Med Biol Eng Comput (2014) 52:1007–1017
DOI 10.1007/s11517-014-1204-4

ORIGINAL ARTICLE

Motor imagery, P300 and error-related EEG-based robot arm movement control for rehabilitation purpose

Saugat Bhattacharyya · Amit Konar · D. N. Tibarewala

Received: 14 December 2013 / Accepted: 22 September 2014 / Published online: 30 September 2014

Abstract The paper proposes a novel approach toward EEG-driven position control of a robot arm by utilizing motor imagery, P300 and error-related potentials (ErRP) to align the robot arm with desired target position. In the proposed scheme, the users generate motor imagery signals to control the motion of the robot arm. The P300 waveforms are detected when the user intends to stop the motion of the robot on reaching the goal position. The error potentials are employed as feedback response by the user. On detection of error the control system performs the necessary corrections on the robot arm. Here, an AdaBoost-Support Vector Machine (SVM) classifier is used to decode the 4-class motor imagery and an SVM is used to decode the presence of P300 and ErRP waveforms. The average steady-state error, peak overshoot and settling time obtained for our proposed approach is 0.045, 2.8 % and 44 s, respectively, and the average rate of reaching the target is 95 %. The results obtained for the proposed control scheme make it suitable for designs of prosthetics in rehabilitative applications.

Keywords Brain–computer interfacing · Motor imagery · P300 · Error-related potential · Position control of robot arm · Electroencephalography

S. Bhattacharyya (✉) · D. N. Tibarewala
School of Bioscience and Engineering, Jadavpur University,
Kolkata 700032, India
e-mail: saugatbhattacharyya@live.com

A. Konar
Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata 700032, India

1 Introduction

Non-invasive methods of recording brain signals are widely used in brain–computer interfacing (BCI) research for control applications. From the available noninvasive means of measuring brain signals, electroencephalography (EEG) is widely used among BCI researchers because of its higher temporal resolution, portability, availability and inexpensiveness [21]. BCI technologies aim at decoding brain signals to detect the cognitive tasks executed by a user. A few well-known brain signal modalities include steady-state visually evoked potential (SSVEP), slow cortical potential (SCP), P300, event-related desynchronization/synchronization (ERD/ERS) and error-related potential (ErRP) [8, 22]. The selection of brain modality is an important issue in EEG-BCI analysis, and it depends on the cognitive task performed by the subject. ERD/ERS originates during motor planning, imagination or execution (also referred to as motor imagery signals) [26], ErRP has shown promising results in detection of visually inspired errors [7], and P300 are detected during identification of rare (single) state from a given set of multiple states [11]. Thus, these signals have relevance in our present study.

Motor imagery-based BCI are highly relevant for usage in rehabilitative applications which includes control of prosthetic devices [16] and controlling motion of wheelchairs [18]. Other applications include mind-driven motion control of mobile [3] and humanoid robots [6], thought-controlled navigation in virtual reality environment [2] and mind-controlled gaming [5]. Its merit lies in the automatic control of external devices without neuromuscular intervention.

One of the open areas of BCI research is designing a control strategy for rehabilitative applications. Early research on BCI employed a single signal modality, such as

P300, SSVEP and ERD/ERS for control applications [27, 28]. But in recent studies, hybrid BCIs (i.e., detection of at least two brain modalities in a simultaneous or sequential pattern) have been emphasized for control applications because of its effectiveness over conventional BCI [18]. For example, Pfurtscheller et al. [25] used a motor imagery-based switch to turn ON/OFF an SSVEP-based BCI. Long et al. [18] have used motor imagery and P300 signals for continuous 2D cursor control and the direction and speed of a wheelchair, respectively. Ferez et al. [12] used ErRP signal to detect errors in motor imagery-based (open loop) cursor-position control.

The paper aims at designing a distinctive scheme of asynchronous position control of a robot arm by decoding motor imagery signals about four different states: left, right, forward and no movement. On reaching the target (goal) position, the subject stops the movement of the robot arm by generating a P300 signal. Ideally, the subject would reach the target with zero positional error. But in practical scenario, errors would be generated during the control of movement by the robot which is detected by the presence of ErRP waveform in the EEG signal. The novelty of the present research lies in the architectural design for positional control of the robot arm. The real-time performance of the proposed EEG-driven position control scheme has been studied using four metrics: percentage success rate, peak overshoot, steady-state (positional) error and settling time.

The rest of the paper is organized as follows. Section 2 presents the methodology and experiments performed to design the EEG-driven position control scheme of a robotic arm. Section 3 provides the experimental results of the proposed scheme during offline training paradigm, online testing paradigm and real-time robot arm control. A summary of the proposed technique is discussed in Sect. 4 followed by the concluding remarks in Sect. 5.

2 Methods and experiments

In this section, we discuss the proposed control scheme required to move a robot arm, the experiments leading to the scheme and, finally, a design description on the real-time controller.

2.1 EEG data acquisition

A NeuroWin (manufactured by NASAN) EEG machine with 19 electrodes (FP1, FP2, F8, F4, Fz, F3, F7, T4, C4, Cz, C3, T5, T6, P4, Pz, P3, T3, O2, O1) is used to undertake the experiments (Fig. 1). The EEG signals are referenced to the right ear, and FPz location is grounded. The EEG is recorded using gold plated electrodes, and the impedances are kept below 5 kΩ. The EEG signals are amplified, sampled at 250 Hz, and band-pass filtered between 0.5 and 35 Hz.

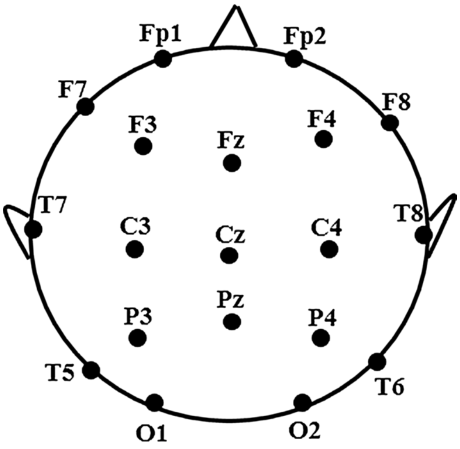

Fig. 1 10–20 electrode locations for the 19 selected channels

It is well established from previous literature [8] that motor imagery (MI) signals originate from the cortical motor areas (primary motor cortex, sensorimotor area and pre-motor cortex), error-related potential originates from the cingulate cortex, and strong P300 signals are generated from the parietal region. Based on the knowledge, we have processed the signals from C3 and C4 locations for MI detection, signals from Fz for error (ErRP) detection and signals from Pz for P300 detection.

2.2 Design of control scheme

The basic block diagram to the proposed scheme is given in Fig. 2a. The proposed control scheme allows the movement (translation/rotation) of the robot arm in any random order as intended by the user. Here, the control scheme employs three detectors for recognition of MI, P300 and ErRP brain states. The MI detectors control the directional motion of the robot arm. The P300 detector stops the motion of the robot arm when it reaches the goal position. The ErRP detector detects the presence of directional (because of MI detector) or positional (because of P300 detector) error in the incoming signal, and on detection, the control system makes the necessary correction. Here, the outputs of the different detectors are fed to the motor driver which controls the movement of the robot accordingly. The logic followed by the motor driver is shown in Fig. 2b.

Here, the robot arm is capable of turning clockwise and counterclockwise, and translating in the forward direction. The user plans the movement of the robot arm and the acquired signals at C3, and C4 electrodes are decoded to understand the motor imagination of the given task,

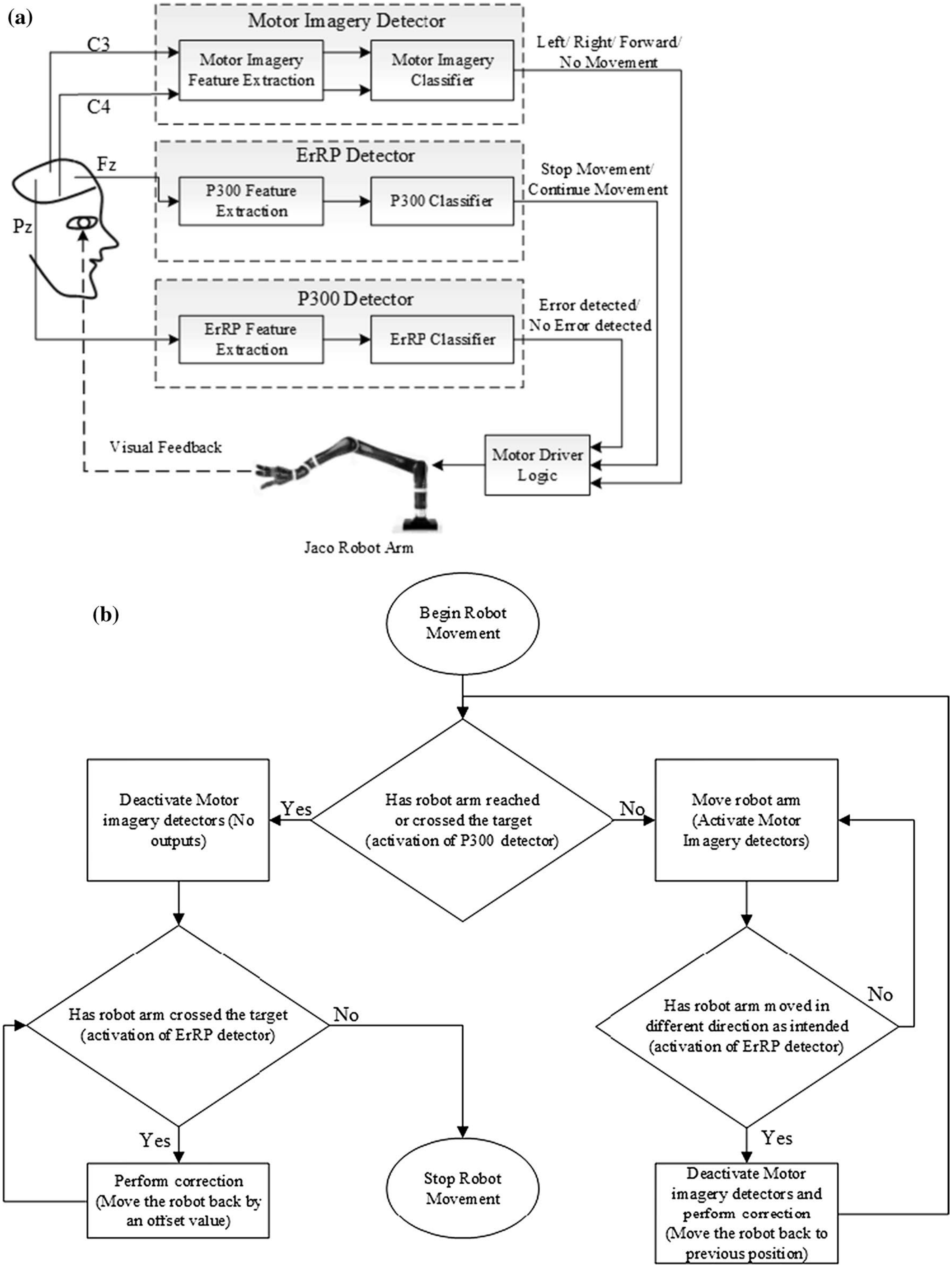
(a)
Motor Imagery Detector
C3
C4
Motor Imagery Feature Extraction
Motor Imagery Classifier
Left/ Right/ Forward/ No Movement
ErRP Detector
Fz
P300 Feature Extraction
P300 Classifier
Stop Movement/ Continue Movement
Pz
P300 Detector
ErRP Feature Extraction
ErRP Classifier
Error detected/ No Error detected
Visual Feedback
Motor Driver Logic
Jaco Robot Arm
(b)
Begin Robot Movement
Has robot arm reached or crossed the target (activation of P300 detector)
Yes
No
Deactivate Motor imagery detectors (No outputs)
Move robot arm (Activate Motor Imagery detectors)
Has robot arm crossed the target (activation of ErRP detector)
No
Yes
Perform correction (Move the robot back by an offset value)
Stop Robot Movement
Has robot arm moved in different direction as intended (activation of ErRP detector)
No
Yes
Deactivate Motor imagery detectors and perform correction (Move the robot back to previous position)

Fig. 2 **a** The block diagram of the proposed scheme. **b** The instruction sequence of the motor logic driver

i.e., move right, move left, move forward or no movement. These outputs produce their corresponding commands necessary to drive the robot, which is shown in Table 1. For example, if the user wants to move the robot counterclockwise, he would imagine moving his hand kinesthetically in the left direction.

The user observes the movement of the robot arm and visually detects the occurrence of error during the

Table 1 Mental tasks and their corresponding control commands

Mental task	Control commands
Left movement imagery	Move counterclockwise
Right movement imagery	Move clockwise
Forward movement imagery	Move forward
No Movement	Pause at current position and wait for next command
Focus on target position (P300)	Stop robot movement
Detection of ErRP	Perform necessary correction

movement of the robot. If the robot moves in a different direction not intended by the subject (directional error), the robot would traverse back to its previous position and the subject would need to rethink the command again. When the subject visually detects the link-end of the robot arm has reached or crossed the target, a P300 waveform is generated which stops the movement of the robot arm. If an ErRP is detected after the detection of P300 signal, it means that the link-end of the robot arm has crossed the desired position (positional error). In this case, the controller would attempt to realign the link-end by an offset (experimentally determined). The detection of the visual error is decoded from the EEG located at Fz, and the P300 signal is detected by the electrode at Pz. It must be noted from the scheme that when the P300 or ErRP detector is activated, then no outputs from the MI classifier are accepted by the motor driver logic, as explained in Fig. 2b.

2.3 Experiments and data processing

This section describes the experiments undertaken to design the three detectors: MI, P300 and ErRP, shown in Fig. 2a. Each subsection describes the stimuli designed for the offline and online sessions (as shown in Fig. 3), creation of the feature vectors [4] and the classifiers designed for each detector. Five normal right-handed subjects in the age-group of 22–28 years have participated in this experiment.

2.3.1 Design of the motor imagery detector

A visual stimulus is designed to train the user to perform the different MI tasks. Here, the stimuli for offline sessions begin with a 30-s recording of baseline EEG followed by a repetition of trials, where the instructions on the MI tasks are given to the user. The timing scheme of an individual trial, given in Fig. 3a, is as follows: a fixation cross for 1 s, followed by a MI task for 3 s and a blank for 2 s. The fixation cross alerts the user to get ready for subsequent motor imagination task. The MI task here instructs the user to perform any one of the four tasks: move left, move right, move forward and no movement. For training, the users performed the tasks in three separate sessions over every alternate days with 50 repetitions of each task in a random order.

The EEG signals acquired from the C3 and C4 locations are spatially filtered by Laplacian method [8] to remove the effect of neighboring electrode. Here, we subtract the

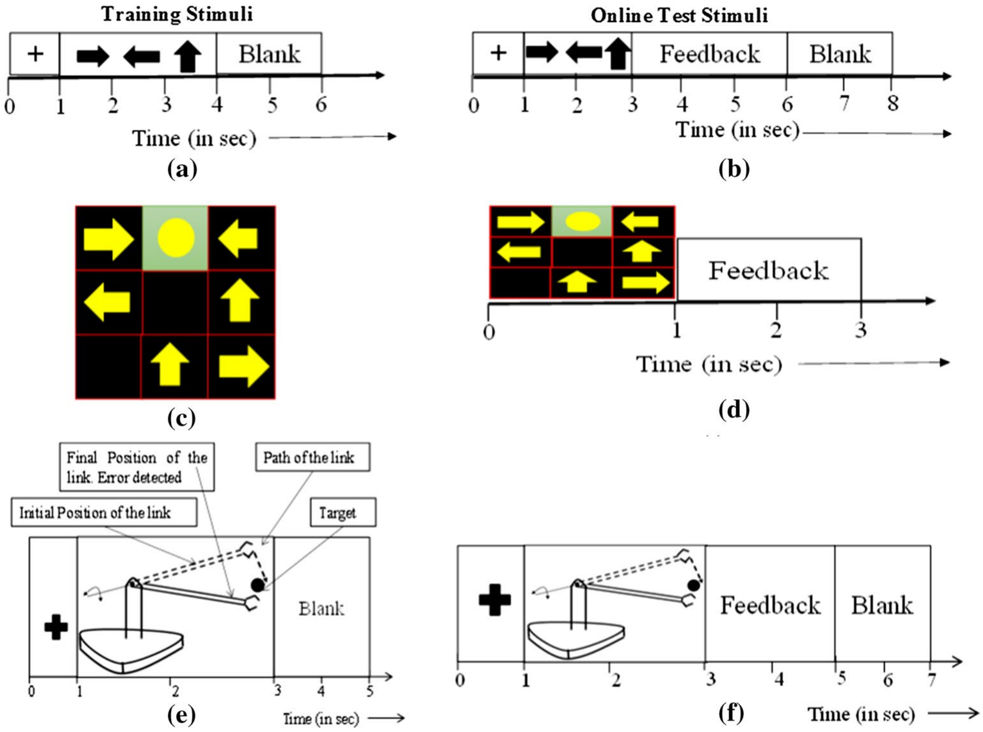

Fig. 3 The timing scheme of the visual stimuli. **a** during training for motor imagery task, **b** during online testing for motor imagery task, **c** stimuli used in P300 task, **d** during online testing of P300 task, **e** during training for error detection task, and **f** during online testing for error detection task. ("+" indicates fixation cross, arrows represent the motor imagination tasks, and feedback represents the feedback period)

average of signals acquired from F3, T7, P3 and Cz from the signal acquired from C3 and similarly, average of F4, Cz, P4 and T8 from C4. From existing literature [20], it is clear that relevant MI information is obtained from 8–12 Hz (μ-rhythm) and 16–24 Hz (central-β rhythm). Thus, the spatially filtered EEG signals are temporally filtered in the frequency range of 8–25 Hz. For this purpose, we have designed an IIR elliptical filter of order 6, pass-band attenuation of 1 dB and stop-band attenuation of 50 dB. The merit of selecting elliptical filter lies in its good frequency-domain characteristics of sharp roll-off, and independent control over the pass-band and stop-band ripples.

The features selected in this paper are the adaptive autoregressive parameters of the EEG from each electrode. Adaptive auto-regressive (AAR) model takes into account the non-stationary behavior of a signal by varying the autoregressive (AR) parameters with time. An adaptive autoregressive model of order p, $AAR(p)$, is described as,

$$x(k) = \sum_{i=1}^{p} a_i(k)x(k-i) + \eta(k) \tag{1}$$

with

$$\eta(k) = N\{0, \sigma_\eta(k)^2\} \tag{2}$$

where $x(k)$ is the kth sample of the series under observation, $\eta(k)$ is the zero-mean-Gaussian noise with variance, $\sigma_\eta(k)^2$, N is a function corresponding to the noise and $a_i(k)$ are the time-varying AR coefficients. As noted from (1), a current sample is predicted by past p samples and the new information introduced by the noise. Thus, $\eta(k)$ is also called the innovation process. There are several algorithms which are used for estimation of the AAR coefficients like least-mean-square (LMS) method, recursive-least-square (RLS) method, recursive AR (RAR) method and Kalman filtering [24].

In this study, we have employed the Kalman filter as the estimation algorithm, which can be summarized by the following equations:

$$e_k = y_k - \hat{\vec{a}}_{k-1}\vec{X}_{k-1} \tag{3}$$

$$\vec{k}_k = \frac{\vec{A}_{k-1}\vec{X}_{k-1}}{\vec{X}^T_{k-1}\vec{A}_{k-1}\vec{Y}_{k-1} + 1} \tag{4}$$

$$\vec{\eta}(k) = \vec{A}_{k-1} - \vec{k}^T_k\vec{X}^T_{k-1}\vec{A}_{k-1} \tag{5}$$

$$\vec{A}_k = \vec{\eta}(k) + \frac{uc.trace\left(\vec{A}_{k-1}\right)}{p}.\vec{I} \tag{6}$$

$$\hat{\vec{a}}_k = \hat{\vec{a}}_{k-1} + \vec{k}^T_k e_k \tag{7}$$

where e_k is the one-step prediction error, $\vec{k}_k$ is the Kalman gain vector, $\vec{I}$ is the identity matrix, $\vec{X}_k = \left[x_{k-1}, x_{k-2}, \ldots, x_{k-p}\right]^T$ is a vector of the values of the past samples, $\hat{a}_k = \left[\hat{a}_{1,k}, \hat{a}_{2,k}, \ldots, \hat{a}_{p,k}\right]^T$ is a vector of the AAR parameters, and $[.]^T$ is vector transpose. Further details on AAR with Kalman filter as estimator are discussed in [24]. In the present context, experiments undertaken reveal that an AAR model of order 6 and update coefficient = 0.0085 discriminates the MI tasks effectively. Further, the feature vectors are normalized in the range [−1, 1]. The final dimensions of the feature vector (for every session) are 200 trials × 2 electrodes × 6 AAR coefficients.

The feature vector thus prepared is then employed to train an AdaBoost classifier to produce the four MI states as outputs. AdaBoost, developed by *Freund and Schapire* [13], is the most influential boosting algorithm. In this family of ensemble technique, first, the weights of each training datum are uniform. After each iteration, the easily classified patterns are assigned lower weights and the difficult patterns are assigned higher weights, thus increasing the focus of the learners toward the difficult ones. After every iterations, the base learners prepare a new prediction rule and after N iterations, N prediction rules are prepared to construct the final distance discriminant, by which the unknown patterns can be recognized. The final prediction rule is equal to the weighted majority vote of all predictors, and the final accuracy of the classifier is effectively boosted.

For our study, we have employed Support Vector Machine (SVM) [1, 9] as the *base learner*. An SVM classifier maps the input vectors in high-dimensional feature space through some nonlinear mapping which can easily separate the data point vectors of two classes by a hyperplane of maximum margin.

Let $\vec{F}_i$ for $i = 1\ldots l$ represent a set of data points with corresponding class labels $y_i \in \{-1,1\}$. The SVM attempts to select a hyperplane to separate the data points based on their class labels. Let the equation of the separating hyperplane be $\vec{w}.\vec{F}_i + b = 0$, where the weight vector, $\vec{w}$ is perpendicular to the hyperplane and $|b|/\|\vec{w}\|$ is the perpendicular distance from the hyperplane to the origin, and $\|\vec{w}\|$ is the Euclidean norm of $\vec{w}$. Let d_+ (d_-) be the shortest distance from the separating hyperplane to the closest positive (negative) class label. Thus, the margin of the hyperplane is given by $d_+ + d_-$. Now, let all the training data satisfy the following constraints:

$$y_i(F_i.w_i + b) - 1 \geq 1. \tag{8}$$

The SVM classifier minimizes the margin $\|\vec{w}\|$ to obtain the optimal hyperplane with maximum margin, subject to constraints given in (8).

Here, we have employed an AdaBoost.M2 extension [10] for multi-class discrimination, which employs a one-against-one strategy for classification. The merit of this extension is it minimizes the pseudo-loss of the whole process.

The stimuli designed for online testing of the classifier are shown in Fig. 3b, which is similar to the one in Fig. 3a with the following modification: In each trial, a feedback period of 3 s is included after the motor activation period of 2 s. The testing is performed by the user over a single session with 10 repetitions of each MI task. From the advent of the signal acquisition, the AAR feature vector is prepared for every 1,000 ms of EEG data before the current time and fed to the pre-trained AdaBoost MI classifier. This classifier is used as the MI detector in our designed control scheme for real-time robot arm control.

2.3.2 Design of the P300 detector

The visual stimuli designed to train the classifier for P300 detection is based on the oddball paradigm [17], shown in Fig. 3c. As shown in Fig. 3c, the screen is divided into nine blocks where one of the blocks contains a yellow ball indicating the "stop" command and the rest comprises the MI tasks discussed in the previous section. Each block is highlighted randomly for 1,000 ms. The user is instructed to focus on the yellow ball when its corresponding block is highlighted and to ignore the rest of the cues. Each user performs the P300 experiment for 3 sessions, and in each session, the block containing yellow ball is highlighted 50 times and the rest of the blocks are highlighted 200 times.

It is known from existing literature [8, 17] that the P300 components in an EEG signal are elicited at around 300 ms from the time of response to a visual or auditory stimuli in the frequency band of 0.1–10 Hz. The incoming signals acquired from the Pz electrode are, first, band-pass-filtered by an elliptical filter of order 4 and bandwidth of 0.1–10 Hz. Then, we determine the averages [29] of four consecutive EEG signals over a period of 0–500 ms, and the resultant output is known as the "*feature vector*."

The feature vectors are then used to train a linear kernel-SVM classifier [1] to distinguish the P300 component from rest of the EEG. The aim of this classifier is to detect the P300 components, which is a peak at around 300 ms in the incoming EEG signals. Details on SVM are given in the previous section.

The testing session is similar to training session with an additional feedback period of 2 s after every instruction cue, as shown in Fig. 3d. In the test session, the block containing the yellow ball is highlighted 20 times and the rest of the blocks are highlighted 80 times. Then, we construct a P300 data vector at every 500 ms from the incoming EEG signals, as above. Next, we average the current data vectors with three previous data vectors and normalize it in the range $[-1, 1]$ to create the feature vectors, which are then fed to the pre-trained classifier as inputs. If y is the output class labels of the classifier, then when $y = 1$, the system decides to stop moving the robot arm and when $y = 0$, the system continues moving the robot arm.

2.3.3 Design of the ErRP detector

The visual stimulus, shown in Fig. 3e, trains the users to timely localize the occurrence of error through visual feedback. Here, the stimuli begins with a 30-s recording of baseline EEG followed by a repetition of trials, where the instructions to the user are given. The timing scheme of an individual trial is as follows: a fixation cross for 1 s, followed by an error detection task for 2 s and a blank screen for 2 s. As shown in Fig. 3e, during the error detection task, a simulated robot link moves toward the target. If the end point of the moving link crosses the target, an error is identified by the user. During training over three sessions, 50 trials have errors incorporated in them and 200 trials have correct responses. The task of the user is to detect the error trials.

Similar to P300 components, an ErRP waveform is a time-locked event occurring when the subject perceives an error during a task. The ErRP is characterized by a negative peak at around 150 ms followed by a positive peak at around 200–500 ms, after an error response is detected at a frequency range of 1–10 Hz [7, 22]. This feature of the ErRP waveforms is used for error detection in our study.

Here too, a filter of the same specification as the one used for P300 signal detection is used to filter the incoming signal from Fz electrode for a frequency range of 1–10 Hz. Similar to P300 component detection, the incoming EEG signals are averaged and used for training a linear kernel-SVM classifier.

The testing session is similar to training session with an additional feedback period of 2 s after every cue. Here too, an additional feedback period of 2 s is incorporated after the instruction cue in the online testing stimuli, shown in Fig. 2f. The test stimuli comprises 20 trials have error responses and 80 trials have correct responses.

The construction of the feature vectors and testing of the ErRP classifier are similar to the one used for P300 detection. When the ErRP detector yields an output $y = 1$, it means an error has occurred. Next, the detector notes the previous state of the system. If the MI detector was activated prior to the activation of the ErRP detector (occurrence of directional error), the system stops the movement of the robot arm and sends it back to its previous position. If the P300 detector was activated prior to the activation of the ErRP detector (occurrence of positional error), the system realigns the robot arm to the target position by an

Table 2 Classification accuracy (C.A. in %) and information transfer rate (ITR in bits/min) of five subjects for training and online testing of the motor imagery (MI), P300 and ErRP detectors

Detectors	Subject ID	Training			Online testing	
		C.A.	TPR	FPR	C.A.	ITR
MI	1	96.00	0.96	**0.00**	75.00	24.88
	2	92.00	0.92	0.06	78.00	23.12
	3	**100.00**	**1.00**	**0.00**	**83.00**	**19.42**
	4	96.00	0.91	0.08	80.00	21.75
	5	97.00	**1.00**	0.04	80.00	21.75
	Average	96.20	0.96	0.04	79.20	22.12
P300	1	88.00	0.85	0.13	80.00	23.47
	2	**100.00**	**1.00**	**0.00**	77.50	22.97
	3	**100.00**	**1.00**	**0.00**	**84.00**	**24.42**
	4	96.00	0.93	0.02	**84.00**	24.38
	5	92.00	0.82	0.06	82.00	23.93
	Average	95.20	0.92	0.04	81.50	23.83
ErRP	1	**98.00**	0.95	0.03	78.50	23.16
	2	**98.00**	**1.00**	**0.00**	**84.00**	**24.17**
	3	92.00	0.95	0.10	80.00	22.70
	4	93.00	0.97	0.09	80.00	23.26
	5	97.00	0.97	0.04	78.00	23.06
	Average	95.60	0.97	0.05	80.10	23.47

Best results marked in bold

offset. This step is the main difference between the ErRP detector and the P300 detector. Now if $y = 0$, then the system continues moving the robot arm.

2.4 Real-time robot arm controller design

In the present application, the control signal is required for a very small duration in the order of 300 ms for the correction of the positional error. Here, the controller logic combines the outputs of all three decoders to perform a movement task. Two different controllers have been designed for the proposed scheme. The first controller acts like a typical on–off controller, and it is deactivated with the recognition of P300 signal. The second controller is activated if an ErRP signal is detected. Here, we generate a control command to turn the robot arm in the reverse direction of its previous movement by a small offset. The controller functions, developed using the above philosophy, are given below.

Let the control signals for the two controllers at time t is $c_1(t)$ and $c_2(t)$, respectively, and the corresponding error at time t is $e(t)$, which is given by

$$e(t) = SP - CP(t) \tag{9}$$

where SP and $CP(t)$ are (visually triggered) set point and current position of the link-end of the robot arm at time t. Let, $P3(t) = 1$ indicate that P300 signal at time t. Then,

$$\begin{aligned} c_1(t) &= Au(t), \quad \text{if } e(t) > 0 \\ &= 0, \quad \text{if } e(t) \le 0 \text{ and } P3(t) = 1 \end{aligned} \tag{10}$$

Table 3 Average of the controller performance for five subjects

Parameters	Controllers	
	$\boldsymbol{c_1 + c_2}$	$\boldsymbol{c_1}$
Average success rate (in %)	**95**	80
Steady-state error (in %)	**0.045**	7
Peak overshoot (in %)	**2.8**	7
Settling time (in sec)	50	**38**

Best results marked in bold

$$\begin{aligned} c_2(t) &= 0, \quad \text{if } e(t) = 0 \\ &= \text{-} \mathit{off}.u(t) \text{ for } t \in [0, \delta], \quad \text{if } e(t) < 0 \text{ and } Err(t) = 1 \\ &= 0, \text{ for } t > \delta \end{aligned} \tag{11}$$

where $u(t)$ is a step function of magnitude A which is preset by the operator, *off* is the small offset value and δ is a very small positive number representing offset time of *Err*. The value of *off* and δ is set experimentally by taking average offset time of ten trials for each individual subject.

2.5 Performance measures used

The complete experiment is processed in a MATLAB 2012b environment run on a computer with the following specifications: Intel core i5 processor @ 3.10 GHz clock speed, 4 GB RAM and 64 bit Windows 7 operating system.

Performance of the detectors is measured on the basis of the true-positive rate (TPR), false-positive rate (FPR),

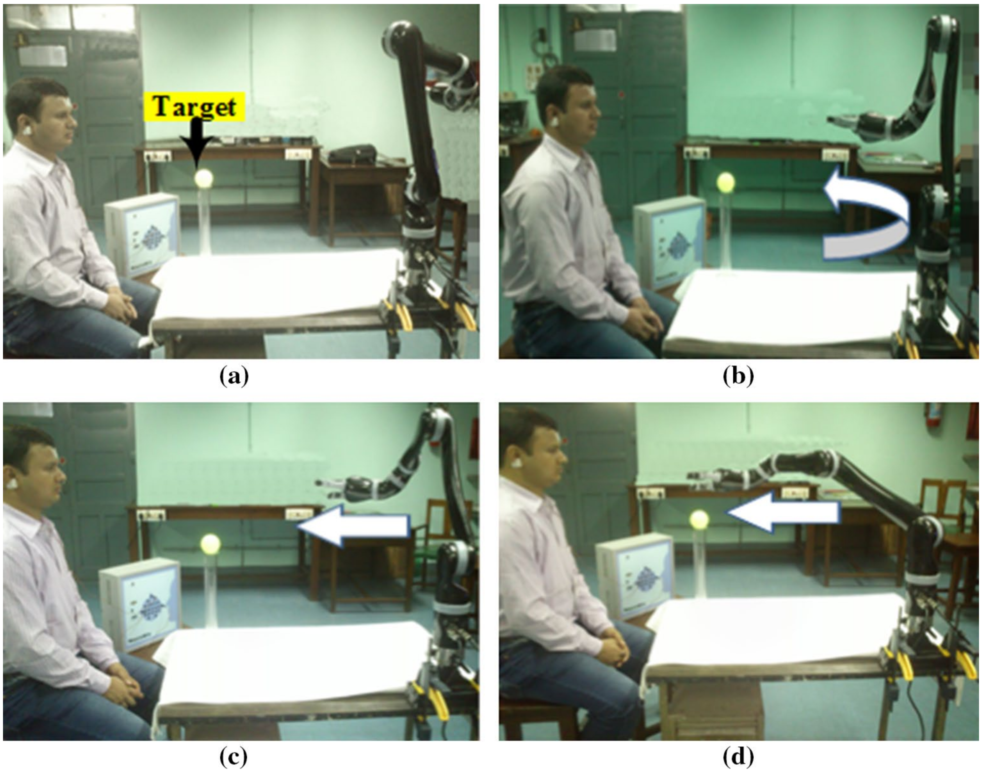

Fig. 4 An example of a subject controlling the Jaco robot arm to move toward the yellow ball (target position). **a** The initial position of the robot at the beginning of the experiment. **b** The subject rotates the robot in counterclockwise direction. **c** The subject moves instructs the robot to move forward. **d** The subject moves the robot forward bringing it closer to the target

classification accuracy (C.A.) and information transfer rate (ITR). An ideal BCI system would have a 100 % TPR and 0 % FPR value [23]. ITR (B_t) represents the bit rate of the method (in bits/min) and is given as

$$B_t = \left(\log_2 N + P\log_2 P + (1-P)\log_2 \frac{1-P}{N-1}\right) \times \frac{60}{T} \tag{12}$$

where N represents the number of possible states and P represents the classification accuracy between 0 and 1. T is the time needed to convey each action in second/symbol, i.e., time interval from the issue of a command to the classified output of the same.

Here, a Jaco robot arm [19] is used to study the real-time performance of the control scheme. The performance of the controller is examined here using three metrics: (i) percentage success rate, (i) steady-state error, (ii) peak overshoot and (iii) settling time. These parameters are defined below for ready reference.

Percentage Success Rate (*%SR*): The number of the times the subject is successful in reaching within 1 % of the target location for a given Z attempts.

Steady-state error (e_{ss})*:* The difference between a given set point and the time-response of the plant configured in a closed loop feedback system after a large (theoretically infinity) time interval following start-up of the plant [15].

Peak overshoot (M_P): The maximum deviation attained by the time-response of a plant from its steady-state value with constant input (usually a unit step function) divided by the steady-state value. M_P when expressed in percentage is called percentage M_P and is computed by

$$\% M_P = \frac{CP_p - CP_{ss}}{CP_{ss}} \times 100\ \% \tag{13}$$

where CP_p is the maximum value in CP (current process value) and CP_{ss} is its steady-state value [15].

Settling Time (t_s): The time taken by the response of a plant to reach and remain within 1 % of its steady-state value [15].

3 Results

This section first summarizes on the results obtained during offline training and online testing of the individual detectors. It is followed by a summary of the result obtained for our proposed scheme during real-time control of the Jaco robot arm.

3.1 Performance analysis of the detectors

The CA, TPR, FPR and ITR during training and online testing of the MI, P300 and ErRP detectors are summarized in Table 2. Separate classifiers are assigned to each subject.

As noted from Table 2, the classifier for subject 3 has the best training result, which is also reflected in the online testing result. From Table 2, it is noted that P300 classifiers of subject 2 and 3 yield the best result during training, and during testing, subject 3 reciprocates its training performance. It is noted from Table 2 that subject 2 yields the best results on training and test phase of the ErRP experiment.

These results suggest that the performance of each subject varies for the same number of experiments each has performed in this study. The reason for such variation is because each subject perceives the experiment in a different manner thus requiring different training time. It is also noted that the subject performing well during the training performs similarly during the online phase.

3.2 Performance of the real-time robot arm controller

The $\% \ SR$, e_{ss}, M_p and t_s are averaged for 5 subjects over 10 experimental instances to obtain the final result of the c_1 and c_2 controllers, given in Table 3. Also, a comparison of the proposed scheme with another scheme which contains only the c_1 controller is given in Table 3. As noted from Table 3, the introduction of ErRP signals to the BCI system improves the peak overshoot and steady-state error.

4 Discussion

In this paper, we have successfully controlled the movement of a robot arm using MI, P300 and ErRP components from an EEG signal. In this scheme, MI components are used to translate or rotate the robot arm, P300 components are decoded to stop the motion of the robot arm, and ErRP components are employed to recognize whether the MI decoder has misclassified the incoming input signals or the link-end of the robot has crossed the target position before the activation of the P300 decoder. Following the detection of the ErRP signal, the system either sends the robot arm back to its previous position or it attempts to realign the link-end of the robot arm with the desired goal position.

The average TPR and FPR obtained following the training of the MI, P300 and ErRP detectors are near to ideal value of 1 and 0, respectively. The classification accuracies obtained during training are 96.2, 95.2 and 95.6 % for the MI, P300 and ErRP detectors, but the accuracies are reduced during online training. Such variation in the accuracies may be attributed to the non-stationary behavior of EEG signals and the mental state of the subject at that

Table 4 Comparison with related works

Serial No.	Study by	Nature of work	Brain signals taken	Algorithms used	Concluding remarks
1	Proposed study	Position control of movement of a robot arm using motor imagery, P300 and ErRP signals	Motor imagery (mentioned in Table 1), P300 and ErRP	AAR, Adaboost-SVM, Moving averages, SVM	Average training accuracy for the three detectors is 94.8 %, Success rate is 95 %
2	Galan et al. [14]	Continuous mental control of a wheelchair using EEG-BCI	Left hand imagination, relaxation and words association to steer left, right and forward, respectively	Power Spectral Density and Linear Discriminant Analysis classifier	Average success rate of reaching the goal is 90 %
3	Lee et al. [17]	Control of movement of a small car using Ensemble Empirical Mode Decomposition as feature	SSVEP at 13, 14 and 15 Hz to steer robot forward, left and right, respectively	Ensemble Empirical Mode Decomposition and Matched filter detector	No. of valid detections is 51.13, ITR is 42.12 bits/min, Average training accuracy is 94.74 %
4	Long et al. [18]	Control of the direction and speed of a simulated or real wheelchair	Left hand, right hand, foot, idle and P300 to turn left, turn right, decelerate, accelerate and no commands	Common Spatial Pattern and Linear Discriminant Analysis	Average training accuracy of 75.4 %

instance. The offline training and online testing of the classifiers are performed using two different datasets acquired at different experimental sessions, and thus, their inter-session variability is high. This increases the chance of misclassification which is evident from our result. Thus, an optimal feature–classifier combination needs to be devised which could tackle such problems, and it is an interesting and open area in BCI research. We would be attempting to solve this problem in our next study.

The % *SR*, e_{ss}, M_p and t_s obtained for our proposed real-time controller are 95, 0.045, 2.8 % and 50 s, respectively. It may be noted from Table 3 that when we remove the ErRP component from our proposed controller, the average success rate reduces by a considerable amount from 95 to 80 %. This result shows how the incorporation of ErRP detector allows the subjects to correct any error occurred during the control task and provides an improvement on the overall results even if individually, the decoders perform poorly. Snapshots of position control of a Jaco robot arm to reach the fixed target (yellow ball) after complete execution of the control algorithm using the proposed scheme are shown in Fig. 4.

The control scheme designed in this paper differs from other similar works in its architectural design, as seen in Table 4. Also, results of our proposed work are compared with those of few other related studies in Table 4. It is observed from the table that our proposed approach performs better than the existing ones in terms of accuracy. Also, the inclusion of ErRP as control signals is a novel approach of this study.

5 Conclusion

Our proposed approach has used three different brain signals: MI, P300 and ErRP in positional control of the robot arm, and it is noted from the results that the incorporation of ErRP improves the performance of the control system. Here, performance of the proposed scheme greatly relies on the classifiers and the controller used in the system. A good classifier should have high classification accuracy and small computational overhead. We have used Adaboost-SVM and SVM classifiers to successfully control the movement of the robot arm, without sacrificing speed of computation. This scheme is designed for thought-based control of prosthetic arms, and experiments undertaken reveal that it serves real-time applications for its fast reaction time (of the order of 1 s). In future, we would like to improve the real-time recognition rate of the system. Also, we would attempt to control the individual links of the robot arm using the MI EEG. If successful, it would be more suitable for rehabilitative applications.

Acknowledgments I would like to thank University Grants Commission, India, University of Potential Excellence Programme (Phase II) in Cognitive Science, Jadavpur University and Council of Scientific and Industrial Research, India.

References

1. Alpaydin E (2004) Introduction to machine learning. The MIT Press, Cambridge, p 415
2. Bermudez i Badia S, Garcia Morgade A, Samaha H, Verschure PFMJ (2013) Using a hybrid brain computer interface and virtual reality system to monitor and promote cortical reorganization through motor activity and motor imagery training. IEEE Trans Neural Syst Rehab Eng 21(2):174–181
3. Bhattacharyya S, Sengupta A, Chakraborti T, Banerjee D, Khasnobish A, Konar A, Tibarewala DN, Janarthanan R (2012) EEG controlled remote robotic system from motor imagery classification. In: 3rd international conference on computing communication & networking tech. (ICCCNT 2012), Chennai, India, pp. 1–8
4. Bhattacharyya S, Sengupta A, Chakraborti T, Konar A, Tibarewala DN (2013) Automatic feature selection of motor imagery EEG signals using differential evolution and learning automata. Med Bio Eng Comp. doi:10.1007/s11517-013-1123-9
5. Bordoloi S, Sharmah U, Hazarika SM (2012) Motor imagery based BCI for a maze game. In: 4th international conference intelligent human computer interaction (IHCI), Kharagpur, India, pp 1–6
6. Chae Y, Jeong J, Jo S (2012) Toward brain-actuated humanoid robots: asynchronous direct control using an EEG-based BCI. IEEE Trans Robotics 28(5):1131–1144
7. Chavarriaga R, del Millan JR (2010) Learning from EEG error-related potentials in noninvasive brain-computer interfaces. IEEE Trans Neural Syst Rehabil Eng 18(4):381–388
8. Dornhege G (2007) Towards brain-computer interfacing. MIT Press, Cambridge
9. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York
10. Eibl G, Pfeiffer KP (2001) Analysis of the performance of AdaBoost.M2 for the simulated digit-recognition-example. In: 12th European conference on machine learning. EMCL'01, pp 109–120
11. Farwell LA, Donchin E (1988) Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalogr Clin Neurophysiol 70(6):510–523
12. Ferrez PW, Millán JR (2008) Simultaneous real-time detection of motor imagery and error-related potentials for improved BCI accuracy. In: 4th international brain-computer interface workshop & training course, pp 197–202
13. Freund Y, Shapire RE (1997) A decision-theoretic generalization of online learning and an application to boosting. J Comput Syst Sci 55(1):119–139
14. Galan F, Nuttin M, Lew E, Ferrez PW, Vanacker G, Philips J, Millan JDR (2008) A brain-actuated wheelchair: asynchronous and non-invasive brain-computer interfaces for continuous control of robots. Clin Neurophy 119:2159–2169
15. Kuo BC (2009) Automatic control systems, 8th edn. Wiley, Delhi
16. Li YN, Zhang XD, Huang ZX (2012) A practical method for motor imagery based real-time prosthesis control. In: 2012 IEEE international conference on automation science and engineering, Seoul, pp 1052–1056
17. Li Y, Pan J, Wang F, Yu Z (2013) A hybrid BCI system combining P300 and SSVEP and its application to wheelchair control. IEEE Trans Biomed Eng 60(11):3156–3166

18. Long J, Li Y, Wang H, Yu T, Pan J, Li F (2012) A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair. IEEE Trans Neural Syst Rehabil Eng 20(5):720–729
19. Maheu V, Frappier J, Archambault PS, Routhier F (2011) Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities. In: IEEE international conference on rehabilitation robotics (ICORR), Zurich, pp. 1–5
20. Mason SG, Birch GE (2003) A general framework for brain-computer interface design. IEEE Trans Neural Syst Rehabil Eng 11(1):70–85
21. Millán JR, Rupp R, Möller-Putz GR, Murray-Smith R, Giugliemma C, Tangermann M, Vidaurre C, Cincotti F, Köbler A, Leeb R, Neuper C, Möller KR, Mattia D (2010) Combining brain-computer interfaces and assistive technologies: state-of-the-art and challenges. Front Neurosci 4:1–15
22. Nieuwenhuis S, Ridderinkhof KR, Blom J, Band GPH, Kok A (2001) Error-related potentials are differentially related to awareness of response errors: evidence from an antisaccade task. Psychophysiology 38:752–760
23. Obermaier B, Neuper C, Guger C, Pfurtscheller G (2001) Information transfer rate in a five-classes brain-computer interface. IEEE Trans Neural Syst Rehabil Eng 9(3):283–288
24. Pfurtscheller G, Neuper C, Schlogl A, Lugger K (1998) Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. IEEE Trans Rehabil Eng 6(3):316–325
25. Pfurtscheller G, Solis-Escalante T, Ortner R, Linortner P, Muller-Putz GR (2010) Self-paced operation of an SSVEP-based orthosis with and without an imagery based brain switch. IEEE Trans Neural Syst Rehabil Eng 18(4):409–414
26. Qiang C, Hu P, Huanqing F (2005) Experiment study of the relation between motion complexity and event-related desynchronization/synchronization. In: 1st International conference on neural interface and Control 2005, pp 14–16
27. Rebsamen B, Guan C, Zhang H, Wang C, Teo C, Ang MH, Burdet E (2010) A brain controlled wheelchair to navigate in familiar environment. IEEE Trans Neural Syst Rehabil Eng 18(6):590–598
28. Royer AS, Doud AJ, Rose ML, He B (2010) EEG control of a virtual helicopter in 3-dimensional space using intelligent control strategies. IEEE Trans Neural Syst Rehabil Eng 18(6):581–589
29. Tsay RS (2010) Analysis of financial time series, 3rd edn. Wiley, New Jersey