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

Yang An, Johnny Wong, Sai Ho Ling *Senior Member, IEEE*

Abstract—Electroencephalogram (EEG) based brain-computer interfaces (BCI) has been considered one of the prevailing non-invasive method to collect human biomedical signals through attaching electrodes to the scalp. However, it is difficult to detect and use these signals for controlling the online BCI robot in a real environment due to the environmental noise. In this study, a novel state recognition model is proposed to determine and improve the EEG signal states. First, a Long Short-Term Memory Convolutional Neural Network (LSTM-CNN) is designed to extract the EEG features along the time sequence. During this process, errors which are caused by mind randomness or external environmental factors may be generated. Thus, an actor-critic based decision-making model is proposed to correct these errors. The model consists of two networks that can be used to predict the final signal state based on both current signal state probability and past signal state probabilities. Then, a hybrid BCI real-time control system application is proposed to control a BCI robot. The Unicorn Hybrid Black EEG device is used to acquire brain signals. The data transmission system is constructed in OpenViBE to transfer data. The EEG classification system is built to classify BCI commands. In the experiment, EEG data from three subjects was collected, to train and test the performance and reliability of the proposed control system. The system records the robot's spending time, moving distance, and the number of objects pushing down. Experimental results are given to show the feasibility of the real-time control system. Compared with 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, offline testing accuracy can achieve 85.22%. When we apply the proposed system to control a BCI robot in a real environment, the best controlling time is 187.4 seconds, and the best running distance is 6.8 meters. This shows that the proposed hybrid BCI real-time control system demonstrated a higher reliability, which can be used in practical BCI control applications.

Index Terms— **Brain Computer Interface, Electroencephalogram, EEG robot**

1. Introduction

The number of people with disabilities in the world exceeds two billion. Among these disabled, 75 million people are physically disabled [1] due to stroke, car accident, work accident, etc. With the need of enhancing quality of life and promoting mobility for disability, along with advancement in neuroscientific technologies, the development of auto-controlled wheelchair based on brain-computer interface (BCI) has attracted growing attention. Brain control is an ideal control method because it helps control machines anywhere and anytime by imagining mental activities. As a result, the BCI has become an important technology to help the disabled. One of the main non-implanted BCIs is electroencephalogram (EEG), the electrical activity of the brain (i.e. brain signals) can be collected from the wearable EEG device [2]. Compared to an implantable BCI, EEG wearable device is ethical and less invasive. In recent years, there have been remarkable developments of the EEG-based BCI applications, such as brain-controlled wheelchairs, brain-controlled arms, and EEG based auxiliary machines. With the advancement in artificial intelligence and EEG classification and analyzing methods, it is possible to develop EEG based applications which can help disabled people complete simple daily tasks.

In recent years, there has been a surge of interest in the developing offline EEG classification methods. For examples, [3] tested the public data set BCI4-2a to build an offline classification model. They used Joint Approximate Diagonalization (JAD) to extend the traditional Common Spatial Pattern (CSP) into a multi-class CSP. This method can reduce uncertainties caused by artifacts. To avoid EEG information loss they also proposed a self-regulated supervised Gaussian fuzzy adaptive system as the classifier. The learning rate can also be automatically adjusted by a coefficient. [4] 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 also collected their own data and increased the amount of training data using the augmentation method. When they controlled the robotic arm in real-time, they set a threshold for the classifier which 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.

Moreover, there has been substantial progress in the development of EEG-based online classification systems to construct BCI real-time control applications. For instances, [5] and [6] proposed real-time control methods to classify two types of EEG signals. They employed OpenViBE software to process and transfer data in real time. [5] used Fast Fourier Transform (FFT) to calculate the frequency spectrum of EEG signals and used the energy changes of Mu and Beta segments as classification features where Mu and Beta are motor-related frequency bands. [6] created diverse ensemble classifiers using the sub-band common spatial pattern. They first divided the signal into multiple sub-signals using filters and performed the traditional CSP feature extraction method on each sub-signal. Then, they used a fuzzy fusion technique, which addressed individual differences in EEG in a noisy environment. Particle Swarm Optimization (PSO) was also employed to optimize fuzzy integrals. Finally, Linear Discriminant Analysis (LDA) was used as the classifier to classify the fuzzified features. In addition, [7] and [8] developed EEG-based real-time control applications. [7] 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 seconds. They used the hard processor system, which includes efficient data processing hardware and memory. The transfer rate is high, so their method can limit the delay time to within ten milliseconds. Thus, their system can achieve an efficient real-time classification application. [8] constructed a signal processing model and a CSP feature extraction model in OpenViBE software. Then, the spatially transformed signal energy was calculated, and the average value of the signal was input into LDA for classification. They also used a Virtual Reality Peripheral Network (VRPN), which can efficiently transmit data and does not require extra support from other software. Compared to [5] and [6], the data transmission technique developed by [7] and [8] are more efficient.

While the developed methods discussed above focusing on EEG binary classification, there have been also a growing number of research focusing process multi-class tasks in real time. For examples, [9] and [10] implemented three classes of motor imagery-based control methods. [9] used Filter Bank Common Spatial Pattern (FBCSP) as the feature extraction method, Mutual Information Based Best Individual Feature (MIBIF) as the feature selection method, and 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 seconds. Because motor imagery requires a certain amount of imagining time, this sliding window-based segmentation method is more suitable for analyzing EEG signals. Similarly, [10] used a moving window to fetch different data segments for processing. They proposed a two-stage training system to train classification models. They first used offline data to build a model and then used the model to apply offline classification and online classification tests, respectively. Then, they used CSP as the feature extraction method and extended the model to a multi-class model, using the One VS One (OVO) strategy. Finally, LDA was used to classify the extracted features. [11] identified four types of EEG signals in real time. They computed three time-domain parameters of three channels as features, and used a neural network as a classifier. However, the data recorded for each trial was 20 seconds, and the data segment of 10 to 20 seconds was taken as training data. It takes a long time to perform the EEG task in this way, so this method gives low control efficiency.

The past few years has also produced a significant body of research in other real-time control applications, such as the use of BCI system on improving motor imagination ability or concentration ability. [12] proposed an EEG-based real-time control model to control an inverted pendulum. They used CSP to extract fifteen channels of features and used LDA as the classifier. Then they fed the recognition output into the fuzzy system to obtain the final control command. This trained model was applied, to improve the subject's responsiveness. [13] applied the spatial filters on detecting six channels of Event-Related Desynchronization (ERD) features and employed Support Vector Machine (SVM) as a classifier. They also used Electromyogram (EMG) to control the stop command. Feedback on the magnitude of control was given each time that the subject performed the task. Then, they used the feedback as the reward to train the control model again. As training times increased, the recognition error rate of EEG signals gradually decreased. By the end of the experiment, subjects could accurately control wheelchair movement in real time. In addition, [14] detected the correlated error potentials, and this feature could be used as feedback to correct the output of imaginary movements. Similarly, [15] developed a CSP spatial filter and then used the CSP filtered features as feedback to improve the subject's imagination. These models can all update model parameters based on actual feedback and improve their performance.

However, despite these research efforts, there is a dearth of studies that aim to achieve a high-performance BCI control system. While existing literature and studies developed mechanism to control the robot using EEG signals, what seems to be lacking is accurate and seamless classification and execution of multiple EEG commands. The proposed control systems in many existing studies can only recognize less than four BCI commands. In addition, most methods can be only used for offline control. Thus, their systems cannot be applied in any real environment. In addition, when people are performing brain tasks, there are energy changes, but these changes are weak and difficult to detect. Performing brain tasks is a process that cannot be finished immediately. During this process, the brain states keep changing, and the changes generate gradually. For example, when the subject is performing motor imagery tasks, the brain signals change from a rest state to a motor imagery state, and the energy in related frequency bands decreases gradually. The control methods proposed in most papers are direct control, which means the result of each recognition is directly used as the final control command. However, in practical applications, the user may be affected by the external environment. Thus, they may generate a control command unconsciously, and this command is not the desired result for the user. Therefore, by using this control approach, the output commands do not follow user's expectations. As a result, detecting the signal state and predicting the current robotic action is a difficult task.

In view of this, a hybrid BCI real time control system estimation technique is proposed in this study to deal with the problems discussed above. The contributions of this study lie in two aspects:

- The hybrid BCI real-time control system developed in this study helps process multiple BCI commands in real time, promoting higher robotic precise and more reliable control in real-time. To implement the hybrid BCI real-time control system, this study develops two components. Firstly, an EEG-based dynamic classification system is proposed to tackle EEG multi-classification tasks. Secondly, a data transmission system is proposed to achieve data communication. By applying the proposed control system, the BCI robot can be controlled by seven BCI commands in real time. The performance of the proposed control method is closer to manual control.
- The actor-critic based decision-making model proposed in this study can learn the user's control habits. The model not only considers the current signal state but also the previous signal states. By continuing to learn the changes in the user's brain states and the corresponding labels, the model can update the parameters and correct the final control command. By using the proposed method, unexpected actions caused by a human's unconscious brain activities can be avoided. It can also reduce the error caused by external influences. Compared to traditional classifiers, the proposed method can better predict reasonable robotic actions.

2. Methodology

2.1 Block diagram of system

The proposed hybrid BCI real-time control system is presented in this section. We use a data transmission system and an EEG dynamic classification system to construct a hybrid BCI real-time control system, which is used to control a BCI robot car in this study. The block diagram of the system is shown in Figure 1. The EEG acquisition device is connected to the computer by a USB Bluetooth adapter. The data transmission system obtains the EEG signal from the Unicorn EEG device by connecting to the computer's I/O port. After processing the online signals, the data was sent to the EEG-based dynamic classification system to classify the commands. After that, the output commands are sent to the BCI robot through WebSocket. Finally, the commands are executed by the BCI robot.

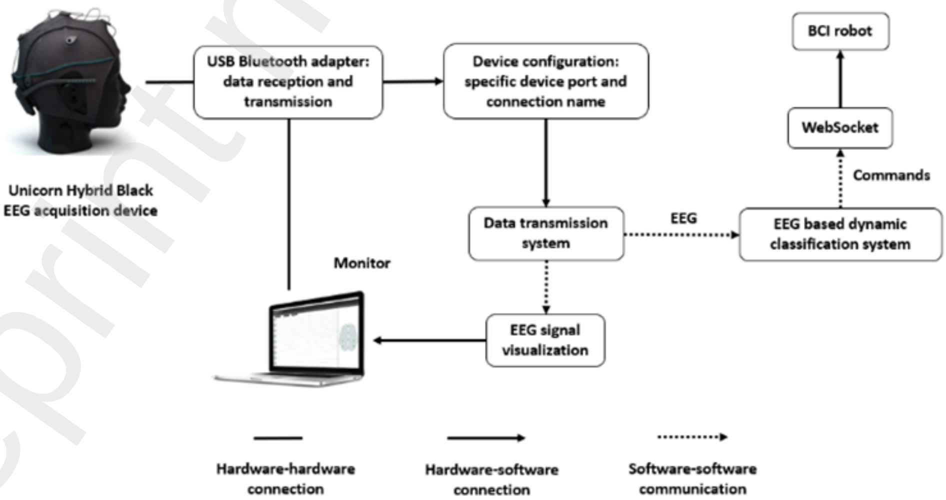

Fig. 1. Block diagram of hybrid BCI real time control system

2.2 Data transmission system

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

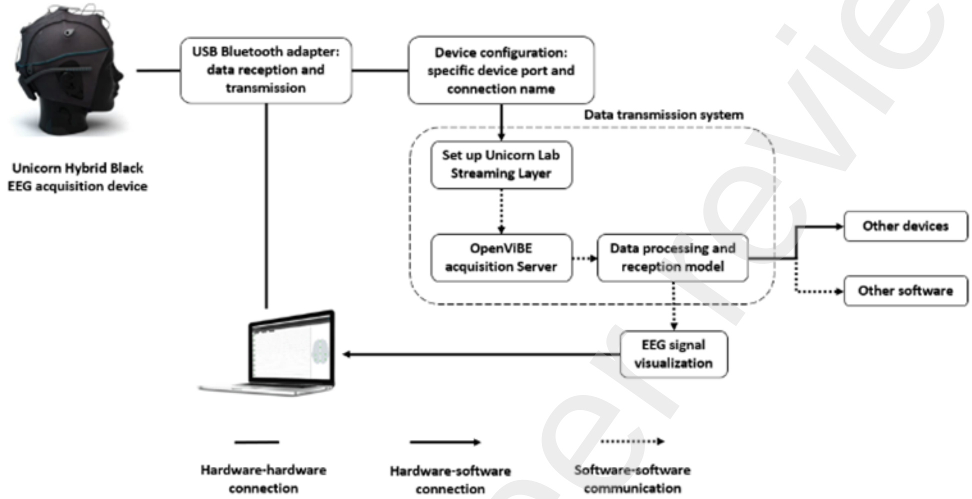

Fig. 2. Structure of data transmission system

OpenViBE is free, open-source software for real-time neurosciences. 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 pre-configured and demo programs which are ready for use. Some function boxes are designed. The EEG data processing and reception model is shown in Figure 3. The acquisition client box is used to receive the EEG data. A channel selector box is used to select effective channels. Bandpass and band stop boxes are used to filter the EEG signal. The GDF file writer box is used to save the EEG signals in GDF format.

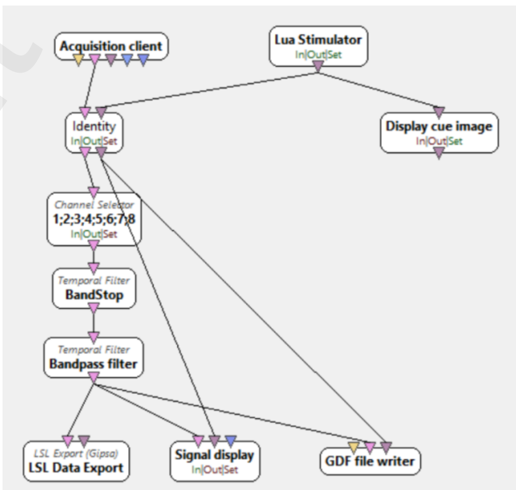

Fig. 3. EEG data processing and reception model in OpenViBE

The Lua stimulator box is used to guide subjects to take corresponding actions. There are 11 stimulation labels, including 1) left-hand motor imagery, 2) right-hand motor imagery, 3) eyeball move to left, 4) eyeball move to right, 5) eyeball move 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 are pre-set. All the prompt labels are randomly presented. The subjects perform the corresponding tasks according to the prompts.

In this system, the lab streaming layer and OpenViBE's 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 measurement time series in research experiments. It handles the networking, time-synchronization and real-time access, as well as, optionally the centralized collection, viewing, and disk recording of the data. After the signal is processed in the OpenViBE designer, the data is exported by the LSL data export box.

The EEG device transmits the collected EEG data to the laptop through the Bluetooth adapter. First, it matches the device port and device name in the laptop and creates a lab streaming layer. The EEG data is sent to the OpenViBE software through the lab streaming layer. OpenViBE receives data through an acquisition server. After that, the data processing and reception model is established in the OpenViBE designer, including the data visualization module and the MATLAB connection module. The visualization module can directly plot the EEG signal in real time. The MATLAB connection module is used to send the processed data to the MATLAB software. The EEG based dynamic classification system is 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 part of the hybrid BCI control system, which is shown in Figure 4. The EEG data is received from the data transmission system. The proposed classification system is used to classify the BCI commands, which are further used to control the BCI robot. The system proposed in this paper can recognize seven different commands. The system includes five models, which are the pre-processing model, denoising model, EEG state recognition model, motor imagery classification model, and eyeball movement classification model. When the EEG signal arrives, it is pre-processed and denoised by the pre-processing model and the denoising model. Then, the processed signal is input into the EEG state recognition model to calculate the probabilities of motor imagery, eyeball movement, EMG and rest state signals. The actor-critic based decision-making model is developed to integrate all the probabilities of the commands and determine the signal state. Eventually, the motor imagery classification model and the eyeball movement classification model are further used to analyze the signal to obtain the final command, which is used to control the BCI robot.

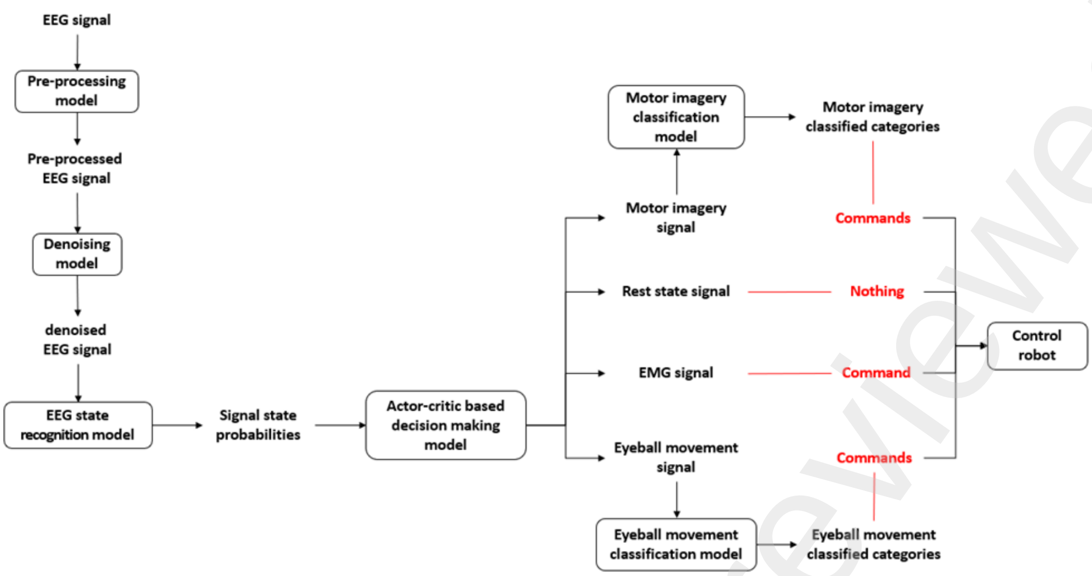

Fig. 4. Structure of EEG-based dynamic classification system

2.3.1. Pre-processing and denoising model

In the pre-processing model, the raw EEG data is filtered by a 1 to 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 which extracts the raw EEG data minus the mean value of all the channels. The function of this step is to correct the reference electrode to close to zero. Finally, bad signal segments are removed. Since the equipment may be affected by noise, the signal energy fluctuates significantly. We define such data as low-quality data, and we remove these data by setting the energy threshold. The output of this model is the pre-processed EEG signal.

The pre-processed signal is input into the denoising model for noise removal. The EEG signals can be easily affected by unexpected noises such as eye blinking, heartbeat, etc. These noisy signals may generate higher energy than the original EEG signal. This artifact may affect the performance of EEG signal classification. Thus, we construct an automagical denoising method to remove the noise from pre-processed EEG signals. The details can be found in [16].

2.3.2 EEG state recognition model

2.3.2.1 Overview of model

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

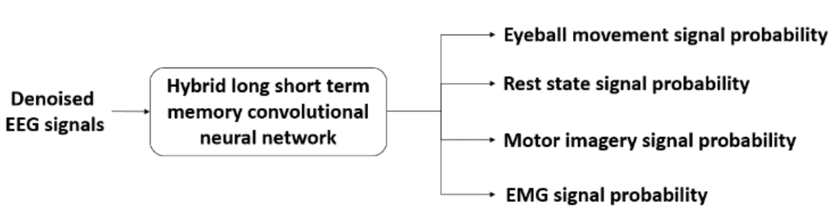

Fig. 5. Block diagram of EEG state recognition model

2.3.2.2 Structure of network

The network structure and network parameters are shown in Figure 6 and Table I. Firstly, 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 are extracted from the signal, and the features are compressed. After that, 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 time period and then rearranged according to the time sequence. Finally, the extracted feature compressed signal is input into the LSTM layer to calculate the signal state probabilities.

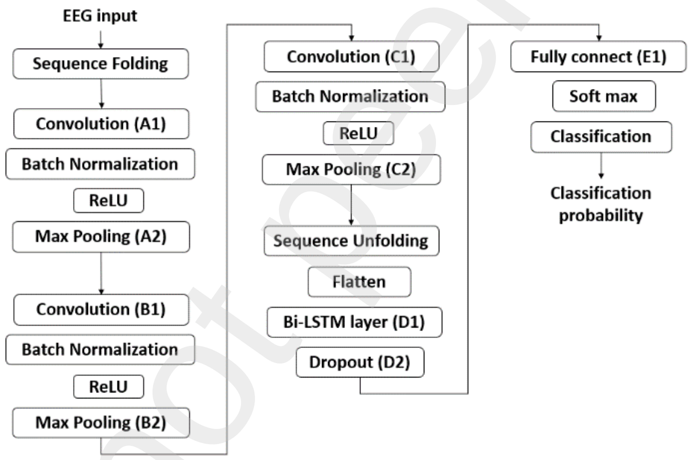

Fig. 6. Hybrid long short-term memory convolutional network (LSTM-CNN) structure used in EEG state recognition model

TABLE I. Long short term memory convolutional network parameters

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

2.3.3 Actor-critic based decision making model

2.3.3.1 Overview of model

The block diagram of the decision-making model is shown in Figure 7. The purpose of this model is to predict the states based on the previous recognition probabilities. The recognition probabilities are input into two models. The first is to make a decision based on the maximum probability. The second is to use an actor network to predict the desired action. The critic network is used to evaluate whether the output from the actor network is good or not, responding to the EEG recognition probabilities. Thus, the critic network can be used to evaluate the outputs from the two models. Finally, the action with the higher score is used as the final action.

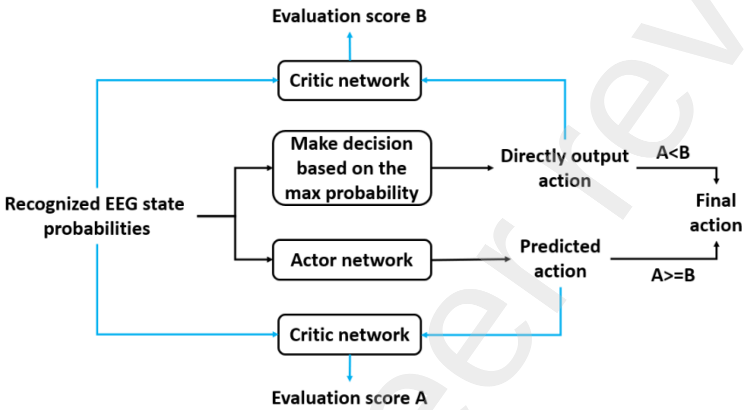

Fig. 7. Block diagram of the decision making model

2.3.3.2 Structure of actor and critic networks

The structure of the actor network is shown in Figure 8. The input is the probabilities of the current signal state and past signal states. The output is the predicting action. There are three hidden layers in this network. After each fully connected layer, the ReLU layer is used as the activation function.

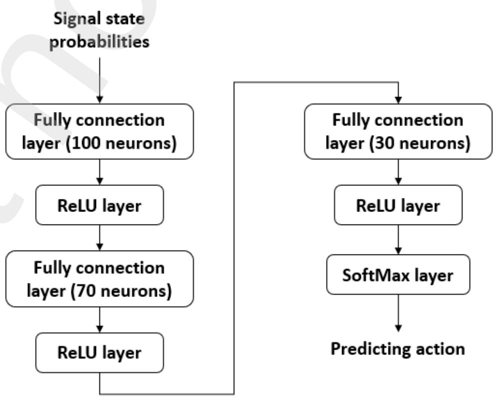

Fig. 8. Structure of actor network

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

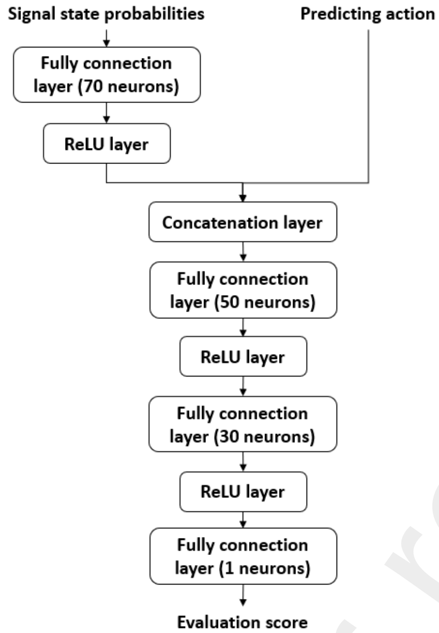

Fig. 9. Structure of critic network

2.3.3.3 Training actor and critic networks

The structure of the two networks does not need to be very complicated. The key is to design the loss functions. The training process is shown in Figure 10. For the critic network, our purpose is to judge whether the output of the actor is good. Thus, we use mean square error as the loss function. We input signal state probabilities and correct actions into the critic network to calculate the score and then calculate the mean square error between the score and 1. After that, we input signal state probabilities and random actions into the critic network to calculate the score and then calculate the mean square error between the score and 0.

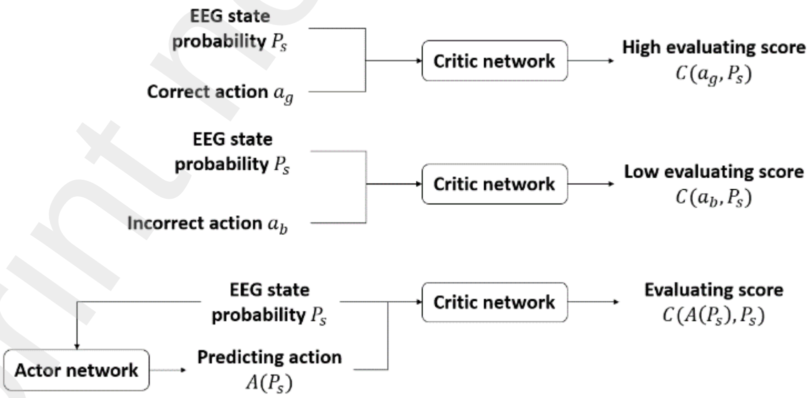

Fig. 10. The process of training the actor network and critic network

For the actor network, our goal is to predict the current action based on the previous and current signal states. Thus, there are two conditions when training the actor network. First, the predicted action should be close to the ideal action, so we use cross entropy as the first loss term. Second, the output actions continue to be input into the critic network to obtain a score. For the actor network, it hopes that the predicting action is a good action. Thus, the score should be close to 1. We calculate the mean square error between the output of the critic network and 1. These two loss terms are added together as the loss function of the actor network.

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

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

where N_m is batch size; P_s is the EEG signal state probability vector; Y_s is the EEG signal action label; a_g is correct predicted action; a_b is incorrect predicted action. The network is then trained using the ADAM updating method. After training, the critic network has the ability to recognize good actions and bad actions. The actor network has the ability to predict actions.

2.3.4 EEG task classification model

2.3.4.1 Overview of model

There are two classification models used to classify specific tasks. The first model is used to classify motor imagery tasks. The second model is used to classify eyeball movement tasks. The block diagram of the classification model is presented in Figure 11. The processed data is first filtered using multiple sets of bandpass filters. For the motor imagery task, we need to stack all of the frequency filtered sub-signals to obtain the combined frequency filtered signals which are used as the target subject EEG data S_c. For the eyeball movement task, the frequency filtered sub-signals are used as the target subject EEG data S_c. Then, the auto-selected regularized common spatial pattern algorithm is applied to the target EEG data and other subjects' EEG data to obtain the Spatial transformed data Z_c. After that, if the task is motor imagery, mutual information based best individual feature selection is applied to select the effective spatial features. Otherwise, if the task is eyeball movement, variance difference based best individual feature selection is applied, and the obtained spatial vectors should be stacked along the channel dimension. Eventually, the final spatial features are input into a convolutional neural network to classify.

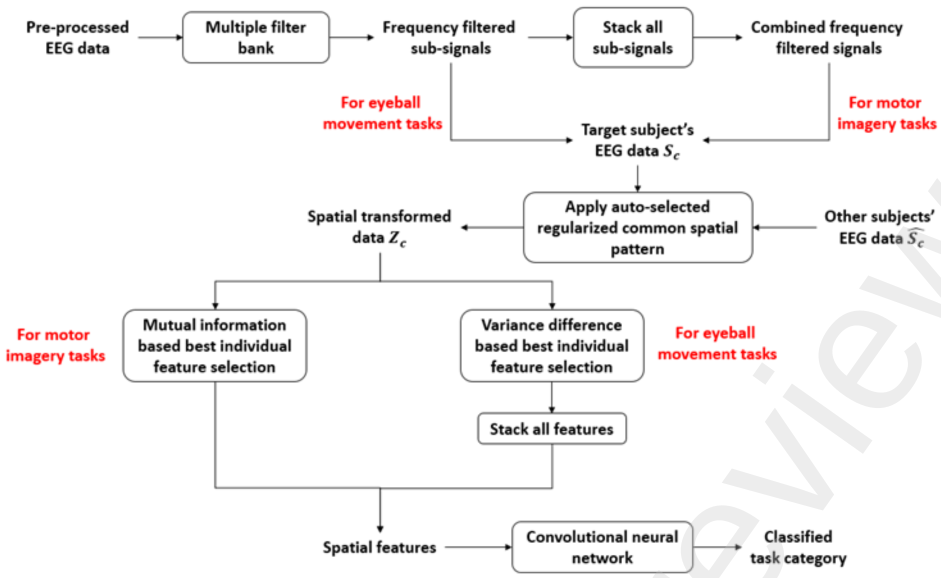

Fig. 11. Block diagram of EEG task classification model

2.3.4.2 Auto-selected regularized common spatial pattern

We first calculate R_c that is the co-variance of S_c and $\widehat{R_c}$ that is the co-variance of $\widehat{S_c}$:

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

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

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 Σ (β_c,γ_c) which is the mixed covariance matrix, by using the regularization parameters β_c and γ_c. We have.

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

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

where N_c is the total channel numbers; β_c controls the variance of the estimated covariance; γ_c is the second regularized parameter, which can reduce large eigenvalues and increase small eigenvalues. Then, decompose mixed covariance matrix and obtain 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 two classes' mixed matrix 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.

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

After that, we use the feature matrix and mutual information based regularization parameter selection method [2] to select the regularization parameters and recalculate the final spatial filter. We apply the spatial filter corresponding to these two parameters to the pre-processed signal and obtain the spatial filtered data Z_c.

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

We apply the filter 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}.

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

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

where $var()$ is the function of calculating variance. Then we can obtain the variance difference D_v. If the variance based best individual feature selection method is used, the channel data from the spatial filtered data Z_c with the largest variance difference is used as the final classification spatial feature. The corresponding labels are defined for the two types of variance features. The feature vector is X_c, and the label vector is Y_c. Their information entropy $H_I(X_c)$ and $H_I(Y_c)$ can be calculated. Then, use their joint probability density function to calculate their mutual information $M_I(X_c,Y_c)$.

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

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

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

where $p(x)$ is the probability of x; $p(y)$ is the probability of y; $p(x,y)$ 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 spatial filtered data Z_c with the largest mutual information is used as the final classification spatial feature.

2.3.4.3 Structure of classifier

Finally, a convolutional neural network (CNN) is introduced to classify the EEG features. The structure and parameters are shown in Figure 12 and Table II.

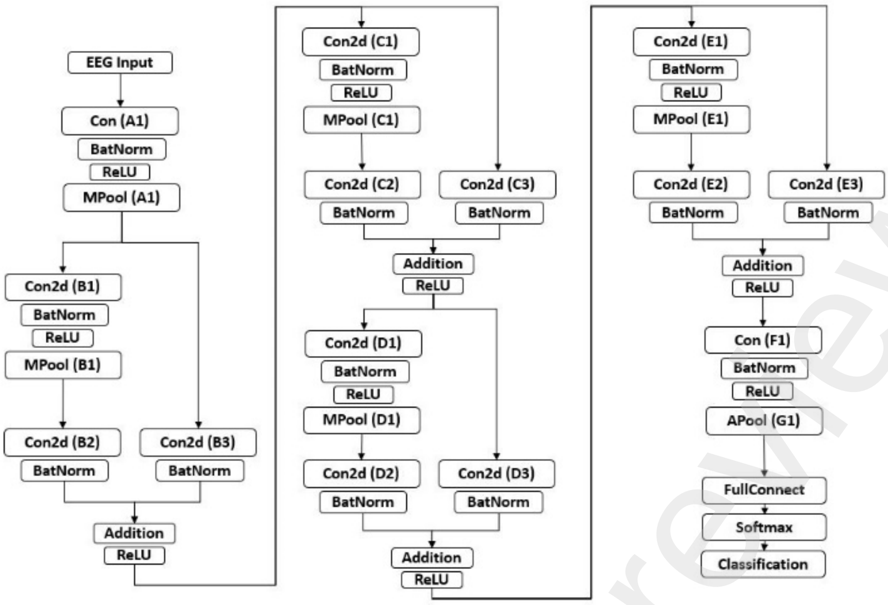

Fig. 12. Convolutional neural network structure used as classifier

TABLE II. Long short term memory convolutional 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

The Unicorn Hybrid Black is a consumer-grade bio-signal amplifier kit (shown in Figure 13). The device can obtain EEG recordings via Bluetooth. It is a dry device (which does not need bio-gel) that contains 8 DC-coupled analog input channels with 24 Bit resolution. The sample rate is 250 Hz. The EEG electrodes of this device have the advantage of fast and easy preparation with high-quality EEG signals [17].

Fig. 13. Unicorn Hybrid Black EEG acquisition device

3.1.2 BCI robot

The robot used in the experiment is shown in Figure 14. The body and the arm of the robot are made of acrylic sheets. There are eight engines where 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 that is used to control the eight engines. Inside this robot, a non-blocking I/O model is built to process the received commands and control the driver. The robot can communicate with other devices or software through WebSocket.

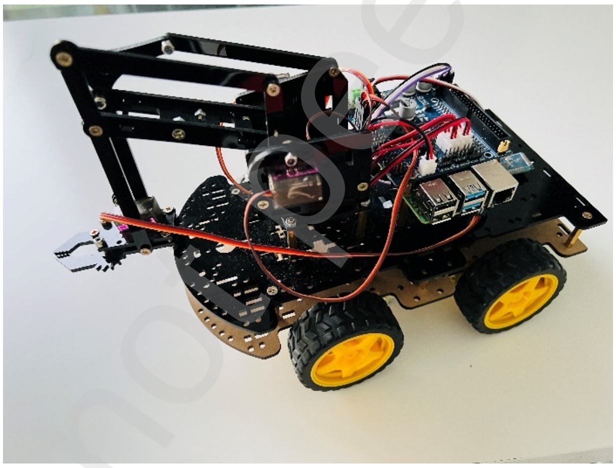

Fig. 14. Raspberry Pi 4 robot

3.2 Experiment preparation

3.2.1 Data collection

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

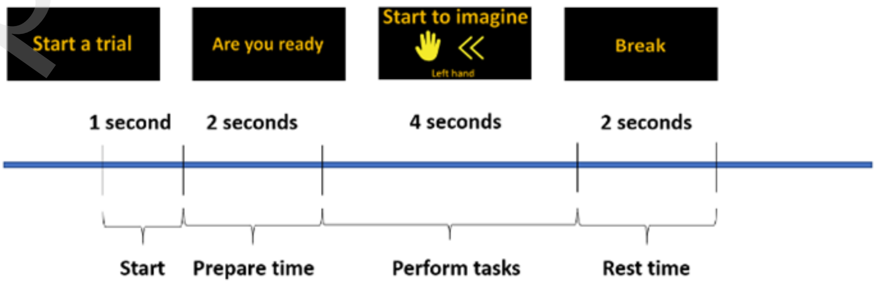

Fig. 15. Timing scheme of the paradigm for one trial

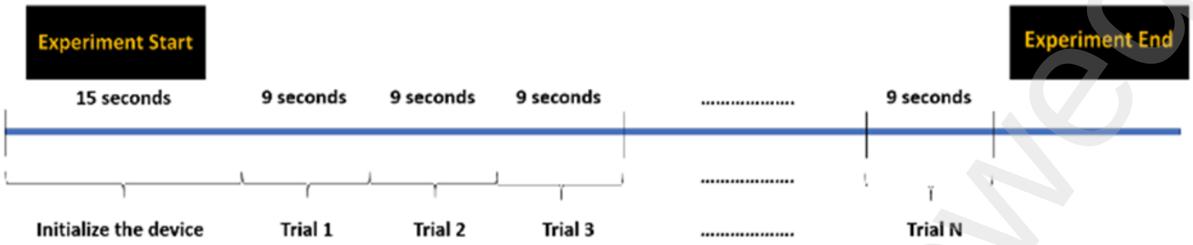

Fig. 16. Timing scheme of the paradigm for one experiment

Subjects performed corresponding tasks according to the screen prompts. The timing scheme of the paradigms is shown in Figures 15 and 16. The first 15 seconds was the beginning of the experiment, and the screen prompts 'experiment start'. After that, there was a one-second cue time before the start of a new trial. When the trial started, the screen prompted the subject to prepare for the brain tasks. The subject had 2 seconds to prepare. Then, the screen showed the specific task. The subject had 4 seconds to perform the task. Finally, the subject had 2 seconds break time. The experiment contains two motor imagery tasks, three eyeball movement tasks, and one EMG task, namely imaginary left-hand movement, imaginary right-hand movement, eyeball move to left, eyeball move to right, and eyeball move to down, and bite teeth. One trial of the experiment contains 9 seconds of data collection. Each experiment consists of 50 trials for each motor imagery task and 30 trials for EMG or eyeball movement task. There are 220 trials of data for one experiment. Each subject had to complete four experiments. Thus, there were a total of 880 trials of data for each subject.

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

3.2.2 Online data segmentation strategy

Performing brain tasks is a process, and it took a certain amount of time. During the execution of these brain tasks, the state of the brain transits from one state to another. The recognition model can calculate the probabilities 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 are correlated. Therefore, when we trained these two models, we could not only take the data of the time when the subjects were performing the task. We should also take the data every second and used the continuous probabilities to train the model.

We define a window of size 1,000. where each entry contains the data records for a period of 4 seconds. We set up the window at the beginning of the filtered signal. The window slides every second along the time sequence of the collected EEG signal. Every second, the window can extract an $8 \times 1{,}000$ data records. During data collection, when a task prompted appears on the screen, the subject began to perform the task for 4 seconds. Thus, every time when a motor event occurred, the 4 seconds after the event marker was the effective time of the task. The data during this 4 second period is the relevant data.

As the window moved along the time sequence, the window overlapped seven times with the effective data in each trial. The maximum length of time for the overlap is 4 seconds. If the window overlaps with the relevant data for 3 seconds or more, this window data is used as one epoch of task state data. Otherwise, if the window overlaps with the relevant data for less than 3 seconds, this window data is used as one epoch of the rest state data. Thus, in each trial, we can extract 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, this window's data is pure rest data. Thus, for each subject, there are 2,640 trials of task state data (880 data × 3 windows) when using this epoch extraction strategy. For each subject there are 3,520 trials of rest state data (880 data × 4 windows) from this strategy.

3.2.3 Commands design

As shown in Table III, a number of control commands are assigned in the experiment and each command correspond with 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, that require a fast response. In the experiments, the arm-related control commands do not require a fast response. Thus, the left-hand imagery movement is used to control the arm to switch direction. The right-hand imagery movement is used to control the arm to move forward. During the control process, the most important command is stop because this command needs to be used at any time to terminate the action which is being executed. The reaction time of the stop command has to be very short, and the machine should have the ability to recognize this command at any time. From the results of the model construction above, the EMG signal can give feedback immediately and accurately. Therefore, the EMG signal generated by biting of the teeth is used as the stop command. Eyeball movement classification is easier than motor imagery. Thus, the leftward and rightward eyeball movement is used to control the robot to turn left and to turn right. Downward eye movement is used to control the machine's forward movement.

TABLE III. 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

3.3 Experiment Setup

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

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

By using the online epoch extraction strategy, we have 400 trials of 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 task data contains seven segmented data. We used the segmented data to train and to test the model. When we test the model, we input the seven segmented data into the model together. The model will output seven probability vectors. For each vector, the output with the greatest probability is used as the recognized label. So, the model generates seven recognized labels. If one or more recognition outputs are the correct label and the other outputs are the rest state label, 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, then the output of this trial is judged to be misclassified.

After training the recognition model, we can use the model to calculate the probabilities of four EEG signals. Then, we input the current state probability vector and 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 then compared the recognition performance. The result is shown in Table IV.

TABLE IV. 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
Mean	**Accuracy (%)**	85.06	91.31	93.11	94.96
	kappa	0.7731	0.8673	0.8946	0.9230
	Se	0.0244	0.0259	0.0264	0.0268

From the results, these four signals are easy to distinguish because the time and spatial domain features of these three signals are distinct. They can all achieve an accuracy of over 80%. The models with the best performance are the EEG state recognition model and combining the recognition model and decision-making model. The decision-making model can help the model avoid errors because this model considers, not only the current signal state probability, but also the past signal state probabilities. Thus, the performance of using the decision-making model is better than using only the recognition model. Finally, we use the best model for further real-time control experiments.

3.3.2 Experiment B: Offline robotic performance evaluation

We used online segmented data to train and build the hybrid BCI real-time control system. There are the six control commands and one rest command that were explained in the command design section. Similar to the data collection steps, command prompts are randomly generated on the screen. According to the instructions that they are given, subjects had to perform corresponding actions of the commands within 4 seconds. There is a 2-second preparation time before the prompt occurs. Each command appears randomly 50 times giving a total of 300 actions. After that, the data of the previous 4 seconds are taken every second and input into the hybrid EEG real-time control system. The system outputs a classification result. The data length is 1,815 seconds, including 15 seconds of system initializing time. During the 4 seconds 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 V shows the real-time classification accuracy for each command.

TABLE V. 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
Mean correct number	50/50	44.33/50	43.67/50	46/50	36/50	35.67/50	255.67/300
Mean accuracy	100%	88.67%	87.33%	92.00%	72.00%	71.33%	85.22%

From the results, 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 that are controlled by motor imagery. Motor imagery depends on the imagination abilities of different subjects. Subjects with better imagination ability may be more likely to perform motor imagery. The overall recognition accuracy can reach more than 85%. Therefore, the classification accuracy of this system is good enough and can be used for further real-time control experiments.

3.3.3 Experiment C: Online robotic performance evaluation

The actual environment used to test the robot is shown in Figure 17. Firstly, we set a starting point and a destination point. Then, we placed obstacles between the two points and place a target object at the destination. The subjects needed to control the robot car from the starting point while avoiding obstacles and reach the destination. During the process, there were some target objects on the road. The subjects need to control the robot arm to push down these target objects.

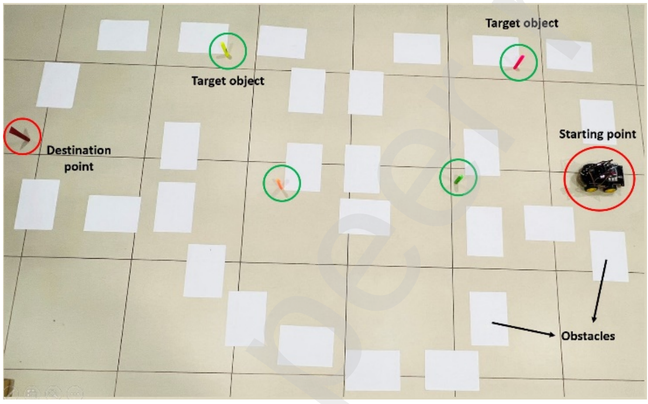

Fig. 17. The actual environment used to test the robot

In this process, we recorded the spending time i.e. the time that the robot car takes to reach the 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 the remote controller to control the car to do the above tasks and record the time, route, and driving distance. These were used as a reference to evaluate the reliability of EEG real-time control systems. By using a remote controller, we used the keyboard to manually control the robot to execute one command every three seconds. If the moving time and distance of using the EEG control system are close to or less than that of using the controller, the proposed control system is stable and reliable.

TABLE VI. Online testing performance of real time control system

	Experiment number	Spending time (s)	Running distance (m)	Target object (5) push down
Subject A	**1**	183.7	6.175	3
	2	191.4	7.410	5
	3	187.2	6.814	4
	Mean	187.4	6.800	4
Subject B	**1**	196.2	6.669	4
	2	206.7	6.894	4
	3	201.3	6.916	5
	Mean	201.4	6.826	4.3
Subject C	**1**	257.3	8.398	4

	2	249.6	8.151	3
	3	228.5	7.657	4
	Mean	245.1	8.069	3.7
Reference (remote controller)		171.2	5.928	5

Table VI shows the travelling time, the moving distance, and the number of targets pushed down by the robot car. From the results, the distance and travelling time spent in the proposed control system can be close to that of using the remote controller. In other words, the performance and stability of the system are considered good. However, the robot car cannot push down all the targets. The action of controlling the arm requires a combination of lots of different commands. Thus, it is more difficult to finish this action in a short time. This is also related to the subject's operational skills.

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. CNN can extract the time domain features of the signal and compress the features. EEG signals have multiple channels, so they can be seen as high-dimensional signals. Compared to other classifiers, LSTM-CNN is more suitable for extracting features along the time sequence of the high-dimensional data.

The signal is input to the recognition model every second, and the length of each input signal is fixed. When the user performed a brain task, especially the motor imagery task, the task could not be completed instantaneously. The process of executing the task takes a certain amount of time. Thus, we assume that the output of our recognition model is accurate, and the probability of the output represents the degree to which the subject completed the task.

For example, a motor imagery task takes 4 seconds to complete. We input the 4 seconds signal into the recognition model every second. Then, we can obtain four probabilities of the signal states. In the beginning, if the user does nothing, then the probability of the rest is 100%, and the probability of the motor imagery is 0. When the user performs the action for 2 seconds, the 2 seconds rest state signal and the 2 seconds motor imagery signal will be input into the recognition model. Then, the probability of motor imagery should be about 50%, and the rest state probability should also be around 50%. When the user performs the motor imagery action for 4 seconds, the whole task interval signal is completely input into the model. In this case, the probability of motor imagery should be 100%, and the rest state probability should also be 0. Therefore, when the user is performing 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 back to 100%. This is the ideal probability-changing process. However, we cannot ensure that the trained recognition model is perfect. So, in the practical application, the output cannot reach the ideal state mentioned above. Therefore, we cannot output the desired probability every time. This means there are some errors if we only consider the current probability output from the recognition model. Thus, the purpose of the proposed decision-making model is to learn the probability-changing process. It will extract the relation among the probabilities at different times.

We use two networks to predict the action. Firstly, the tasks are performed gradually, and the signal input into the model at each time overlaps with the signal at the previous time. In other words, they have shared information. Thus, the current signal state is related to the past signal states. As a result, the input of the first network is the state probability vector which contains both the current state probability and the past state probabilities. The purpose of this network is to learn the relation among these signal state probabilities. Based on the relation, the correct current ideal action can be predicted.

However, the action probability is a discrete variable, so we do not know whether the predicted probability is good or not. Thus, we create another network that evaluates whether the action predicted by the actor network is close to the ideal action. Its input is the current and previous state probabilities and current predicting action. The output is the evaluation score of this action, that is, 0 means bad, and 1 means good. In the beginning, we first trained the critic network so that the network has a certain ability to judge whether the action is good or not. Then, we trained both networks at the same time so that they reached an equilibrium state. Finally, the actor had 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.

In order to prove the feasibility of the proposed model, we drew the state changes of EEG signals according to the task execution by the subjects. Figure 18 shows the task category that the subject should perform, where the abscissa represents the time, and the ordinate represents the signal state. When the subject was performing 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 type of these three actions should be the same because they are continuous actions.

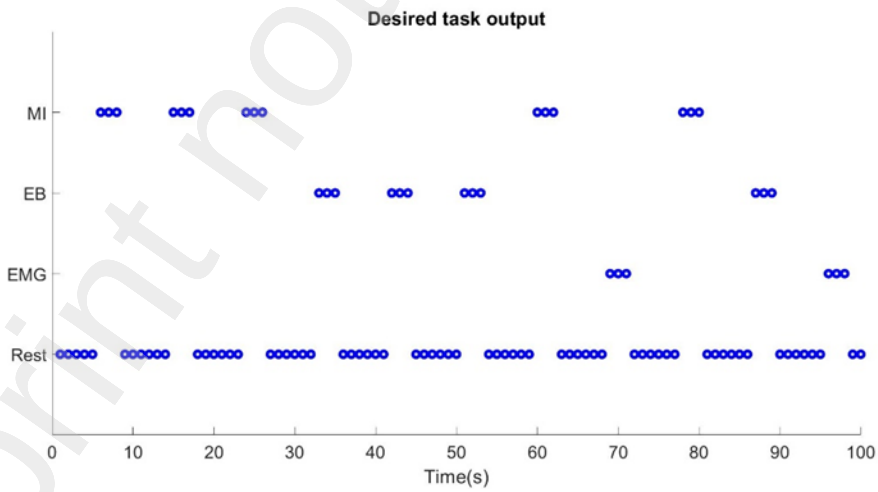

Fig. 18. The desired task label

Figure 19 shows the predicted actions using only the EEG recognition model. It can be seen that 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 by 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, due to external influences, when the subject is resting the subject may also generate some unexpected brain activities. Figure 20 shows the predicted actions using the actor-critic based decision-making model. When we apply this model on the experiment, we can see that most of the delay problems have been resolved, and the randomly generated actions due to external noise have also been corrected.

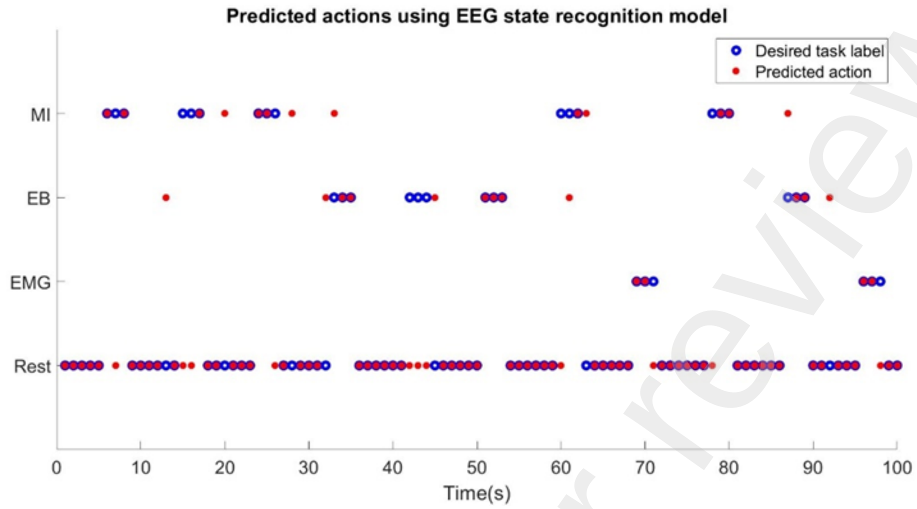

Fig. 19. The predicted actions using EEG recognition model

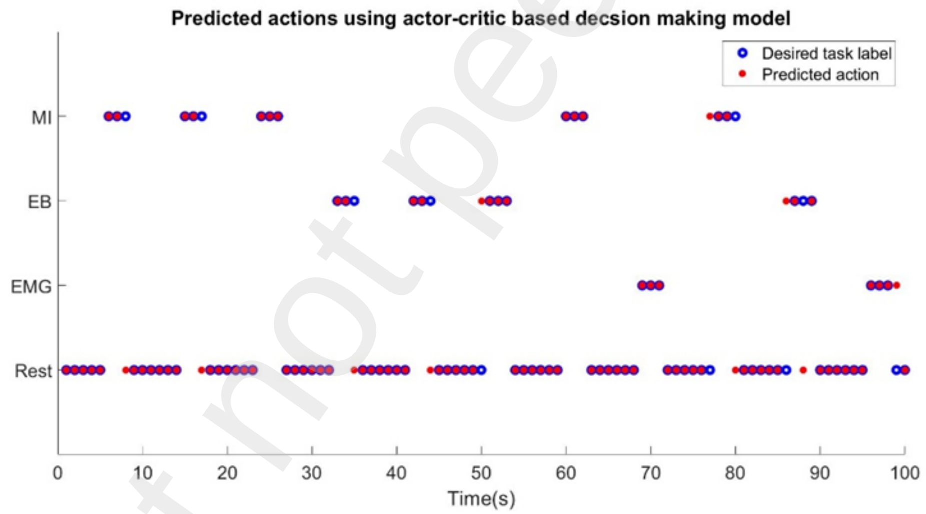

Fig. 20. The predicted actions using actor-critic based decision making model

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

4.2 Evaluate the performance of hybrid BCI real time controlled robot

The movement time reflects the reaction performance of the EEG system. During the control process, the robot car should constantly change direction and cooperate with the forward and stop commands to reach the destination. During the process, it needs to move by continuously changing the direction of the arm and controlling the robotic arm. If the time is short, it means that the EEG system can more accurately distinguish different motor imagery actions and eyeball movement actions. If it takes only a short time, it means the EEG system has been able to respond quickly to commands.

The moving distance reflects the stability of the EEG system. If the moving distance of the robot car is very long, it means that the robot often makes mistakes during the control process. This means that the system is not stable. In contrast, if the car can use the shortest distance to reach the destination, it means the robot can accurately follow the classified commands to find the best way to reach the destination. Thus, it also proves the stability of the control system. In Figures 21, 22 and 23, the trajectory of the car for three different subjects was shown.

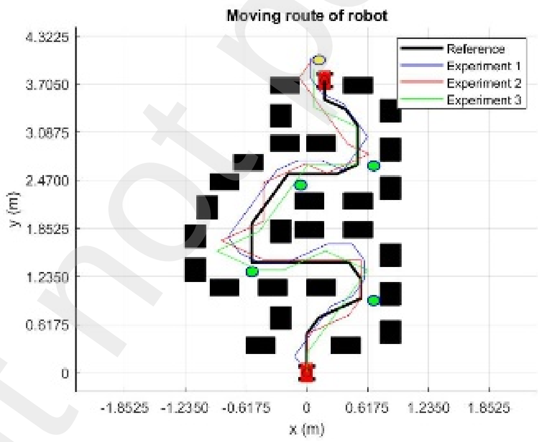

Fig. 21. The robotic moving route of subject A

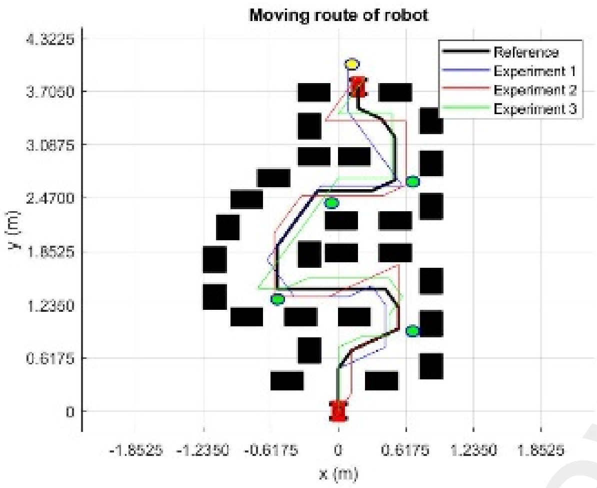

Fig. 22. The robotic moving route of subject B

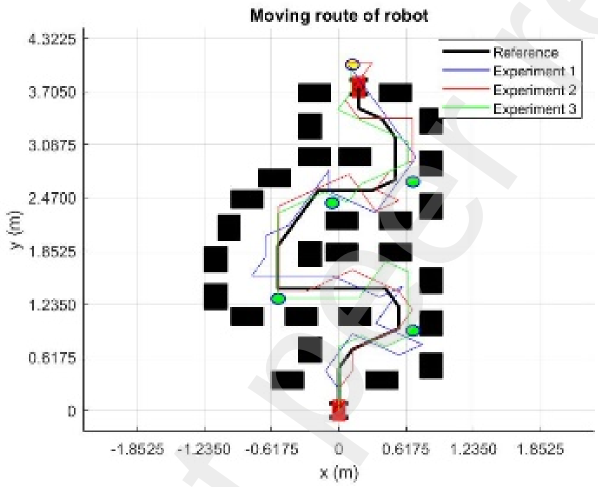

Fig. 23. The robotic moving route of subject C

When the ideal motion trajectories controlled by the remote controller is compared with the actual motion trajectories controlled by the proposed control system, we can be seen that both the ideal and actual trajectories are able to avoid obstacles and reach the destination. Although there was some deviation using the control system, the deviation was not large. It is noteworthy to mention that the subjects may feel tired after a period of concentration on handling the EEG real-time control systems and the robot car, and this his could also affect the results. Overall, this BCI real-time control system is accurate, stable, and reliable. It has the potential to be adopted in practical BCI control applications.

5. Conclusions

In this paper, an EEG state recognition model was proposed, which can be used to determine EEG signal states. In this model, a LSTM-CNN structure was introduced to extract both spatial features and time sequence features. An actor-critic-based decision-making model was proposed to predict the desired action, based on the signal state probabilities. The brain signals collected from three subjects were used to train and test the proposed models. Experimental results were given to compare the two proposed models. Compared to the traditional recognition method, the combined EEG state recognition model and the actor-critic-based decision-making model achieved the best performance. In this study, the best accuracy is 94.96%, and the kappa value is 0.9230.

In addition, we proposed a hybrid BCI-based real-time control system which is used to control a BCI robot car. This system includes two sub-systems, namely i) the data transmission system and ii) EEG dynamic classification system. In the data transmission system, a data acquisition server from the EEG device to other software is built, and a data processing and reception model is included, to achieve online EEG processing and analysis. In the EEG dynamic classification system, six signal analysis models were constructed to classify BCI commands. In the experiment, we trained an accurate and reliable system that can be used to control the BCI robot in a real environment. In the experimental results, the offline testing accuracy can achieve 85.22%. The best online controlling time was 187.4 seconds, and the best running distance was 6.8 meters. These results are close to the performance of the remote control.

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

6. REFERENCE

[1]. A. Ćwirlej-Sozańska, A. Wilmowska-Pietruszyńska and B. Sozański, "Validation of the Polish version of the World Health Organization Disability Assessment Schedule (WHODAS 2.0) in an elderly population (60–70 years old)," *International Journal of Occupational Safety and Ergonomics*, vol. 24, no. 3, pp. 386-394, 2018/07/03, 2018.
[2]. J. R. Wolpaw, D. J. McFarland and T. M. Vaughan, "Brain-computer interface research at the Wadsworth Center," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 222-226, 2000.
[3]. A. Jafarifarmand and M. A. Badamchizadeh, "Real-time multiclass motor imagery brain-computer interface by modified common spatial patterns and adaptive neuro-fuzzy classifier," *Biomedical Signal Processing and Control*, vol. 57, pp. 101749, 2020/03/01/, 2020.
[4]. Z. Tayeb, J. Fedjaev, N. Ghaboosi, C. Richter, L. Everding, X. Qu, Y. Wu, G. Cheng and J. Conradt, "Validating Deep Neural Networks for Online Decoding of Motor Imagery Movements from EEG Signals," *Sensors*, vol. 19, no. 1, 2019.
[5]. M. Tariq, P. M. Trivailo and M. Simic, "Motor imagery based EEG features visualization for BCI applications," *Procedia Computer Science*, vol. 126, pp. 1936-1944, 2018/01/01/, 2018.
[6]. S. L. Wu, Y. T. Liu, T. Y. Hsieh, Y. Y. Lin, C. Y. Chen, C. H. Chuang and C. T. Lin, "Fuzzy Integral With Particle Swarm Optimization for a Motor-Imagery-Based Brain–Computer Interface," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 1, pp. 21-28, 2017.
[7]. T. y. Mwata-Velu, J. Ruiz-Pinales, H. Rostro-Gonzalez, M. A. Ibarra-Manzano, J. M. Cruz-Duarte and J. G. Avina-Cervantes, "Motor Imagery Classification Based on a Recurrent-Convolutional Architecture to Control a Hexapod Robot," *Mathematics*, vol. 9, no. 6, 2021.
[8]. X. Shen, X. Wang, S. Lu, Z. Li, W. Shao and Y. Wu, "Research on the real-time control system of lower-limb gait movement based on motor imagery and central pattern generator," *Biomedical Signal Processing and Control*, vol. 71, pp. 102803, 2022/01/01/, 2022.
[9]. J. Choi, K. T. Kim, J. H. Jeong, L. Kim, S. J. Lee and H. Kim, "Developing a Motor Imagery-Based Real-Time Asynchronous Hybrid BCI Controller for a Lower-Limb Exoskeleton," *Sensors*, vol. 20, no. 24, 2020.
[10]. Y. Wang, B. Hong, X. Gao and S. Gao, "Implementation of a Brain-Computer Interface Based on Three States of Motor Imagery," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007, pp. 5059-5062.
[11]. A. Reust, J. Desai and L. Gomez, "Extracting Motor Imagery Features to Control Two Robotic Hands," in *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2018, pp. 118-122.
[12]. J. Yue, Z. Zhou, J. Jiang, Y. Liu and D. Hu, "Balancing a simulated inverted pendulum through motor imagery: An EEG-based real-time control paradigm," *Neuroscience Letters*, vol. 524, no. 2, pp. 95-100, 2012/08/30/, 2012.
[13]. K. Choi and A. Cichocki, "Control of a Wheelchair by Motor Imagery in Real Time," in *Intelligent Data Engineering and Automated Learning – IDEAL 2008*, Berlin, Heidelberg, 2008, pp. 330-337.
[14]. P. W. Ferrez and J. d. R. Millán, "Simultaneous real-time detection of motor imagery and error-related potentials for improved BCI accuracy," in *Proceedings of the 4th international brain-computer interface workshop and training course*, 2008, pp. 197-202.
[15]. R. Ortner, J. Scharinger, A. Lechner and C. Guger, "How many people can control a motor imagery based BCI using common spatial patterns?," in *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2015, pp. 202-205.
[16]. Y. An, H. K. Lam and S. H. Ling, "Auto-Denoising for EEG Signals Using Generative Adversarial Network," *Sensors*, vol. 22, no. 5, pp. 1750, 2022.
[17]. g. t. m. e. GmbH. "Unicorn Hybrid Black," https://www.unicorn-bi.com/.