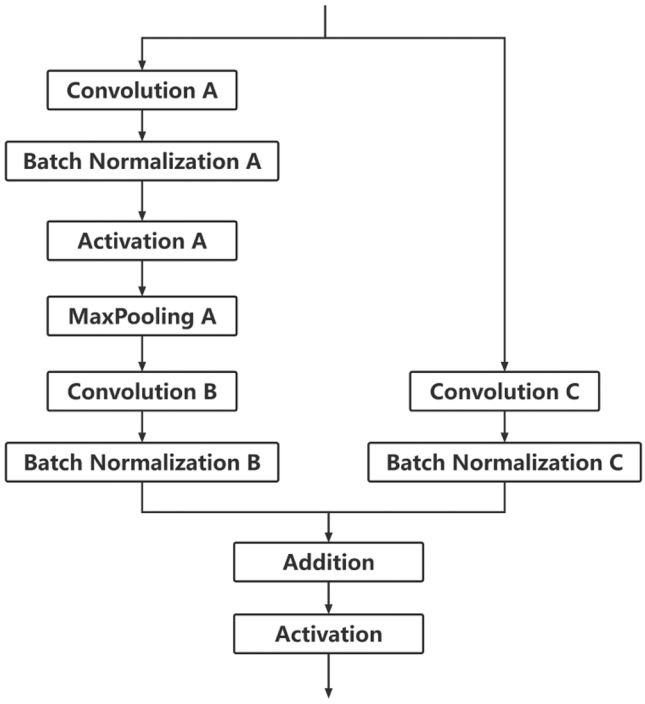

Fig. 23. Hidden layer unit.

4.3. Evaluate the performance of hybrid BCI real time controlled robot

To demonstrate the effectiveness of our proposed BCI control system, we have incorporated feature extraction and classification algorithms from two highly relevant papers' works of [26] and [27], respectively. We used the same data collection, transmission, and control commands. We then employed the control methods proposed in [26] and [27] to allow the subjects to control the target robot. The systems proposed in these two studies were jointly controlled by multiple biological signals.

[26] proposed a multimodal Human-Machine Interface (mHMI) system that combines EEG, EMG, and EOG to jointly control the motion of the robotic arm. This was similar to the control concept of the proposed system. [27] proposed a real-time electroencephalogram (EEG)-EMG Human–Machine Interface (EEG-EMG HMI)-based control system that combines EEG and EMG signals to jointly control mechanical lower limb movement.

[26] first used wavelet transform to process the signal and then extracted the ratio of the 8–12 Hz ERS and ERD changes in the C3 and C4 channels, respectively. This ratio was then input as a feature into the SVM for classification. For EOG signals, the author set a threshold to determine whether the eyes were moving. If the signal amplitude is greater than the threshold, it indicates that the subject performs eye movements, and a positive or negative pulse signal is used to determine whether the eyes are looking left or right. Common features of surface EMG, such as mean absolute value, waveform length, zero crossing, slope sign change, and mean absolute value slope, are used to classify

Table 7
Compare the accuracy (%) of the network using different activation functions.

	Eyeball movement			Motor imagery		
Activation function	ReLU	Tanh	Sigmoid	ReLU	Tanh	Sigmoid
Subject A	98.61	89.44	88.89	88.25	78.25	77.50
Subject B	97.50	87.78	88.06	80.75	73.00	72.25
Subject C	96.94	87.22	87.22	80.00	71.25	72.00
Subject D	98.33	88.89	89.17	87.75	80.25	79.75
Subject E	98.06	88.61	88.33	84.50	76.50	76.75
Mean	97.89	88.39	88.33	84.25	78.85	75.65
Std	0.67	0.89	0.76	3.83	3.70	3.40

Table 8
Compare the accuracy (%) of the network using different number of hidden layer units.

	Eyeball movement			Motor imagery		
Number of hidden layer units	3	4	5	3	4	5
Subject A	98.33	98.61	98.33	88.00	88.25	87.75
Subject B	96.94	97.50	97.22	81.00	80.75	80.50
Subject C	96.67	96.94	95.83	80.25	80.00	79.75
Subject D	98.61	98.33	98.89	87.25	87.75	87.50
Subject E	98.06	98.06	97.78	84.25	84.50	84.50
Mean	97.72	97.89	97.61	84.15	84.25	84.00
Std	0.86	0.67	1.17	3.52	3.83	3.77

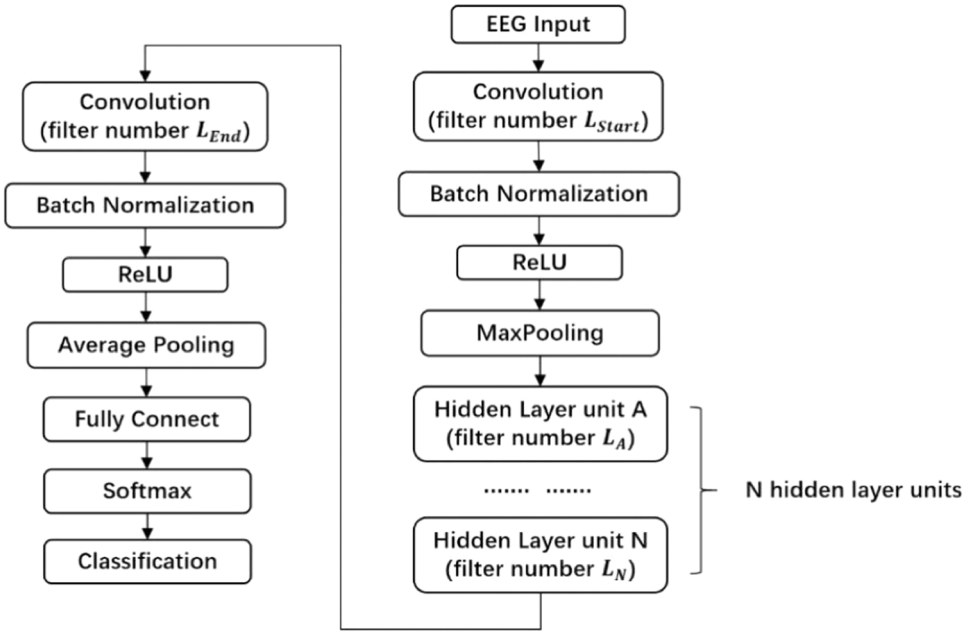

Fig. 24. Whole structure of the classification network.

Table 9
Compare the accuracy (%) of the network using different number of convolutional filters.

Number of convolution filters	Eyeball movement Structure R1 (16 32 64 128 256 512)	Structure R2 (32 64 128 256 512 1024)	Structure R3 (64 128 256 512 1024 2048)	Motor imagery Structure R1 (16 32 64 128 256 512)	Structure R2 (32 64 128 256 512 1024)	Structure R3 (64 128 256 512 1024 2048)
Subject A	97.78	98.61	97.50	87.75	88.25	88.75
Subject B	96.94	97.50	96.11	80.00	80.75	81.00
Subject C	96.11	96.94	97.22	81.25	80.00	80.50
Subject D	98.89	98.33	97.50	86.75	87.75	87.00
Subject E	98.33	98.06	97.78	83.00	84.50	83.50
Mean	97.61	97.89	97.22	83.75	84.25	84.15
Std	1.11	0.67	0.65	3.39	3.83	3.64

Table 10
Compare the performance of proposed control system to other papers' control systems.

	Proposed hybrid BCI real time control system			mHMI system [26]			EEG-EMG HMI system [27]		
	Error times	Accuracy (%)	ITR (bits/min)	Error times	Accuracy (%)	ITR (bits/min)	Error times	Accuracy (%)	ITR (bits/min)
A	2	96.67	82.31	7	90.67	50.44	8	89.16	49.16
B	7	90.91	51.60	11	87.36	43.60	9	86.72	46.19
C	8	89.87	57.31	16	84.31	38.87	14	85.57	38.99
D	3	95.38	76.77	8	89.74	49.76	9	88.89	47.09
E	5	92.75	67.35	9	88.52	49.38	9	87.02	48.78
Mean	5.00	93.12	67.07	10.20	88.12	46.41	9.80	87.47	46.04
Std	2.55	2.89	12.86	3.56	2.47	5.03	2.39	1.52	4.13

EMG-related commands. [27] first extracted the EEG and EMG features using the CSP algorithm. They then calculated the root mean square as the EMG feature vector, calculated the variance, and selected the first three channels as EEG features. Finally, these features were input into the LDA classifier for classification.

We conducted an experiment in which five subjects were given the opportunity to utilize the two systems mentioned above as well as our proposed system to control a machine trolley along a designated route. To assess the performances of the three systems, we compared the number of command errors, accuracy, and information transmission rate (ITR) achieved by each system. Table 10 presents the results of this comparison. ITR is an important indicator for evaluating BCI systems [33]. It was initially used in the communication field to measure the communication and computing speed of systems and was then used to evaluate the efficiency of the BCI control system. The ITR can be calculated as:

$$ITR = \left(\log_2 N_k + A_c \log_2 A_c + (1 - A_c) \log_2 \left[\frac{(1 - A_c)}{N_k - 1} \right] \right) \times \frac{60}{T_a} \tag{19}$$

where A_c is the classification accuracy of each mode, N_k is the number of possible movement intentions, and T_a is the average running time for each command.

From the results, compared to the other subjects, subjects A and D had fewer errors in executing commands. Because these two subjects had experience in BCI-related experiments, they were more likely to concentrate during the process of collecting data and controlling robots. During the experiment, our goal was to allow the subjects to control the robot to reach the final target point by following a designated route. This designated route was generated using a remote controller. During the process of subjects following the designated route, the actual walking path of the robot may deviate because of inaccurate models or subject errors. When a deviation occurs, the system redefines the correction route based on the actual position of the robot and the number of errors increases. The subject continued to follow a new updated route.

From the perspective of system performance, the overall effect of using the proposed method was significantly better than that of the mHMI and EEG-EMG HMI systems, including accuracy and ITR. The ITR can accurately evaluate the amount of information transmitted by BCI systems. It considers three indicators: the number of command categories, target recognition accuracy, and single-target selection time. From the efficiency of the control system, when using the mHMI system and EEG-EMG HMI system, the system often outputs incorrect commands, and the speed of executing each command is lower than that of the proposed hybrid BCI real-time control system.

Through statistical analysis, we calculated the p-values of the proposed hybrid control, mHMI, and EEG-EMG HMI systems. The p-values of accuracy were 0.0187 (comparing the proposed hybrid control system and mHMI system) and 0.0047 (comparing the proposed hybrid control system and EEG-EMG HMI system), which are all less than 0.05, indicating that the accuracy of the proposed hybrid control system is indeed higher than that of the mHMI and EEG-EMG HMI systems. The p-values of ITR were 0.0101 (comparing the proposed hybrid control system and mHMI system) and 0.0083 (comparing the proposed hybrid control system and EEG-EMG HMI system), which are all less than 0.05, indicating that the BCI data transmission performance of the proposed hybrid control system is better than that of the mHMI system and EEG-EMG HMI system. We also compare the time spent and running distance of the three systems in Table 11.

From the perspective of overall running distance and time, our proposed hybrid control system can reach the destination faster and walk shorter distances because our proposed control system has a low error rate and does not require multiple route corrections. Movement time reflects the reaction performance of the EEG system. During the control process, the robot car should constantly change its direction and cooperate with the forward and stop commands to reach the destination. During the process, the robot must move by continuously changing the direction of the arm and controlling the robotic arm. If the time is short, the EEG system can more accurately distinguish between different motor imagery and eyeball movement actions. If it takes only a short time, this means that the EEG system has been able to respond quickly to commands. The distance moved reflects the stability of the EEG system. If the moving distance of the robot car is very long, the robot often makes mistakes during the control process. This implies that the system was unstable. In contrast, if the car can use the shortest distance to reach the destination, the robot can accurately follow the classified commands to find the best way to reach the destination. This also proves that the stability of the proposed control system is better than that of the other two control systems. In addition, we also compare fifteen times of running routes using the three systems. The robotic running routes are shown in Figs. 25, 26, and 27.

From the driving route map, it is evident that the running route and

Table 11
Compare the spending time and running distance of the BCI robot controlled by proposed system to that of the BCI robot controlled by other papers' control systems.

	Proposed hybrid BCI real time control system			mHMI system [26]			EEG-EMG HMI system [27]		
	Second per action	Total Distance (m)	Spending Time (s)	Second per action	Total Distance (m)	Spending Time (s)	Second per action	Total Distance (m)	Spending Time (s)
A	3.03	6.42	181.80	3.72	6.67	279.00	3.68	6.73	305.44
B	3.33	6.71	260.26	3.86	7.11	351.26	3.82	6.92	347.62
C	3.37	6.73	266.23	3.94	7.19	401.88	4.03	7.01	390.91
D	3.09	6.63	200.85	3.69	6.92	287.82	3.77	6.69	305.37
E	3.20	6.43	220.80	3.66	6.42	300.12	3.69	6.54	302.58
Mean	3.20	6.58	225.99	3.77	6.86	324.02	3.80	6.78	330.38
Std	0.15	0.15	36.76	0.12	0.32	51.75	0.14	0.19	38.67

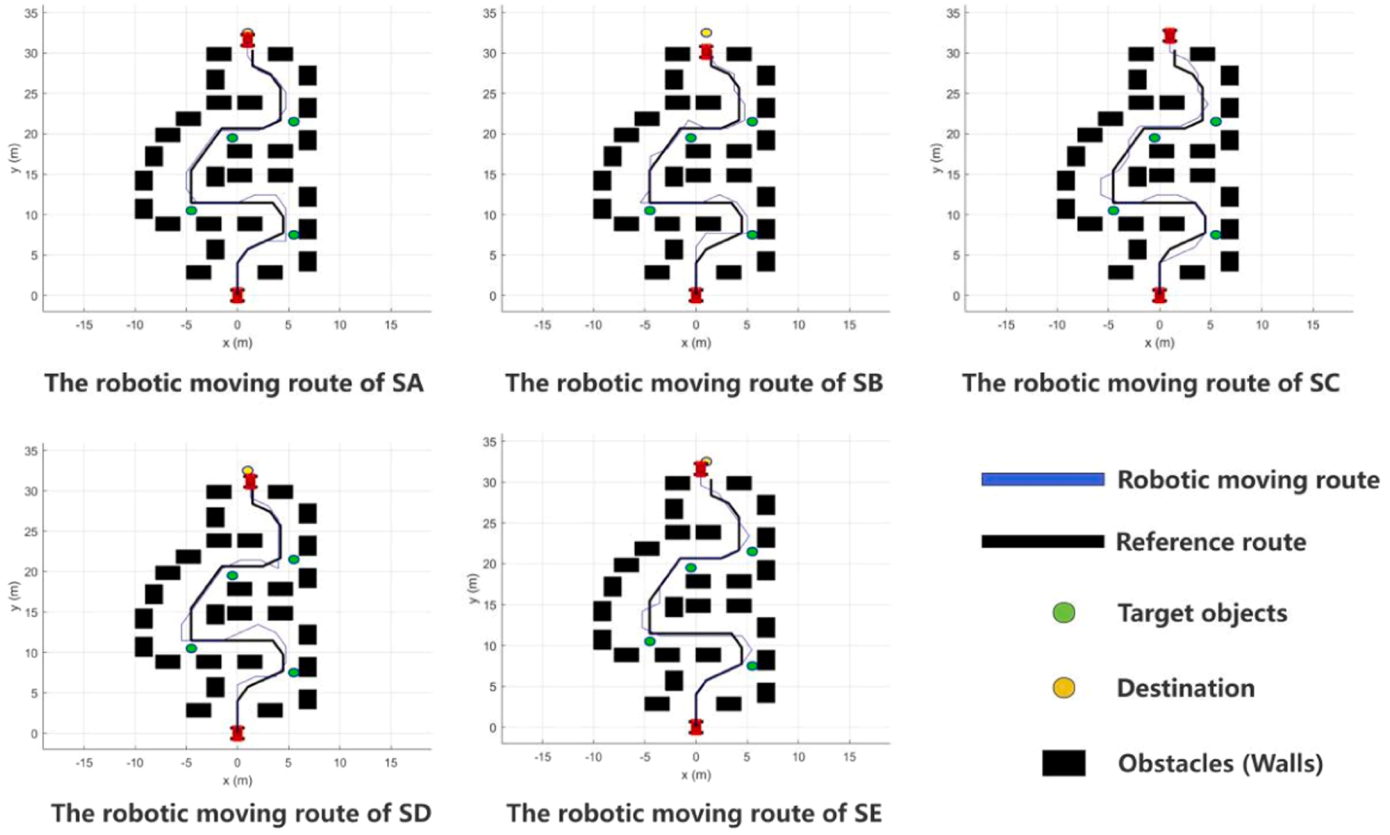

Fig. 25. The robotic running routes using proposed hybrid control system (In each map, the black line is reference route. The blue line is actual robotic moving route. The green point presents the target object. The robot is required to push down the object at each green point. The yellow point presents the final destination. The black square presents the obstacles or walls in the real environment. The x-axis and y-axis represent the horizontal and vertical length (metre) and width (metre) of the map where the robot is located, respectively).

reference route obtained using our proposed method are better fitted, with fewer errors, and the control route is smoother. From the overall route, the positions most prone to errors are at each turn, especially at positions with larger turning angles, which require the subject to alternately send turns and forward commands multiple times. Alternating command switching is more difficult than continuous straight walking. Another error-prone location was at each target object point. When the robot reaches the target object, the subject must continuously cooperate with the steering commands and the direction and extension of the robotic arm to push down the target object. This requires the subject to concentrate on muscle movements and motor imagery. Owing to the extensive experimental experience of subjects A and D, they could complete the task in a shorter time. We can also see from the results in Table 11 that the total spending time for subjects A and D is less than that of other subjects.

When the ideal motion trajectories controlled by the remote controller are compared with the actual motion trajectories controlled by the proposed control system, we can see that both the ideal and actual trajectories are able to avoid obstacles and reach the destination. Although there is some deviation when using the control system, it is not large. It is noteworthy that the subjects may feel tired after a period of concentration on handling the BCI real-time control system and controlling the robot car, which could also affect the results. The quality of the test results is also related to the subject's operational skills. Overall, the BCI real-time control system was accurate, stable, and reliable. This has the potential to be adopted in practical BCI control applications.

5. Conclusions

In this study, an EEG state recognition model is proposed, which can be used to determine the EEG signal states. In this model, an LSTM-CNN structure is introduced to extract both spatial and time sequence features. An actor-critic-based decision-making model is proposed to predict the desired action based on signal state probabilities. The brain

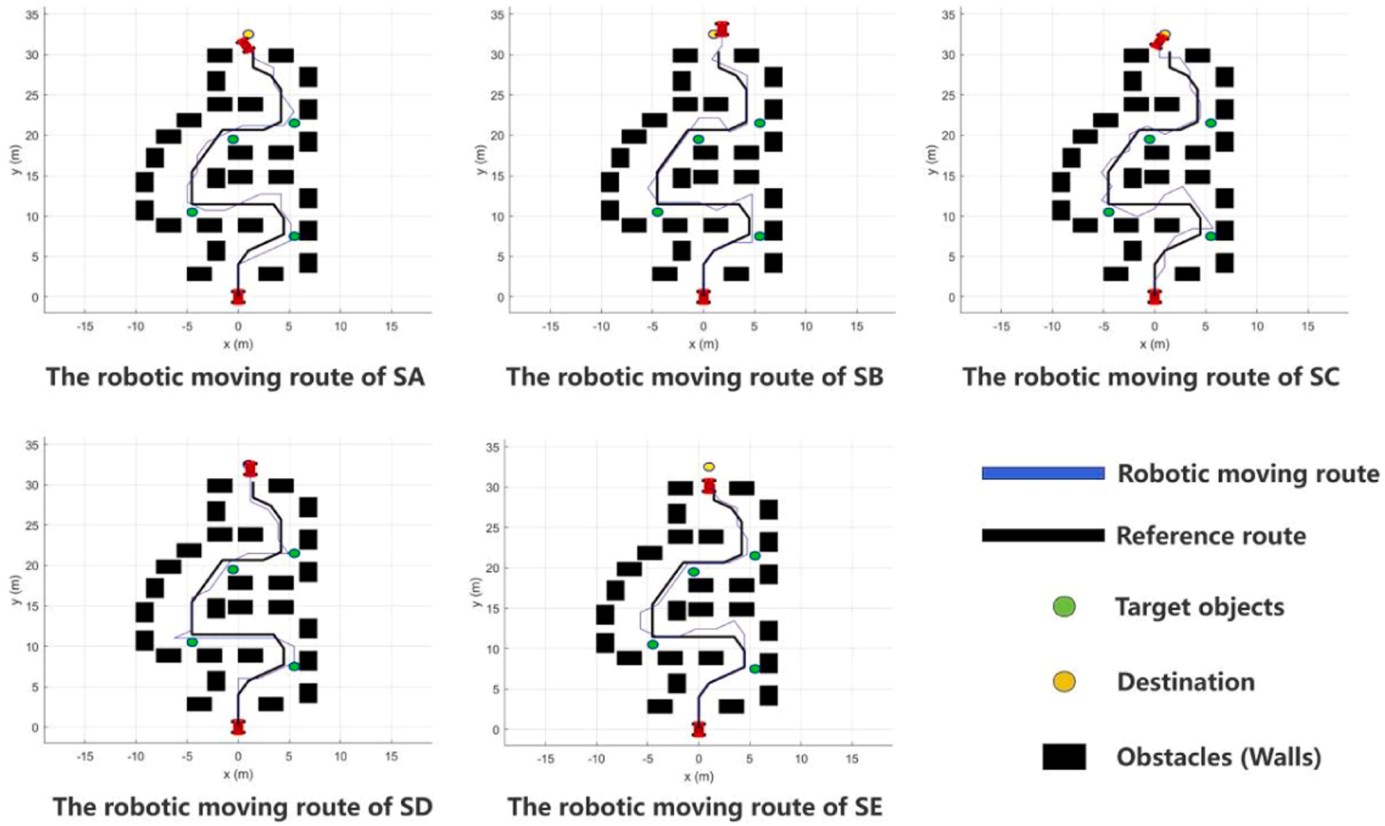

Fig. 26. The robotic running routes using mHMI system.

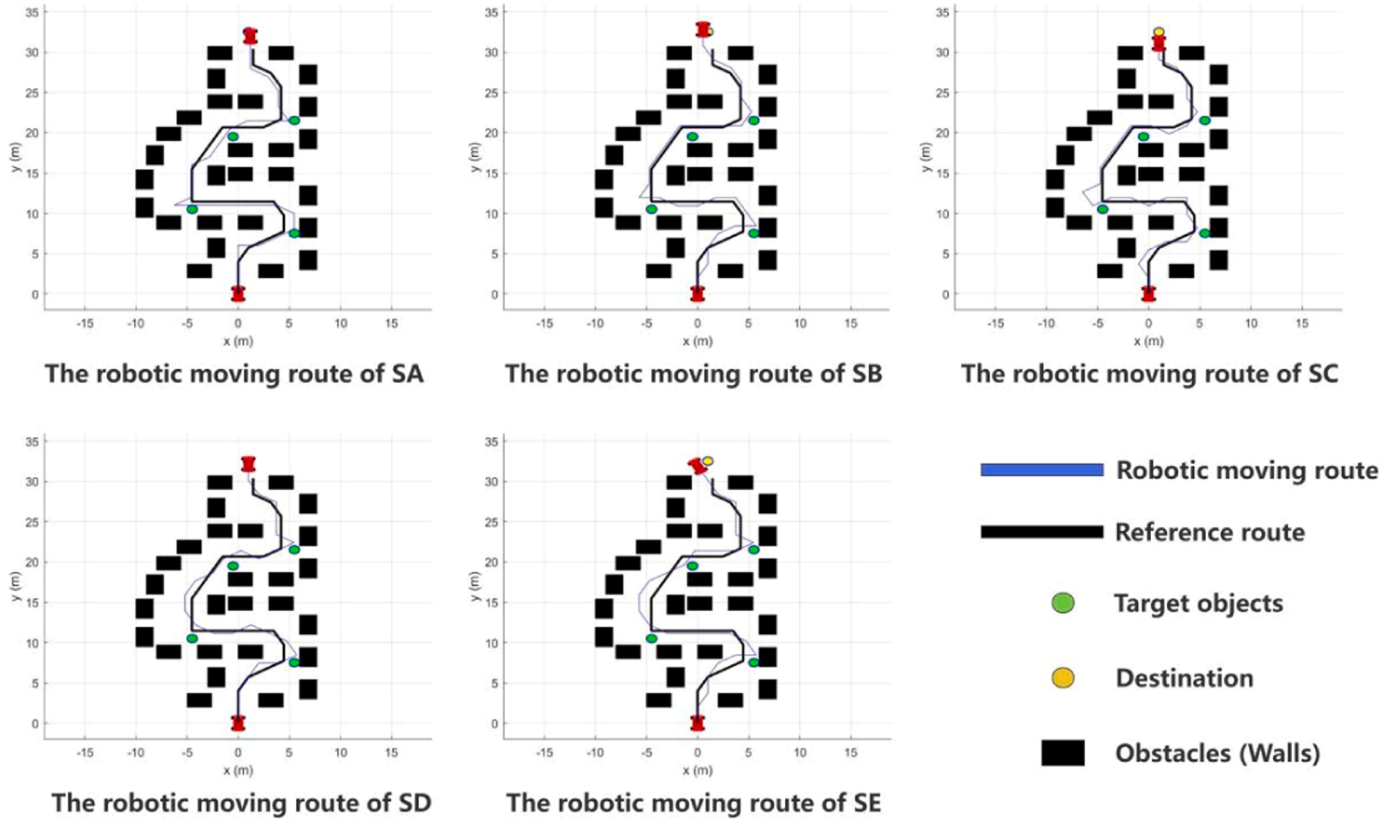

Fig. 27. The robotic running routes using EEG-EMG HMI system.

signals collected from three subjects were used to train and test the proposed models. Experimental results were used to compare the two proposed models. The combined EEG state recognition model and actor-critic-based decision-making model achieved the best performance. In this study, using the proposed method, the mean accuracy was 95.34% and the mean kappa value was 0.9289. Compared to the traditional convolutional neural network, the mean accuracy improved by 3.75%, and the mean kappa value improved by 0.0591. Compared with the EEG state recognition model, the mean accuracy improved by 2.08%, and the mean kappa value improved by 0.0311.

In addition, we propose a hybrid BCI-based real-time control system that is used to control a BCI robot car. This system includes two subsystems: i) a data transmission system and ii) an EEG dynamic classification system. In the data transmission system, a data acquisition server from the EEG device to other software was built, and a data processing and reception model was included to achieve online EEG processing and analysis. In the EEG dynamic classification system, six signal analysis models were constructed to classify the BCI commands. In the experiment, we trained an accurate and reliable system that can be used to control a BCI robot in a real environment. In the experimental results, the offline testing accuracy was 87.20%. The mean online control accuracy was 93.12%, and the mean ITR was 67.07 bits/min. These results prove that the proposed BCI control system is better than state-of-the-art systems. Compared to the multimodal human–machine interface system, the mean accuracy improved by 5%, and the information transmission rate was improved by 20.66 bits/min. Compared with the EEG–EMG Human–Machine Interface-based control system, the mean accuracy improved by 5.65%, and the information transmission rate was improved by 21.03 bits/min.

In the future, BCI-related medical devices such as EEG-based wheelchairs and BCI-based robotic arms can be developed based on the proposed system. This system can be used in motor rehabilitation training. In addition, the proposed approach could be extended to other industries and applied to different working environments, for example, applying EEG- or BCI-based robotic applications in a construction or engineering environment to support construction activities and material handling on-site.

CRediT authorship contribution statement

Sai Ho Ling: Conceptualization, Investigation, Methodology, Project administration, Supervision, Validation, Writing – review & editing. **Yang An:** Conceptualization, Data curation, Formal analysis, Methodology, Software, Writing – original draft. **Johnny Wong:** Funding acquisition, Validation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

Data availability

The data that has been used is confidential.

References

[1] K.A.M. Ginis, H.P. van der Ploeg, C. Foster, B. Lai, C.B. McBride, K. Ng, M. Pratt, C. H. Shirazipour, B. Smith, P.M. Vásquez, Participation of people living with disabilities in physical activity: a global perspective, Lancet 398 (2021) 443–455.

[2] S.A. Mansi, I. Pigliautile, C. Porcaro, A.L. Pisello, M. Arnesano, Application of wearable EEG sensors for indoor thermal comfort measurements, Acta IMEKO 10 (2021) 214–220.

[3] A. Kaur, Wheelchair control for disabled patients using EMG/EOG based human machine interface: a review, J. Med Eng. Technol. 45 (2021) 61–74.

[4] C. He, Y.Y. Chen, C.R. Phang, C. Stevenson, I.P. Chen, T.P. Jung, L.W. Ko, Diversity and Suitability of the State-of-the-Art Wearable and Wireless EEG Systems Review, IEEE J. Biomed. Health Inform. (2023) 1–14.

[5] S. Luo, Q. Meng, S. Li, H. Yu, Research of intent recognition in rehabilitation robots: a systematic review, Disabil. Rehabil.: Assist. Technol. (2023) 1–12.

[6] E. Kim, J. Shin, Y. Kwon, B. Park, EMG-Based Dynamic Hand Gesture Recognition Using Edge AI for Human–Robot Interaction, in: Electronics, 2023.

[7] M. Sasaki, K. Matsushita, M.I. Rusyidi, P.W. Laksono, J. Muguro, M.S.A. bin Suhaimi, W. Njeri, Robot control systems using bio-potential signals, in: AIP Conference Proceedings, AIP Publishing LLC, 2020, pp. 020008.

[8] H. Yang, J. Wan, Y. Jin, X. Yu, Y. Fang, EEG- and EMG-Driven Poststroke Rehabilitation: a review, IEEE Sens. J. 22 (2022) 23649–23660.

[9] A.J. Moshayedi, N.M.I. Uddin, X. Zhang, M. Emadi, Andani, Exploring the role of robotics in Alzheimer's disease care: innovative methods and applications, Robot. Intell. Autom. 43 (2023) 669–690.

[10] A.J. Moshayedi, S.K. Sambo, A. Kolahdooz, Design and development of cost-effective exergames for activity incrementation, 2nd Int. Conf. Consum. Electron. Comput. Eng. (ICCECE) 2022 (2022) 133–137.

[11] A. Jafarifarmand, M.A. Badamchizadeh, Real-time multiclass motor imagery brain-computer interface by modified common spatial patterns and adaptive neuro-fuzzy classifier, Biomed. Signal Process. Control 57 (2020) 101749.

[12] Z. Tayeb, J. Fedjaev, N. Ghaboosi, C. Richter, L. Everding, X. Qu, Y. Wu, G. Cheng, J. Conradt, Validating deep neural networks for online decoding of motor imagery movements from EEG signals, Sensors 19 (2019).

[13] H. Gou, Y. Piao, J. Ren, Q. Zhao, Y. Chen, C. Liu, W. Hong, X. Zhang, A solution to supervised motor imagery task in the BCI Controlled Robot Contest in World Robot Contest, Brain Sci. Adv. 8 (2022) 153–161.

[14] Ty Mwata-Velu, J. Ruiz-Pinales, H. Rostro-Gonzalez, M.A. Ibarra-Manzano, J. M. Cruz-Duarte, J.G. Avina-Cervantes, Motor imagery classification based on a recurrent-convolutional architecture to control a hexapod robot, Mathematics 9 (2021).

[15] X. Shen, X. Wang, S. Lu, Z. Li, W. Shao, Y. Wu, Research on the real-time control system of lower-limb gait movement based on motor imagery and central pattern generator, Biomed. Signal Process. Control 71 (2022) 102803.

[16] M. Tariq, P.M. Trivailo, M. Simic, Motor imagery based EEG features visualization for BCI applications, Procedia Comput. Sci. 126 (2018) 1936–1944.

[17] S.L. Wu, Y.T. Liu, T.Y. Hsieh, Y.Y. Lin, C.Y. Chen, C.H. Chuang, C.T. Lin, Fuzzy Integral with particle swarm optimization for a motor-imagery-based brain–computer interface, IEEE Trans. Fuzzy Syst. 25 (2017) 21–28.

[18] J. Choi, K.T. Kim, J.H. Jeong, L. Kim, S.J. Lee, H. Kim, Developing a motor imagery-based real-time asynchronous hybrid bci controller for a lower-limb exoskeleton, Sensors 20 (2020).

[19] Y. Wang, B. Hong, X. Gao, S. Gao, Implementation of a brain-computer interface based on three states of motor imagery, 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. 2007 (2007) 5059–5062.

[20] A. Ak, V. Topuz, I. Midi, Motor imagery EEG signal classification using image processing technique over GoogLeNet deep learning algorithm for controlling the robot manipulator, Biomed. Signal Process. Control 72 (2022) 103295.

[21] H. Li, H. Ji, J. Yu, J. Li, L. Jin, L. Liu, Z. Bai, C. Ye, A sequential learning model with GNN for EEG-EMG-based stroke rehabilitation BCI, Front. Neurosci. 17 (2023).

[22] G. Canal, Y. Diaz-Mercado, M. Egerstedt, C. Rozell, A. Low-Complexity, Brain–Computer Interface for High-Complexity Robot Swarm Control, IEEE Trans. Neural Syst. Rehabil. Eng. 31 (2023) 1816–1825.

[23] B. Xu, D. Liu, M. Xue, M. Miao, C. Hu, A. Song, Continuous shared control of a mobile robot with brain–computer interface and autonomous navigation for daily assistance, Comput. Struct. Biotechnol. J. 22 (2023) 3–16.

[24] X. Wang, H.T. Chen, Y.K. Wang, C.T. Lin, Implicit Robot Control Using Error-Related Potential-Based Brain–Computer, Interface, IEEE Trans. Cogn. Dev. Syst. 15 (2023) 198–209.

[25] K. Karas, L. Pozzi, A. Pedrocchi, F. Braghin, L. Roveda, Brain-computer interface for robot control with eye artifacts for assistive applications, Sci. Rep. 13 (2023) 17512.

[26] J. Zhang, B. Wang, C. Zhang, Y. Xiao, M.Y. Wang, An EEG/EMG/EOG-Based Multimodal Human-Machine Interface to Real-Time Control of a Soft Robot Hand, Front. Neurorobotics 13 (2019).

[27] S.Y. Gordleeva, S.A. Lobov, N.A. Grigorev, A.O. Savosenkov, M.O. Shamshin, M. V. Lukoyanov, M.A. Khoruzhko, V.B. Kazantsev, Real-Time EEG–EMG human–machine interface-based control system for a lower-limb exoskeleton, IEEE Access 8 (2020) 84070–84081.

[28] J.H. Cho, J.H. Jeong, S.W. Lee, NeuroGrasp: real-time eeg classification of high-level motor imagery tasks using a dual-stage deep learning framework, IEEE Trans. Cybern. 52 (2022) 13279–13292.

[29] S. Abdullah, M.A. Khan, M. Serpelloni, E. Sardini, Hybrid EEG-EMG based brain computer interface (BCI) system for real-time robotic arm control, Adv. Mater. Lett. 10 (2019) 35–40.

[30] Y. An, H.K. Lam, S.H. Ling, Auto-Denoising for EEG signals using generative adversarial network, Sensors 22 (2022) 1750.

[31] Y. An, H.K. Lam, S.H. Ling, Multi-classification for EEG motor imagery signals using data evaluation-based auto-selected regularized FBCSP and convolutional neural network, Neural Comput. Appl. 35 (2023) 12001–12027.

[32] g.t.m.e. GmbH, Unicorn Hybrid Black, in.

[33] J.R. Wolpaw, N. Birbaumer, W.J. Heetderks, D.J. McFarland, P.H. Peckham, G. Schalk, E. Donchin, L.A. Quatrano, C.J. Robinson, T.M. Vaughan, Brain-computer interface technology: a review of the first international meeting, IEEE Trans. Rehabil. Eng. 8 (2000) 164–173.