ORIGINAL ARTICLE

CrossMark

Robot Navigation Using a Brain Computer Interface Based on Motor Imagery

Majid Aljalal[1] · Ridha Djemal[1] · Sutrisno Ibrahim[1]

Abstract

An interface between a human brain and a computer (or any external device) can be implemented for interchanging orders using a brain–computer interface (BCI) system. Motor imagery (MI), which represents human intention to execute actions or movements, can be captured and analyzed using brain signals such as electroencephalograms (EEGs). The present study focuses on a synchronous control system with a BCI based on MI for robot navigation. We employ a new feature extraction technique using common spatial pattern (CSP) filtering combined with band power to form feature vectors. Linear discriminant analysis (LDA) is employed to classify two types of MI tasks (right hand and left hand). In addition, we have developed posture-dependent control architecture that translates the obtained MI into four robot motion commands: going forward, turning left, turning right, and stopping. The EEGs of eight healthy volunteer male subjects were recorded and employed to navigate a simulated robot to a goal in a virtual environment. On a predefined task, the developed BCI robot control system achieved its task in 170 s with a collision number of 0.65, distance of 23.92 m, and successful command rate of 80%. Although the performance of the complete system varied from one subject to another, the robot always reached its final position successfully. The developed BCI robot control system yields promising results compared to manual controls.

Keywords Brain computer interface (BCI) · Common spatial pattern (CSP) · Electroencephalogram (EEG) · Motor imagery · Robot navigation

1 Introduction

Brain–computer interface (BCI) systems can generally be classified into invasive and noninvasive models. An invasive BCI captures signals inside the brain, whereas a noninvasive BCI captures signals from outside the brain (such as on the scalp). Although the signals captured from invasive BCI systems are relatively strong, surgery is required [1]. An electroencephalogram (EEG) is an electrophysiological technique for capturing electrical activity in the brain created during task performance [2, 3].Several applications have

been developed based on EEG signals, including mobile robot control, robotic prostheses for disabled individuals, the diagnosis of certain brain disorders, and entertainment.

In addition to the use of robots in industry, there is an increasing demand for robots in daily life, particularly for disabled individuals. Robots can be controlled by a healthy person with the help of an input device such as a mouse and keyboard. However, such interfaces are not useful for people with physical disabilities such as multiple sclerosis (MS) or amyotrophic lateral sclerosis (ALS) who, in most cases, cannot walk, use their hands and arms, or even speak. Thus, people with these conditions cannot effectively deliver their thoughts or actions to robots via conventional interfaces. The development of brain-controlled robots would be very useful in such cases. In other cases, a robot might be sent to specific places, particularly those inaccessible to humans, to take photos without any third party having knowledge of the robot's orientation.

EEG signals can be divided into three categories, namely: *event related desynchronization/event related synchronization* (ERD/ERS), P300, and *steady state visually evoked*

✉ Majid Aljalal
algalalmajid@hotmail.com

Ridha Djemal
rdjemal@ksu.edu.sa

Sutrisno Ibrahim
suibrahim@ksu.edu.sa

[1] Department of Electrical Engineering, College of Engineering, King Saud University, Saudi Arabia, P.O. Box 800, Riyadh 11421, Saudi Arabia

potential (SSVEP). These signals are based on brain electrode positioning and the external stimuli of the subjects. The first category, (ERD/ERS-based BCI),consists of controlling a robot with EEG signals recorded during mental task performance, e.g., motor imagery, mental arithmetic, and mental rotation [4].Choosing the best BCI type is dependent on the attempted scenario. For example, for robot navigation in an environment in which landmarks are stored in its memory, either the P300 or the SSVEP approaches can be used only when necessary. This allows the user to choose a desired landmark and let the robot use its autonomous system to reach the landmark. If the environment is unknown or the robot encounters unexpected obstacles, motor imagery is preferable for direct control of robot steering.

Several studies related to EEG-based BCI systems such as SSVEP [5, 6], P300 [7], and ERD/EDS [8, 9] focus on mobile robot control. In addition, several studies have employed BCIs for robotic arm control with ERD/ERS and SSVEP brain signals [10, 11].The system developed by Pfurtscheller et al. [12] combines ERD/ERS and SSVEP BCIs to control an electrical hand prosthesis.

The appropriate analysis and processing (filtering, feature extraction and classification) of MI states can generate accurate commands for control. Many useful techniques can extract features from EEG signals, such as common spatial pattern (CSP) [8, 9, 13], wavelet transform (WT) [13, 14], fast Fourier transform (FFT) [15, 16], and logarithmic band power (LBP) [17–19]. In this study, we propose a novel feature extraction technique using CSP. Rather than using variance, we propose to employ LBP to form feature vectors. Linear discriminant analysis (LDA) is employed for the classification of two types of MI tasks (right hand and left hand).

In this study, a BCI system based on MI was developed to allow 2D direct control of robot motions in an unknown environment. Figure 1 shows the proposed system for mobile robot control. The overall system is divided into two subsystems: the first is the BCI system, and the second is the robot control system. The BCI system acquires EEG signals from a human brain and classifies them into two user mental states, left-hand and right-hand motor imagery. The robot control system provides four low-level motion commands, "going forward," "turning left," "turning right," and "stopping." These commands are generated using a posture-dependent control paradigm that receives classification outputs or directional commands (left and right) as input. The performance of the developed system is evaluated at three levels. The first level includes an evaluation of the BCI system. At the second level, we evaluate the performance of the posture-dependent state controller. The third level of evaluation is applied to the complete system, including robot simulation. Further details about the overall system and its performance are presented in ensuing sections.

The remainder of the paper is organized as follows. Section 2 describes the EEG recording process and briefly explains the proposed methods, including preprocessing, feature extraction, classification, the control unit, and a simulated robot as well as evaluation methods. The study results and a discussion are provided in Sect. 3. Finally, the paper is concluded in Sect. 4.

2 Method

2.1 Subjects

The eight male subjects used in this experiment were healthy, right-handed, and between 29 and 32 years of age. The subjects were asked not to eat or drink for at least 60 min prior to the experiment. None of the subjects had previous BCI experience. Before the experiment commenced, the procedures and objects of the experiment were explained to the subjects. Written informed consent was obtained from all the participants involved in the study for both their participation and the publication of any accompanying data (including the figures in this paper).

2.2 Data Recording

We performed EEG recording of the subjects using the g.tech software and hardware package [20] in a research

Fig. 1 Proposed system architecture of mobile robot control system by BCI based on MI
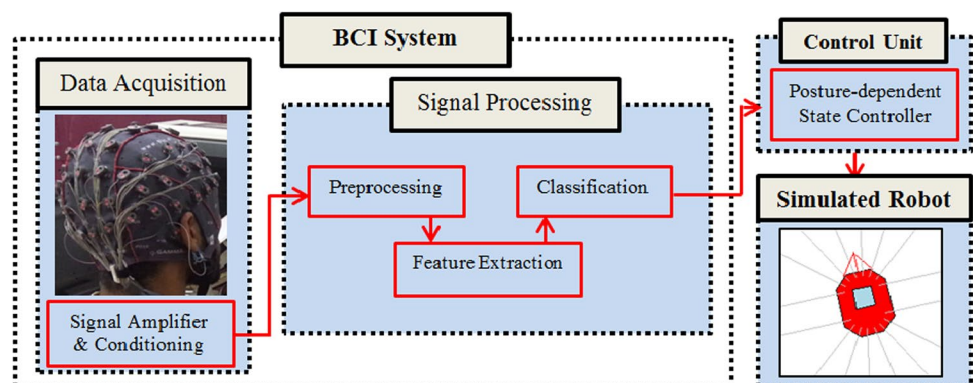
laboratory at King Saud University, Saudi Arabia. Figure 2 shows our EEG data acquisition setup. The data acquisition setup includes the following: g.Gamma cap^2 with 64 active EEG electrodes, g.Scarabeo, electrode driver (terminal), multichannel signal amplifier g.HIamp, and g.Recorder software (as shown in Figs. 2, 3). In accordance with the international 10–20 system [21], the EEG data were recorded using a 64-channel EEG. The right ear lobe was used as a reference while the forehead (Fz) was used as ground. All the signals from the channels were amplified, sampled at a frequency of 256 Hz, and filtered at 0.5–100 Hz using a Butterworth filter. To suppress line noise, a 60 Hz notch filter was employed. All impedances were kept below 30 kΩ. The signals recorded had a duration of at least 35 min for each subject. The total duration of preparation and electrode setup was 20–40 min.

2.3 Experimental Paradigm

The subjects sat in a comfortable chair one meter away from a computer screen. A short acoustic warning tone with a fixation cross was shown on the black screen at the beginning of each trial (t = 0 s). This tone signaled the subject that the trial had started. At the third second (t = 3 s), a sign in the form of an arrow was presented on the screen until t = 7 s. This arrow faced either left or right, asking the subject to achieve a desired task (imagine the movement of the left hand if the left arrow appears or imagine the movement of the right hand if the right arrow appears) until the fixation cross and arrow disappeared at t = 7 s. No feedback was provided during task performance. After performing the task, the screen returned to black for 2–4 s, providing a random short break. After the break, the cross with acoustic stimulus appeared, signaling the next trial. Each cue (left arrow or right arrow) was shown 12 times, in a randomized order, within each run. Each subject completed five runs, and the paradigm is illustrated in Fig. 4. At the conclusion of the experiment, each subject had completed 120 trials of motor imagery (60 right-hand and 60 left-hand) with three files. The first file consists of recorded EEG signals (including 120 trials with break signals). The labels of the trials (right hand or left hand) are included in the second file. The third file contains the time points of the 120 cues.
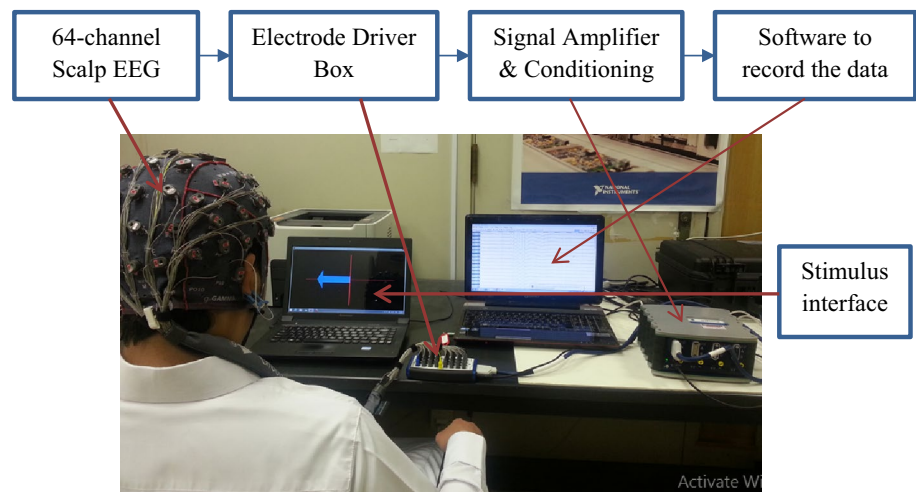
Fig. 2 Our EEG data acquisition setup

Fig. 3 g.GAMMAcap^2 with g.SCARABEO electrodes (left), driver box (middle), and g.HIamp (right) [19]
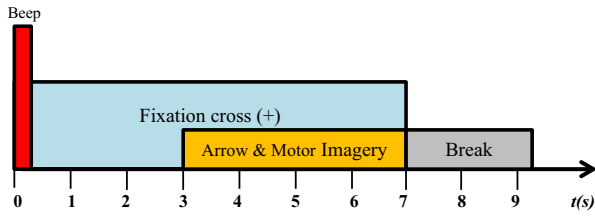
Fig. 4 Timing scheme of the experimental paradigm

2.4 Signal Preprocessing

Before filtering, the EEG data were segmented into E-matrices with a size of $ch \times T$ for each trial, where T indicates the number of samples per channel during a specific time interval, and ch specifies the number of channels. In our experiment, time intervals of 2 s with offsets of 0.0 s were applied for all subjects per trial. During signal recording, the subjects were asked to perform the motor imagery task for 4 s to enable greater flexibility for finding the optimal time interval and offset. Because robot control is a real time application, it is necessary to decrease the time required to instruct the robot as quickly as possible without significantly impacting output accuracy. For this purpose, we investigated the optimal time interval for an EEG epoch and discovered an optimal time interval of 2 s, which has been previously confirmed [22].

The EEG signals contain artifacts resulting from the eye, also known as EOG artifacts [23], and/or body movements. These artifacts make the algorithm efficient in some states but inefficient in others. Although the subjects were told not to move their eyes or blink while recording the data, they still generated some noise. It should be noted that we intentionally performed two steps of EEG filtering. The first filtering was done during recording and is intended to remove noise, artifacts and other unnecessary signals. A frequency band of 0.5 - 100 Hz is most commonly used during EEG recording to obtain a wide frequency spectrum for further analysis and investigation. The second filtering was largely intended to capture a region or frequency of interest. For motor imagery application, the most useful frequency range is in the alpha–beta band. Thus, in order to extract important information, we used a bandpass Butterworth filter of order 5 with a frequency band of 8–34 Hz during preprocessing.

2.5 Feature Extraction using CSP

Several types of feature extraction techniques can be employed for two-class imagery task discrimination. In this experiment, we employed the most popular method, CSP based on variance. We used CSP-based logarithmic band power and employed a CSP algorithm as a spatial filter that leads to peak variances for the discrimination of two classes (left-hand and

right-hand motor imagery) and to reduce the number of channels used [24]. Computing the projection matrix constructs a set of common spatial pattern filters. The algorithm starts by computing the normalized spatial co-variance for both classes. This is achieved by the following equation,

$$C_R = \frac{E_R E_R'}{trace(E_R E_R')}, C_L = \frac{E_L E_L'}{trace(E_L E_L')} \tag{1}$$

where E_R and E_L denote a single trial under two conditions (right hand and left hand) of size $ch \times$, where E' is the transpose of E, and $trace(EE')$ is the sum of the diagonal elements of EE'. Calculating the average over all the trials of each class produces $\bar{c_R}$ and $\bar{c_L}$, the averaged normalized covariances. For our data, both $\bar{c_R}$ and $\bar{c_L}$ are on the order of 62×62. The total composite spatial co-variance is then obtained by

$$C_c = \bar{C}_R + \bar{C}_L \tag{2}$$

This is factorized into the eigenvectors matrix U_C and the diagonal matrix of eigenvalues λ_C such that

$$C_C = U_C \lambda_C U_C' \tag{3}$$

After arranging the eigenvalues in descending order, the whitening transformation P is computed as follows:

$$p = \sqrt{\lambda_C^{-1} U_C'} \tag{4}$$

Subsequently, we must find

$$S_R = P\bar{c}_R P' \text{ and } S_L = P\bar{c}_L P' \tag{5}$$

To test these calculations, the sum of the corresponding eigenvalues of S_R and S_L should be the identity matrix, and S_R and S_L should have the same eigenvectors such that

$$S_R = B\lambda_R B' \tag{6}$$

$$S_L = B\lambda_L B' \tag{7}$$

$$\lambda_R + \lambda_L = I \tag{8}$$

where B is any orthonormal matrix that satisfies

$$B'(S_R + S_L)B = I \tag{9}$$

The smallest eigenvalues with the corresponding eigenvectors for S_L have the largest eigenvalues for S_R, and vice versa. This is an indicator that the eigenvalues for one class are minimized while the eigenvalues for the other class are maximized at that same point. Thus, the co-variance between the two classes is successfully maximized. A set of common spatial pattern filters (projection matrix) can be obtained as

$$W = P'B \tag{10}$$

2.6 Feature Extraction and Channel Selection

From the previous section and Eq. (10), we have

$$W' C'_R W = \lambda_R = \begin{vmatrix} \lambda_1 & . & . \\ . & \lambda_2 & . \\ . & . & \lambda_{ch} \end{vmatrix} \text{ and}$$

$$W' C'_L W = 1 - \lambda_R = \begin{vmatrix} 1 - \lambda_1 & . & . \\ . & 1 - \lambda_2 & . \\ . & . & 1 - \lambda_{ch} \end{vmatrix}$$

where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{ch}$. Therefore, the first CSP filter w_1 provides the maximum variance of class 1, and the last CSP filter w_{ch} provides the maximum variance of class 2. For dimensionality reduction, only the first and last m filters should be used, such that

$$W_{Csp} = [w_1 w_2 \ldots w_m w_{ch-m+1} w_{ch-m} \ldots w_{ch}] \in R^{2m \times ch} \quad (12)$$

and the filtered signal $S(t)$ is given by

$$s(t) = W_{Csp} e(t) = [s_1(t) s_2(t) \ldots s_d(t)]' \quad (13)$$

where d is the reduction number and is equal to $2 \times m$. The reduction number is the number by which the channels should be reduced. Thus, for each class's EEG sample matrix, we only select the small number of signals (m) which are most important for discriminating between the two classes. Finally, the feature vectors $f = (f_1, f_2, f_3, \ldots, f_{2m})'$ can be calculated by the following equation:

$$f_i(\text{var}) = \log \left[\frac{\text{var}[s_i(t)]}{\sum_{i=1}^{2m} \text{var}[s_i(t)]} \right] \quad (14)$$

The CSP-variance method for left- and right-hand MI tasks did not provide peak variances [25, 26] as expected for discriminating the two classes. This made the features of left- and right-hand MI tasks proximal to each other, as shown in the upcoming results section. Thus, in order to separate these features and redistribute them, we propose to extract new features using logarithmic band power (LBP) rather than variance. The new features are extracted as follows:

$$f_i(LBP) = \log \left[\frac{1}{N} \sum_{n=1}^{N} |s_i(t)|^2 \right] \quad (15)$$

Band power is a commonly used method in EEG analysis for estimating the power of EEG signals [27, 28]. In this study, we apply the band power method on the signals filtered with the W_{Csp} filter. In other words, we perform conventional CSP but replace variance with band power. The variance is defined by $\frac{1}{N} \sum_{n=1}^{N} (s_i(t) - \mu_i)^2$, which is similar to Eq. (15) if the mean value is ignored. We also retain the use of logarithmic operation as normally used in the CSP-variance method. This modification redistributes the features

of left- and right-hand MI tasks for easy classification. As a result of CSP-LBP, d features were obtained. In our work, we selected just the first and last filters ($m = 1$); hence, the size of our feature vector is 2.

(11)

2.7 Classification and Cross-Validation

In this study, we employed linear discriminant analysis (LDA), a well-known binary classification method based on mean vectors and covariance matrices of feature vectors for individual classes. Linear discriminant analysis uses a hyperplane to distinguish between classes, minimizing the variance within a class and maximizing the variance between classes [29]. The LDA classifier is explained as follows.

Let the input feature vectors $f_1, f, f_3, \ldots, f_n$ be defined for training, where $f_i = [f_1 f_2 ... f_d]$. n_1 is the group of vectors created from Class 1, and n_2 is the group of vectors created from Class 2. LDA is based on the transformation of a d-dimensional vector f to the scalar z,

$$z = w^T f \quad (16)$$

Using LDA, the task is to obtain an optimal projection w in d-dimensional space so that the distribution of z is easy to discriminate. This is achieved as follows:

$$W = \left[\frac{\mu_2 - \mu_1}{C_1 - C_2} \right]^T \quad (17)$$

where μ_1 and C_1 are the mean vector and covariance matrix, respectively, for Class 1, and μ_2 and C_2 are the mean vector and covariance matrix, respectively, for Class 2. The weight vector can then be given by

$$b = \frac{W}{2} (\mu_1 + \mu_2) \quad (18)$$

and the output of the classifier is

$$a = xW - b \quad (19)$$

where x is an unknown vector (new sample) classified according to its feature position with respect to the separating line in the space.

We employed k-fold cross validation to estimate our classification accuracy as follows. The complete dataset was randomly partitioned into k equal subsets. One subset was employed for validation (test), and all the other subsets were employed for training to produce an initial classification accuracy. This operation was then repeated k times (fold). The subset employed for validation was different each time. After completing all the operations, a single classification accuracy was found by calculating the average value over all the obtained classification accuracies [30]. The classifier module was given the extracted features of 120 trials, and we applied 5-fold cross validation. Therefore, each training

set included 96 trials (80%) and each testing set included 24 trials (20%). The classification accuracy can be given by

$$accuracy = \left(\frac{N_{correct}}{N_{total}} \right) \times 100\% \qquad (20)$$

where $N_{correct}$ is the number of correctly classified vectors, and N_{total} is the overall number of vectors.

Figure 5 shows a block diagram of our proposed approach for signal processing. First, the EEG signal is segmented into time windows of 2 s. Second, the output of the segmentation process is fed to an 8–34 Hz Butterworth bandpass filter. Next, the CSP algorithm with a reduction number (d) of 2 is applied to the filtered signals. Then, rather than using variance, LBP is applied to the CSP feature to form the featurevectors. Finally, LDA is employed as a classifier to discriminate between the two classes (right-hand and left-hand MIs). The method was implemented using MATLAB R2013b on a Windows 8 PC with an Intel i3 Core, 2.30 GHz processor.

2.8 Control Unit

After the classification process, the control unit translates the obtained categories into robot motion commands, as shown in Fig. 6. These motion commands are then fed to a simulated robot. In our study, we developed a posture-dependent control architecture that translates the obtained categories (classification output: right hand and left hand) to four motion robot commands ("going forward," "turning left," "turning right," or "stopping"), as shown in Fig. 7. First, the obtained categories are mapped into two-directional commands (left or right). Then, these directional commands are mapped into four low-level motions to navigate the robot to the destination (target position) using the developed posture-dependent control paradigm. In this paradigm, low-level commands are issued depending on the postural state of the robot. Figure 7 shows a state-machine diagram of the proposed paradigm. For maintaining stability, the robot was developed to prevent it from walking while turning left or right. While the robot remains in the "stopping" state, a "left" directional command forces the robot goes into a "no change" state until a "left" or "right" directional command is received. If a "left" (or "right") directional command is received, the robot continuously turns to the left (or to the right) until a "left" or "right"
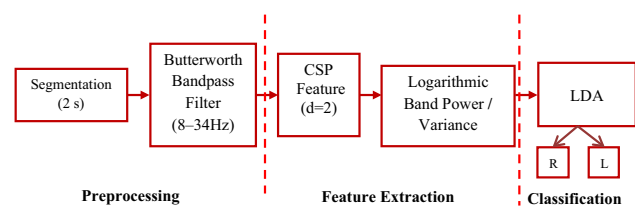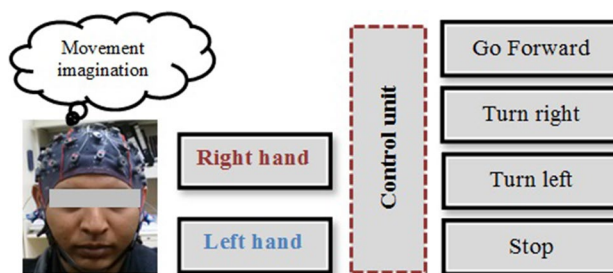

Fig. 6 Converting the two-directional commands (right and left) into four low-level robot motions

directional command is detected again. If a "left" directional command is detected, the robot stops turning. However, if a "right" directional command is detected, the robot walks continuously forward in the direction it is facing. Finally, while the robot remains in a "going forward" state, a "left" or "right" directional command forces the robot to stop (i.e., go into a "stopping" state). In other words, a "left" or "right" directional command stops the robot if it is walking forward, while only a "left" directional command stops the robot if it is turning left or right. Table 1 shows the robot motions based on our posture-dependent control architecture.

2.9 Simulated Robot and Environment

The MobileSim software for simulating mobile robots (Omron Adept MobileRobots) was employed as a software platform in
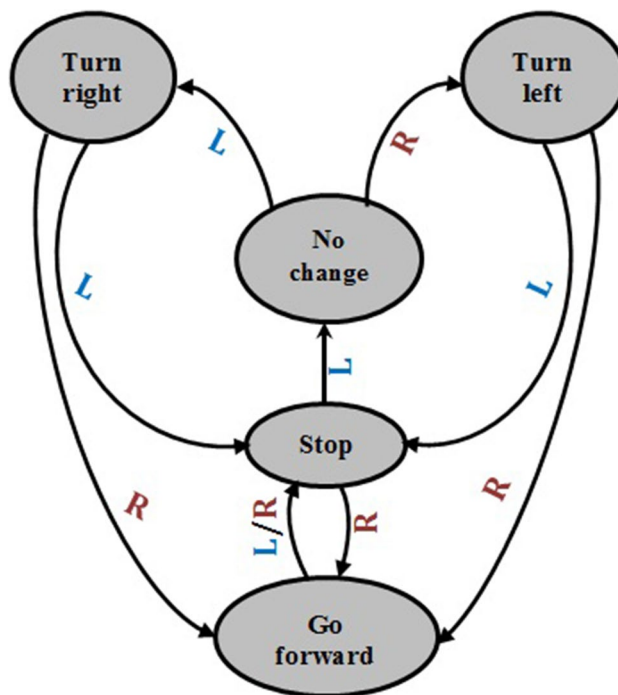

Fig. 5 Block diagram of proposed approach for signal processing

Fig. 7 Proposed posture-dependent control architecture

Table 1 Motion outputs based on our posture-dependent control architecture

Output motion	Last state	Required Imagination
Going forward	Stopping	Right hand
	Turning left	Right hand
	Turning right	Right hand
Turning left	No change	Left hand
Turning right	No change	Right hand
No change	Stopping	Left hand
Stopping	Going forward	Left/right
	Turning left	Left hand
	Turning right	Left hand
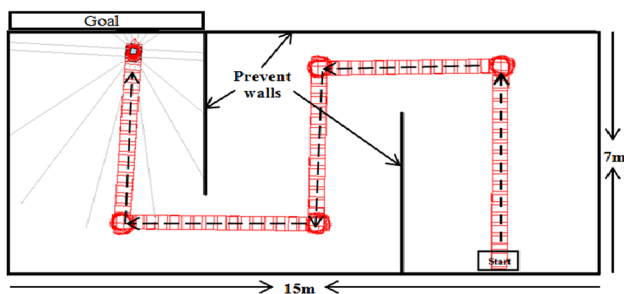
Fig. 8 The path to be tracked by the simulated robot in one attempt

our study. As previously mentioned, our control system sends low-level motion commands to a simulated robot according to the proposed posture-dependent control architecture shown in Fig. 7 and discussed in the previous section. C code was developed to control the behavior and speed of the robot. The robot's rotating speed was set to 22 s/360° (0.28 rad/sec), and its movement speed was set to 0.5 m/s. We employed Mapper3 software to develop the robot's operating environment and place obstacles such as walls (Omron Adept MobileRobots). Figure 8 illustrates the virtual environment map employed in our experiment with a screen image of a successful navigation attempt by the simulated robot. The converted real size of the virtual environment is 15×7 m^2. The environment is divided into three 5×7 m^2 rooms. Each attempt initiates with the robot at the starting point (Start) and terminates when the robot reaches its destination point (Goal). The relative size of the target is 5×1 m^2. The robot stops moving forward or rotating if it crashes into any of the prevent walls.

2.10 Evaluation

2.10.1 Performance of the BCI System

The BCI system's performance was evaluated in terms of classification accuracy, the ratio of correctly classified trials to total trials executed by each subject [30]. This measures

how well the BCI system differentiates between the implemented mental tasks. In this study, there are two mental tasks (right hand and left hand) for differentiation. High classification accuracy leads to sound performance for the complete mobile robot control system, which is based on the BCI system. In this evaluation, all the classification accuracies were calculated by fivefold cross validation as explained previously. The extracted feature vectors of 120 trials were sent to the classifier module (LDA). The training set comprises 96 trials (80%) and the testing set comprises 24 trials (20%). The classification accuracy can be given by Eq. (20).

2.10.2 Performance of the Posture-Dependent State Architecture

The commands issued from the posture-dependent state architecture can be correct or incorrect dependent on a trial's classification results. Any produced command includes at least one trial. For example, to make the robot turn to the right, two correctly classified trials (left and right) are required. If the first trial is classified as "left hand," and the second trial is classified as "right hand," the issued command will be "turning right," which is a successful command. However, if the first trial is classified as "left hand," but the second trial is also classified as "left hand," the issued command will be "turning left," which is an unsuccessful command. Similarly, if the first trial is classified as "right hand," the issued command will be "walking forward," which only requires one trial, right hand. This is also an unsuccessful command. The posture-dependent state architecture was developed to avoid certain misclassifications. For example, for the robot to stop walking when it is walking forward, either a "left" or "right" directional command is required. To ensure a reliable evaluation, the average successful command rate was calculated over 100 runs for each command ("going forward," "turning left," "turning right," and "stopping").

2.10.3 Navigation Performance

The task in a single attempt is as follows. From the starting point, the robot should be navigated to its destination without hitting any walls. Due to certain limitations, the robot was navigated offline by the EEG signals of the eight subjects. Each trial (right hand or left hand) to be issued was randomly chosen from a list of trials (trials file) and fed to the BCI system for signal processing. To evaluate the complete system, the robot's navigation performance was measured in terms of several metrics per attempt. The first metric is *task time*, the total time required by the robot to achieve its task (in seconds). The second metric is the *distance traveled* by the robot to reach to its destination (in centimeters). The third metric *is the number of collisions* that the robot had

with walls. The number of total commands required for the robot to reach its destination is considered as well. For each subject, the robot's attempt was repeated twenty times, and the average was taken for the metric calculations. To enable a comparison with BCI performance, these metrics were also calculated for manual control.

3 Results and Discussion

3.1 Performance of the BCI system

Table 2 provides the evaluated systems' classification accuracy for each of the eight subjects, based on LDA, for classifying two classes of EEG-related right-hand and left-hand MI. The table shows that using CSP-LBP produced better results than those obtained from CSP-variance with an average classification accuracy of 76.46% over the eight subjects. This improvement was clear with all subjects except *hsm* and *ysf*. This demonstrates the effectiveness of the proposed method (CSP-LBP) compared to the conventional CSP-variance method. To complement our quantitative analysis, we also performed a qualitative analysis by visualizing the extracted features in 2D plots. Figure 9 shows a 2D plot of 100 feature vectors randomly selected from all subjects at

Table 2 Classification accuracies of eight subjects

Subjects	Feature extraction method	
	CSP + Variance	CSP + Logarithmic band power
Trial No.	120	120
akm	80.00	81.67
khl	75.83	78.33
mjd	77.50	78.33
mhd	80.00	81.67
hsm	68.33	67.50
abm	85.00	87.50
ysf	73.33	72.50
khd	61.67	64.17
Avg.	75.21	76.46

the 8–34 Hz frequency band for the two approaches. The figure shows that our method redistributes the left- and right-hand features in a manner that enables straight forward classification.

3.2 Performance of the Posture-Dependent State Architecture

A low-level command issued from the posture-dependent state architecture can either be successful or unsuccessful. This depends on the classification results of the trials and whether a trial is correctly or incorrectly classified. In the mapping step, a trial classified as right hand will be mapped as a "right" directional command. Similarly, a trial classified as left hand will be mapped as a "left" directional command. Each low-level command requires at least one trial (one directional command). To ensure a reliable evaluation, the average number of successful commands for all the subjects was calculated over 100 runs for each command. Table 3 details the required and issued commands of the first subject (*akm*). We use this subject to provide a detailed explanation of the posture-dependent state controller evaluation. Table 3 shows the successful command rate for each command, as well as the average successful command rate. The 100 runs were divided into groups according to the likely last state for each required command.

For example, to make the robot go forward, 100 runs, divided into three groups based on the likely last state (stopping, turning left, or turning right), were issued by EEGs of subject *akm*. The results of the obtained commands are distributed as follows: 90 runs (31 + 28 + 31) were translated as "going forward" commands, 3 runs were translated as the "no change" state, and 7 runs were translated as "stopping" commands. Thus, out of 100 required commands, 90 commands were translated as successful commands while 10 commands were translated as unsuccessful commands, resulting in a successful command rate of 90%. The identical steps were performed for the remaining commands, and the successful command rates were 56, 75, and 91% for the "turning left," "turning right," and "stopping" commands, respectively.

The required and issued commands and successful command rates of all the subjects are summarized in Table 4.

Fig. 9 2D plot of randomly selected feature vectors from all subjects at 8–34 Hz frequency band using (**a**) CSP + variance and (**b**) CSP + logarithmic band power
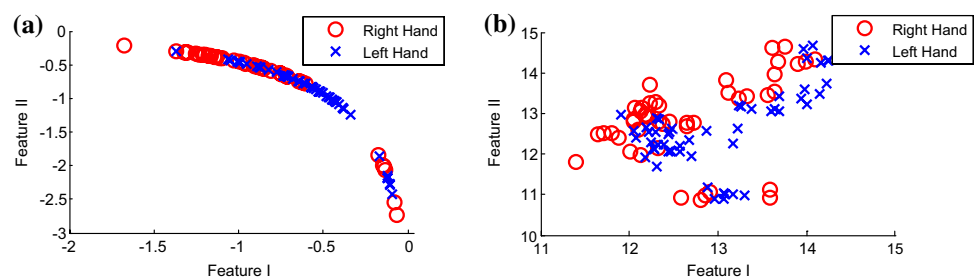
Table 3 Details of required and issued commands of subject *akm*

Required commands	Last state (no. of trials)	Issued commands					Successful commands rate
		GF	TL	TR	NCH	SP	
Going forward (GF)	Stopping (34 trials)	31	0	0	3	0	90.00%
	TL (33 trials)	28	0	0	0	5	
	TR (33 trials)	31	0	0	0	2	
No change (NCH)	Stopping (100 trials)	22	0	0	78	0	78.00%
Turning left (TL)	NCH (100 trials)	0	72	28	0	0	$72 \times 0.78 = 56.16\%$
Turning right (TR)	NCH (100 trials)	0	4	96	0	0	$96 \times 0.78 = 74.88\%$
Stopping (SP)	GF(50 trials)	0	0	0	0	50	91.00%
	TL (25 trials)	4	0	0	0	21	
	TR (25 trials)	5	0	0	0	20	
Average							**78.00%**

The table shows promising results, particularly for subjects such as *abm,* who presented a successful command rate of 83.75%. However, low success rates are associated with some of the subjects. As mentioned earlier, the results are directly affected by the classification accuracy of the BCI system. This is clearly apparent for the "turning left" and "turning right" commands. Because the commands "turning left" and "turning right" require two correctly classified trials (left and right), they are more likely to be incorrectly translated than other commands. For example, to make the robot turn to the right, two consecutive trials (left and right) must be classified correctly. If at least one trial is classified incorrectly, the command will be translated as an unsuccessful command. To avoid this problem, we programmed the robot as follows. If the robot receives a "turning left" command, the robot will continuously turn to the left until it is "stopping" or "going forward" in the required direction. Similarly, if the robot receives a "turning right" command, it will continuously turn to the right until a "stopping" or "going forward" command is received. For example, to make the robot turn to the right, two correctly classified trials (left and right) are required. If the first trial is classified as "left hand" while the second trial is classified as "right hand," the

issued command will be "turning right," and the robot will continuously turn to the right until a "stopping" or "going forward" command is received. However, if the first trial is classified as "left hand" and the second trial is also classified as "left hand," the issued command will be "turning left," and the robot will continuously turn to the left until a "stopping" or "going forward" command is received. Turning left or right does not change the tracked path of the robot as long as it continues to go in the required direction. Turning left rather than turning right simply increases the task time. However, if the first trial is classified as "right hand," the issued command will be "going forward." This erroneous command will change the path tracked by the robot. Table 5 shows the new successful command rates achieved after implementing this improved strategy. The highest and lowest successful command rates are 90% and 71.67%, respectively. The average successful command rate increased from 71.41% to 80.92% over all the subjects.

3.3 Navigation Performance

Figure 10 shows the path to be tracked and sequential snapshots of the simulated robot's navigation for a single attempt

Table 4 Successful command rates of all subjects

Subjects	Issued commands				Average (%)
	Walking forward	Turning left	Turning right	Stopping	
akm	90	56	75	90	78.00
khl	73	56	57	90	69.00
mjd	82	59	53	91	71.25
mhd	91	69	70	89	79.75
hsm	82	43	42	77	61.00
abm	80	82	77	96	83.75
ysf	72	60	48	86	66.50
khd	63	58	43	84	62.00
Average	79.13%	60.38%	58.13%	87.88	71.41%

Table 5 Successful command rates of all subjects after improvement

Subjects	Issued commands			Average (%)
	Walking forward	Turning left or right	Stopping	
akm	90	80	90	86.67
khl	73	77	90	80.00
mjd	82	82	91	85.00
mhd	91	78	89	86.00
hsm	82	56	77	71.67
abm	80	94	96	90.00
ysf	72	71	86	76.33
khd	63	68	84	71.67
Average	79.13%	75.75	87.88	80.92

(one navigation task). During a single attempt, the task time, distance traveled, and number of collisions were measured. Additionally, the number of total issued commands and successful command rate were calculated. These performance metrics from the BCI experiments were averaged over 20 attempts for each subject. As an illustrative example, we show the navigation results of the first subject, *akm*, in detail. Figure 11 shows the traces of 20 navigation attempts by EEGs of subject *akm* using the simulated robot, initiating at the starting position and ending at the destination. The robot's collisions are indicated with bold red marks. Table 6 summarizes the performance metrics of subject *akm*. The average performance metrics are as follows. The average collision number over 20 attempts is 0.3, and the average distance traveled is 23.69 m. The average time required by the robot to reach its destination is 130 s, the average number of issued commands is 17.9, and the average number of unsuccessful commands is 2.45. However, in all attempts, the robot successfully reached its destination.

Table 7 summarizes the navigation performance metrics of all subjects. The performance metric results vary from one subject to another. However, in all attempts, the robot successfully reached its destination. As shown in the table, subject *abm* presented the highest performance with a relatively lower number of collisions, shorter task time, shorter distance travelled, and fewer issued commands. This is because *abm* tended to have the highest classification accuracy and successful command rate (see Tables 2, 5). In contrast, subjects *hsm* and *khd* exhibited the poorest performance due to their lowest classification accuracy and successful command rates. Because the left-hand MI is responsible for turning the robot left or right (see Fig. 7), its incorrect classification forces the robot to go forward, which may lead to collisions and increased travel distance. The results of subjects *hsm* and *khd* verify this claim. Similarly, because right-hand MI is responsible for going forward, its incorrect classification

forces the robot to turn left or right, causing an increase in the task time required to reach its destination. The results of subject *khd* verify this claim. Higher classification accuracy leads to a lower number of required commands for reaching the destination with a lower number of unsuccessful commands. This becomes clear in mapping the obtained results in Table 2 with the results in Table 7. The performance metrics are averaged over all the subjects and shown in the last row of Table 7. Overall, higher classification accuracy with a well-developed postural-dependent control paradigm leads to sound performance, i.e., relatively fewer collisions, shorter task time, shorter distance travelled, and fewer issued commands. This is verified by the comprehensive results in Table 7.

For comparison with BCI control, the aforementioned metrics were calculated and averaged over five attempts using manual control. The averaged results of the navigation experiments based on BCI and manual control are shown in Table 8. During the manual control attempts, the robot was navigated through the same track (see Fig. 8) and achieved a distance traveled of 23.24 m without collisions. However, during the BCI control attempts, the robot travelled 23.92 m with an average of 0.65 collisions. The average task time was 85 s under manual control, and 170.1 s under BCI control. This is because, but not limited to, the fact that at least two seconds were required to process the mental task (imagination of right- or left-hand movement). Another reason for the increased task time under BCI control is classification errors. Thus, more commands are required to complete the task (reaching the destination), increasing the task time. Classification errors also increase the collision number and distance travelled. The practical successful command rate for manual control is 100% whereas that of BCI control is 79.3%, which is close to the theoretical value shown in Table 5. However, the robot always successfully reaches its final position (the success rate under BCI control is 100%).

4 Conclusion

This study describes a synchronous control system called BCI using two mental tasks (left-hand and right-hand motor imagery) for robot navigation in an unknown environment. We employ CSP with logarithmic band power instead of variance to form feature vectors. Linear discriminant analysis (LDA) is employed for classification. We developed a posture-dependent control architecture with four motion robot commands ("going forward," "turning left," "turning right," and "stopping") to translate acquired categories (right hand and left hand) into robot motion commands. The EEGs of eight healthy volunteer male subjects navigated a simulated robot to reach a destination point in a virtual environment.
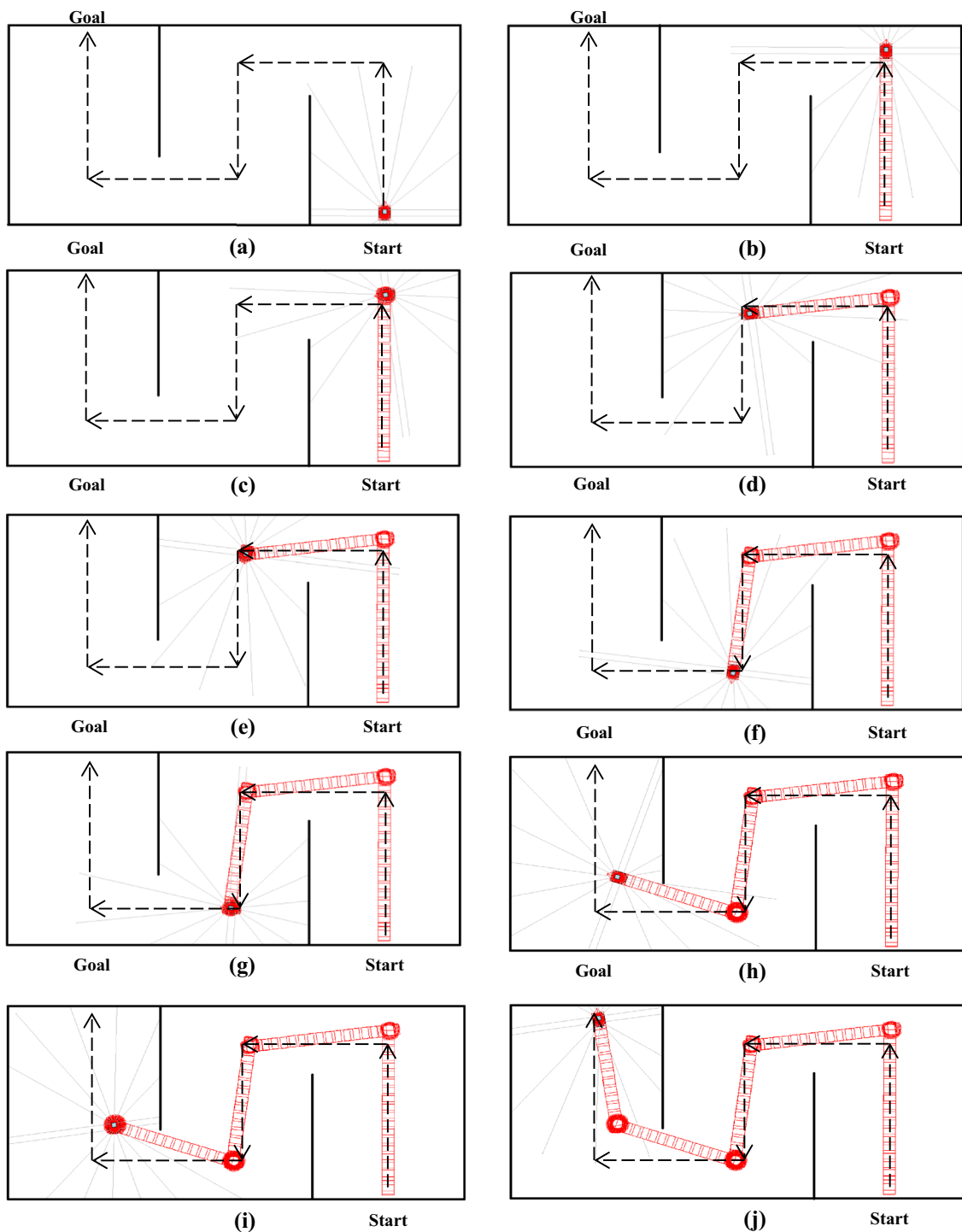
Fig. 10 Screen shots of the navigation of the simulated robot in one trial (**a**) Robot started at the starting position with the "stopping" state. (**b**) Robot was going forward after receiving "going forward" command, and then stopped after receiving "stopping" command (**c**) Robot was turning to the left after receiving "turning left" command. (**d**) Robot was going forward and then stopped again. (**e**) Robot was again turning to the left. (**f**) Robot again was going forward and then stopped. (**g**) Robot was turning to the right after receiving "turning right" command. (**h**) Robot again started to walk forward and then stopped. (**i**) Robot turned to the right. (**j**) Robot again started to walk forward and then stopped at the destination
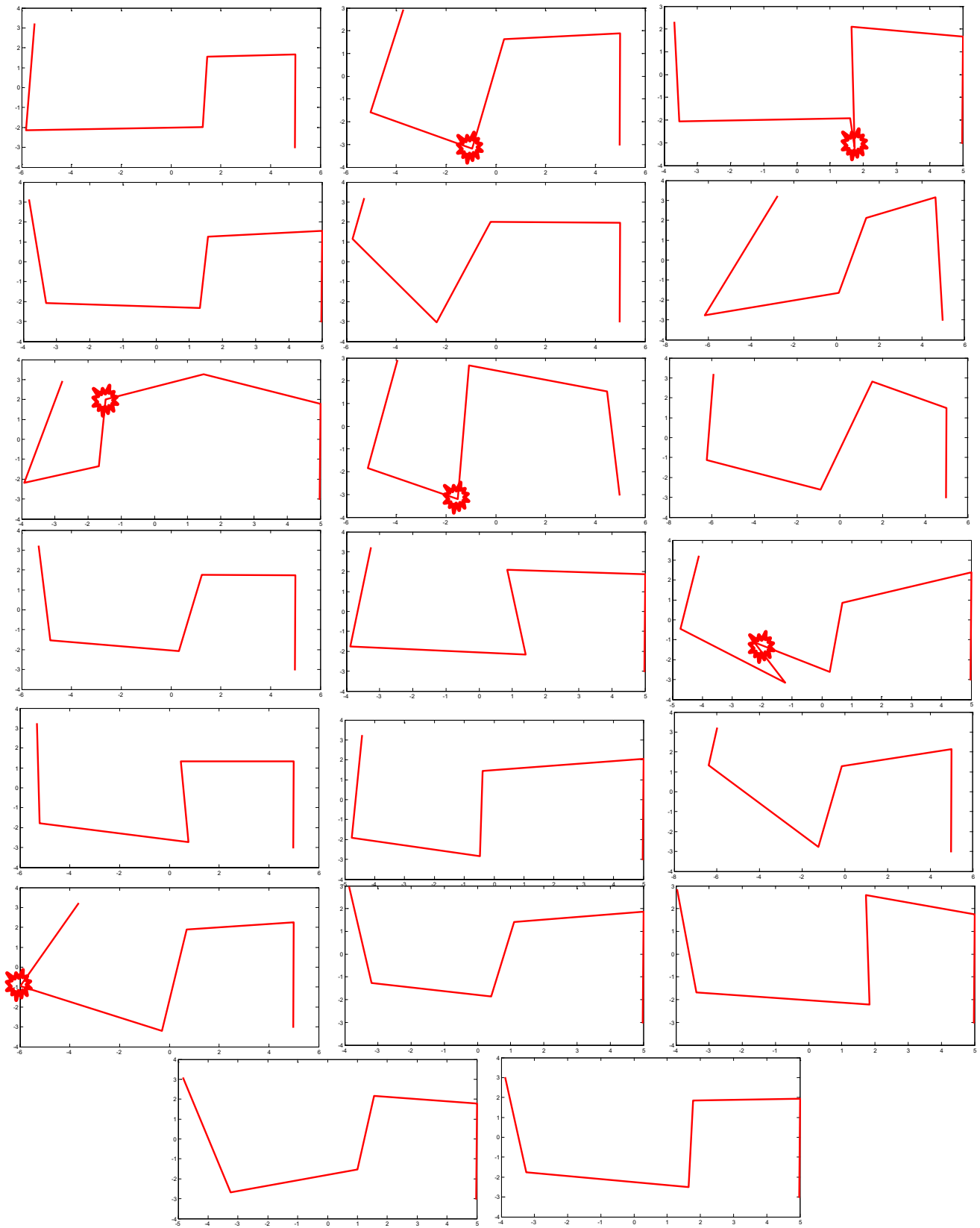
Fig. 11 Traces of twenty attempts during navigations of subject *akm* using the simulated robot starting from the starting position until reaching the destination. The collisions are indicated using bold red marker

Table 6 Navigation performance metrics of subject *akm* with 20 attempts

	Total commands (times)	Successful commands (times)	Unsuccessful commands (times)	Time (s)	Distance travelled (m)	Collisions (times)
Attempt#1	15	14	1	93	20.14	0
Attempt#2	17	15	2	122	22.79	0
Attempt#3	14	14	0	90	22.42	0
Attempt#4	15	14	1	99	22.32	0
Attempt#5	14	14	0	104	24.10	0
Attempt#6	15	14	1	125	24.35	0
Attempt#7	15	14	1	113	23.15	0
Attempt#8	17	15	2	120	25.73	1
Attempt#9	16	15	1	119	24.11	0
Attempt#10	14	14	0	100	22.46	0
Attempt#11	14	14	0	110	23.71	0
Attempt#12	25	24	1	213	26.85	1
Attempt#13	24	18	6	171	23.22	0
Attempt#14	26	19	7	192	26.89	0
Attempt#15	20	23	7	169	22.87	1
Attempt#16	20	21	9	180	24.96	1
Attempt#17	19	16	3	118	24.26	0
Attempt#18	22	18	4	114	23.70	1
Attempt#19	22	19	3	160	24.26	1
Attempt#20	14	14	0	89	21.52	0
Average	17.9	16.45	2.45	130.05	23.69	0.3

Table 7 Averaged navigation performance metrics of all subjects

Subjects	Total commands (times)	Successful commands (times)	Non-successful commands (times)	Task time (s)	Distance travelled (m)	Collisions (times)
akm	17.9	16.45	2.45	130.05	23.6905	0.3
khl	24.6	19.25	5.35	177.45	23.8675	0.65
mjd	20.75	17.55	4.2	146.95	23.701	0.65
mhd	17.5	15.95	1.55	119.35	23.175	0.4
hsm	34.7	25.35	9.35	207.55	25.755	1.3
abm	17.35	15.8	1.65	113.2	21.9375	0.3
ysf	24.95	19.4	5.55	176.65	24.41	0.65
khd	36.7	24.45	12.25	289.6	24.8055	0.95
Average	24.31	19.28	5.29	170.10	23.92	0.65

Table 8 Averaged navigation performance metrics based on BCI and manual control

	BCI	Manual
Collisions (times)	0.65	0
Distance travelled (m)	23.92	23.24
Task time (s)	170.10	85.00
Total commands (times)	24.31	14
Successful commands (times)	19.28	14
Successful commands rate	79.3%	100%

We evaluated the proposed system on three levels. The proposed feature extraction and classification methods were evaluated with a classification accuracy metric, while the developed posture-dependent control architecture was evaluated using a successful command rate metric. The performance of the overall system was validated along with that of the simulated robot on a predefined task using task time, distance travelled, number of collisions, and other metrics. Based on the average classification accuracy over all the subjects, CSP achieved better performance using LBP than

variance. The developed posture-dependent architecture achieves a successful command rate of ~ 80%, which is sufficient for navigating a robot with relatively few errors. Although the performance varies from subject to subject, the robot always successfully reaches its final position (achieves a 100% success rate). Finally, the developed mobile robot control system using BCI yielded promising results compared to manual controls. Future research directions include testing the proposed method using an actual robot instead of a simulated one. Moreover, we plan to consider an online approach in future work.

References

1. Shih, J. J., Krusienski, D. J., & Wolpaw, J. R. (2012). Brain-computer interfaces in medicine. *Mayo Clinic Proceedings, 87*(3), 268–279.
2. Anderson, R. A., Musallam, S., & Pesaran, B. (2004). Selecting the signals for a brain-machine interface. *Current Opinion in Neurobiology, l4*(6), 720–726.
3. Sanei, S., & Chambers, J. (2007). *EEG signal processing*. England: John Wiley & Sons.
4. Alonso, N., Fernando, L., & Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors, 12*(2), 1211–1279.
5. Dasgupta, S., Fanton, M., Pham, J., Willard, M., Nezamfar, H., Shafai, B., & Erdogmus, D. (2010). Brain controlled robotic platform using steady state visual evoked potentials acquired by EEG. *Proceedings of. 2010 Conference Record Forty Fourth Asilomar Conference Signals, System Computers*, California, USA (pp. 1371–1374).
6. Ortner, R., Guger, C., Prueckl, R., Grünbacher, E., & Edlinger, G. (2010). SSVEP-based brain-computer interface for robot control. *Proceedings of 12th International Conference Computers Helping People with Special Needs,* Vienna, Austria (pp. 85–90).
7. Pires, G., Castelo-Branco, M., & Nunes, U. (2008). Visual P300-based BCI to steer a wheelchair: A Bayesian approach. *Proceedings of IEEE Engineering Medicine Biol Soc*, British Columbia, Canada (pp. 658–661).
8. Choi, K., & Cichocki, A. (2008). Control of a wheelchair by motor imagery in real time. *Proc. 9th Int Conf Intell Data Eng Autom Learning*, (pp. 330–337).
9. Choi, K. (2011). Control of a vehicle with EEG signals in real-time and system evaluation. *European Journal of Applied Physiology, 112*(2), 755–766.
10. Shedeed, H. A., Issa, M. F., & El-sayed, S. M. (2013). Brain EEG signal processing for controlling a robotic arm. *8th International Conference on Computer Engineering & Systems (ICCES),* (pp. 152–157).
11. Müller-Putz, G. R., & Pfurtscheller, G. (2008). Control of an electrical prosthesis with an SSVEP-based BCI. *IEEE Transactions on Biomedical Engineering, 55*(1), 361–364.
12. Pfurtscheller, G., Solis-Escalante, T., Ortner, R., Linortner, P., & Muller-Putz, G. R. (2010). Self-paced operation of an SSVEP-based orthosis with and without an imagery-based "brain switch:" a feasibility study towards a hybrid BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 18*(4), 409–414.
13. Song, W., Wang, X., Zheng, S., & Lin, Y. (2014). Mobile robot control by BCI based on motor imagery. *Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2*(26–27), 383–387.
14. Barbosa, A. O. G., Achanccaray, D. R., & Meggiolaro, M. A. (2010). Activation of a mobile robot through a brain computer interface. *Proc Conf Rec 2010 IEEE Int Conf Robot Autom*, (pp. 4815–4821).
15. Varona-Moya, S., Velasco-Álvarez, F., Sancha-Ros, S., Fernández-Rodríguez, Á., Blanca, M. J., & Ron-Angevin R. (2015). Wheelchair navigation with an audio-cued, two-class motor imagery-based brain-computer interface system. *7th International IEEE/EMBS Conference on Neural Engineering (NER),* (pp. 174–177).
16. Fan, X. A., Bi, L., Teng, T., Ding, H., & Liu, Y. (2015). A brain–computer interface-based vehicle destination selection system using P300 and SSVEP Signals. *IEEE Transactions on Intelligent Transportation Systems, 16*(1), 274–283.
17. Leeb, R., Friedman, D., Muller-Putz, G., Scherer, R., Slater, M., & Pfurtscheller, G. (2007). Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: a case study with a tetraplegic. *Computational Intelligence and Neuroscience, 2007,* 1–7.
18. Tsui, C. S. L., & Gan, J. Q. (2007). Asynchronous BCI control of a robot simulator with supervised online training. *Proceedings of 8th International Conference Intellignet Data Engineering Automated Learn*, (pp. 125–134).
19. Tsui, C. S. L., Gan, J. Q., & Roberts, S. J. (2009). A self-paced brain–computer interface for controlling a robot simulator: An online event labelling paradigm and an extended Kalman filter based algorithm for online training. *Medical & biological engineering & computing, 47*(3), 257–265.
20. G.tec (2016) Advanced biosignal acquisition, processing and analysis. *Product catalogue*, g.tec.
21. Jasper, H. H., & Andrews, H. L. (1938). Electro-encephalography. III. Normal differentiation of occipital and precentral regions in man. *Archives of Neurology & Psychiatry, 39,* 95–115.
22. Djemal, R., Bazyed, A., Belwafi, K., Gannouni, S., & Kaaniche, W. (2016). Three-class EEG-based motor imagery classification using phase-space reconstruction technique. *Brain Sciences, 2016*(6), 36.
23. Perrin, X., Chavarriaga, R., Colas, F., Siegwart, R., & Millán, J. D. R. (2010). Brain-coupled interaction for semi-autonomous navigation of an assistive robot. *Robotics Autonomous Systems, 58*(12), 1246–1255.
24. Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Müller, K. R. (2008). Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine, 25*(1), 41–56.
25. Ramoser, H., Muller-Gerking, J., & Pfurtscheller, G. (2000). Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering, 8*(4), 441–446.
26. Townsend, G., Graimann, B., & Pfurtscheller, G. (2006). A comparison of common spatial patterns with complex band power features in a four-class BCI experiment. *IEEE Transactions on Biomedical Engineering, 53*(4), 642–651.
27. Mason, S. G., Bashashati, A., Fatourechi, M., Navarro, K. F., & Birch, G. E. (2007). A Comprehensive Survey of Brain Interface Technology Designs. *Annals of Biomedical Engineering, 35*(2), 137–169.

28. Brodu, N., Lotte, F., Lécuyer, A. (2011). Comparative study of band-power extraction techniques for Motor Imagery classification. *2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*, Paris (pp. 1–6).

29. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. England: John Wiley & Sons.

30. Refaeilzadeh, P., Tang, L., Liu, H. (2009). Cross-validation. *Encyclopedia of database system*. Berlin, Germany: Springer.