

Asynchronous P300-Based Brain–Computer Interfaces: A Computational Approach With Statistical Models

Haihong Zhang*, *Member, IEEE*, Cuntai Guan, *Senior Member, IEEE*, and Chuanchu Wang, *Member, IEEE*

Abstract—Asynchronous control is an important issue for brain–computer interfaces (BCIs) working in real-life settings, where the machine should determine from brain signals not only the desired command but also when the user wants to input it. In this paper, we propose a novel computational approach for robust asynchronous control using electroencephalogram (EEG) and a P300-based oddball paradigm. In this approach, we first address the mathematical modeling of target P300, nontarget P300, and noncontrol signals, by using Gaussian distribution models in a support vector margin space. Furthermore, we derive a method to compute the likelihood of control state in a time window of EEG. Finally, we devise a recursive algorithm to detect control states in ongoing EEG for on-line application. We conducted experiments with four subjects to study both the asynchronous BCI's receiver operating characteristics and its performance in actual online tests. The results show that the BCI is able to achieve an averaged information transfer rate of approximately 20 b/min at a low false positive rate (one event per minute).

Index Terms—Asynchronous control, brain–computer interface, electroencephalogram (EEG), P300.

I. INTRODUCTION

THE BRAIN–COMPUTER interface (BCI) is an emergent multidisciplinary technology that allows a brain to control a computer directly, without relying on normal neuromuscular pathways [1], [2]. The most important applications of the technology are mainly meant for the paralyzed people who are suffering from severe neuromuscular disorders, as BCI potentially provides them with communication, control, or rehabilitation tools to help compensate for or restore their lost abilities.

Among various brain signal acquisition methods, the electroencephalography (EEG) is of particular interest to the BCI community [1], [3]–[5]. The EEG records the electrical brain signal from the scalp, where the signal originates from postsynaptic potentials, aggregates at the cortex, and transfers through the skull to the scalp [6]. The EEG bears merits as it is non-invasive, technically less demanding, and widely available at relatively low cost [4]. On the other hand, it also brings great challenges to signal processing and pattern recognition, since it

has relatively poor signal-to-noise ratio and limited topographical resolution and frequency range [7].

An important issue in EEG-based BCIs is *asynchronous control*, i.e., the machine should be able to infer from the EEG whether the user intends to operate the interface (this state is referred to as *the control state* hereafter) or not to operate (this state is referred to as *the noncontrol state* hereafter). However, traditional EEG-based BCIs work in a *synchronous* style, assuming that the user is always in the control state. Thus, the machine is continuously translating concurrent EEG signals to certain control commands. If the user does not intend to control the interface at all, false interpretations of the brain signal and false actions in the BCI will occur. On the other hand, a user-friendly BCI should allow the user to freely switch, without the aid of any other external inputs, between the control state and the noncontrol state without causing false actions. Thus, asynchronous control is desirable [8]. The key of asynchronous control in an EEG-based BCI, from the signal processing viewpoint, is an effective computational approach to distinguishing between the EEG in the control state and that in the noncontrol state.

Recent years have seen an increasing research interest in asynchronous control [8]–[10]. These studies were focused on the detection of the motor imagery signal. However, there have been few studies on asynchronous control using another important brain signal, the P300 [11].

The P300 is an endogenous, positive polarity component of the evoke-related-potential (ERP) elicited in the brain in response to infrequent/oddball auditory, visual, or somatosensory stimuli in a stream of frequent stimuli. Farwell and Donchin [12] first demonstrated the use of P300 for BCI in a so-called *oddball paradigm*. In the paradigm, the computer displays a matrix of cells representing different letters, and flashes each row and column alternately. A user trying to input a letter needs to pay attention to the letter for a short while. In this process, when the row/column containing the intended letter flashes, a P300 will be elicited in EEG and may be detected by an appropriate algorithm. Thus, by comparing brain's responses to the flashing buttons, the computer is able to determine which the target letter is. For people with visual impairments, this paradigm can be extended by using auditory or tactile stimuli [13].

The signal processing community has been playing an increasingly important role in developing faster and more robust BCI systems. And a great deal of efforts have been paid to pre-processing, feature extraction, and pattern classification of P300 EEG. For example, people studied signal processing methods to

Manuscript received July 20, 2007; revised November 23, 2007. *Asterisk indicates corresponding author.*

*H. Zhang is with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 119613, Singapore (e-mail: hhzhang@i2r.a-star.edu.sg).

C. Guan and C. Wang are with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 119613, Singapore (e-mail: ctguan@i2r.a-star.edu.sg; ccwang@i2r.a-star.edu.sg).

Digital Object Identifier 10.1109/TBME.2008.919128

decompose the multielectrodes EEG signals into some special components. Those components have fixed scalp distributions and sparsely activated and maximally independent time courses that contribute to P300. A typical method known as independent component analysis (ICA) was widely studied [14]–[17]. However, as argued in [18], application of ICA to EEG analysis remains a challenging problem because of the nonstationarity of EEG signals, as well as the multitude of active brain sources contrasting with the relative paucity of sensors.

For classification of P300 against non-P300 signals, a variety of algorithms were studied, ranging from stepwise-discriminant analysis [12] to recently prevalent classifiers using support vector machines (SVMs) [19]–[22]. An interesting comparative study on five established P300 classification techniques is given in [23], with the results indicating that all the techniques can attain acceptable performance levels.

However, as stated earlier, prior P300-based BCIs were only tested in synchronized (or synchronous) control environments. These systems were not able to determine from EEG if a user is in the control state. This implies inconvenient human–machine interactions. As argued in [24], this may also cause significant user frustration.

The present paper is the first attempt to address the issue of asynchronous control in the P300-based BCI domain. In particular, our objective is to make the system capable of inferring, from the ongoing EEG, if a user intends to input a command through the BCI at each time point. To this end, we devise a computational approach that comprises two major parts.

First, we develop a method to compute the likelihood of the control state in a time window of EEG. Due to a large number of variables and high intertrial variability in the EEG, it is difficult to model the probability functions directly on the multichannel time series of EEG signals. Instead, we transform the EEG signal into a univariate feature space created by a SVM, while preserving the discriminative information for P300 detection. In the univariate feature space, we employ user-specific Gaussian models for target P300, nontarget P300, and noncontrol signals. Based on these statistical models, we further derive a likelihood model of control/noncontrol states given a period of EEG.

Second, we devise an algorithm to detect control states in the ongoing EEG. This enables the system to automatically compute when the user tries to input a command through the interface. The algorithm consists of recursive steps enumerating possible time windows, and determining if the control state persists in one of the windows by comparing the likelihood against a preset threshold. The threshold here determines an appropriate tradeoff/balance between the speed and the accuracy and the false positive rate of the interface.

To assess the proposed BCI, we conducted experiments with four subjects, using both offline and online studies. The offline study allowed us to evaluate the asynchronous BCI's receiver operating characteristics (ROC), with a protocol following prior arts in asynchronous BCI studies [10], [25]. The online study was carried out using a real-time EEG processing platform that allowed assessing the performance of the BCI in actual asyn-

chronous scenarios. The results from offline analysis suggest that our approach can provide satisfactory performance for asynchronous control. In particular, the BCI is able to effectively transfer 20 bits information per minute, at a false positive rate (the occurrence rate of false detections in the noncontrol state) as low as 1/min. The online results indicate that, in the real situation, it can achieve comparable performance, i.e., 15 b/min at 0.71 false positive per min.

The rest of the paper is organized as follows. Section II describes related neurophysiological properties of the P300 for BCIs. Section III illustrates the system setup, defines essential terms in the P300-based asynchronous control. Section IV elaborates our probability models for EEGs in the control and the noncontrol states, followed by the description of the control state detection algorithm in Section V. The experiment is described in Section VI, with the data analysis elaborated in Section VII. Concluding remarks are given in Section IX.

II. BRIEF NEUROPHYSIOLOGICAL BACKGROUND OF P300

The P300 is generally connected to the cognitive processes of decision making, context updating, and the assessment of stimulus relevance [26]. However, it still remains a controversial topic that brain region actually serves as the generator of the P300, despite a vast body of related research thus far using various methodologies [27]. Nevertheless, evidence from various studies suggests that there may be multiple generators [26]. Intracranial recording and lesion studies often demonstrate the correlation between the P300 and specific activities in the temporo-parietal region [28]. Moreover, functional magnetic resonance imaging (fMRI) studies consistently identified that a “target detection” network distributed over mainly parietal and inferior frontal lobes may make a critical contribution to the scalp P300 [29], [30].

The P300 occurs at a latency of 300–600 ms after a target “oddball” stimulus and has a parietal distribution on the scalp. It is widely accepted that the amplitude of the P300 varies directly with the relevance of the eliciting events and inversely with the probability of the stimuli (or the interstimulus interval) [12]. And by increasing the interval, a modified paradigm is able to provide higher communication speed [31]. Exogenous factors such as stimulus size, stimulus duration, and eccentricity, however, may not give rise to significant changes in the waveforms of P300 [32], [33]. This enables flexible P300 BCI design with one trained P300 model applicable to different P300 interfaces for the same subject.

Prior studies have indicated that virtually all subjects under the oddball paradigm will generate a P300. For example, in [34], four wheelchair-bound disabled subjects (three with complete paraplegia) were able to generate P300, and achieved comparable “typing” speed to that of two able-bodied subjects using a P300 speller. Furthermore, three amyotrophic lateral sclerosis (ALS) patients participated in a study in [35] with a four-choice oddball paradigm. And it turned out that, for two of the three subjects, the elicited responses were classified accurately enough to control the BCI.

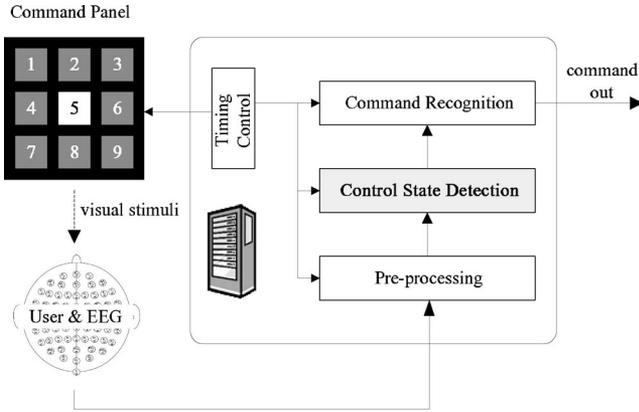


Fig. 1. Schematic graph of an asynchronous P300-based BCI system.

III. SYSTEM OVERVIEW AND PROBLEM FORMULATION

In this paper, we propose a P300-based BCI framework, as illustrated in Fig. 1. As stated earlier, the system flashes each button successively in a random order. While a button flashes, the concurrent EEG signal is captured by an amplifier and a data acquisition device. The captured signal is transferred to the computer, where it is processed and interpreted.

In the P300 signal processing, the EEG data are organized in terms of *epochs* and *rounds*.

- 1) *Epoch*: An epoch is associated with a particular button, and denotes the EEG segment relevant to the flashing event of that button. The button can be either attended or unattended. Thus, the epoch contains information of the brain's response to the particular stimulation. In this study, we select the time segment from 150 to 500 ms after the onset of a flashing to form an epoch.
- 2) *Round*: A round is a complete cycle in which every button flashes once and only once. Thus, each round contains N_s epochs, where N_s is the number of buttons on the display. Again, the chronological order of the button flashing events is randomly generated in a round, so as to produce "oddball" stimuli.

Suppose there are N_s buttons one can select on the interface display. If the i th button flashes in the t th round, we denote the epoch by s_i^t . The complete t th round consists of N_s epochs: $S = \{s_i^t\}$, where $i = 1, \dots, N_s$. Fig. 3(a) illustrates the organization of epochs in multiple rounds.

The control state detection essentially requires an automatic way to decide, from one round or multiple rounds of EEG epochs, if the user pays continuous attention to any button. Though it is possible to make such a decision with a single round, one usually uses multiple rounds to deal with the very low signal-to-noise ratio of single P300 epochs [33], [34]. In other words, P300-based BCIs often require the user to pay continuous attention to the desired button through a few rounds.

Now, we denote the control state by Ξ , in which the user pays attention to a particular button for a few rounds continuously. We denote the noncontrol state by Ψ , in which the user does not focus on any button on the interface. Thus, an epoch may fall

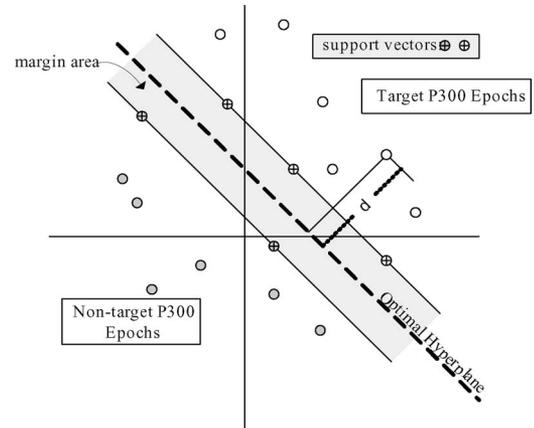


Fig. 2. Basic idea of support vector machines. In this paper, each small circle represents the feature vector of an EEG epoch.

into one of three classes: target epochs, nontarget epochs, and garbage epochs.

- 1) *Target epochs* Θ : Epochs associated with flashings of the target button (to which the user is attending).
- 2) *Nontarget epochs* Φ : Epochs associated with flashings of nontarget buttons (to which the user is not attending).
- 3) *Noncontrol epochs* O : All epochs in the noncontrol state.

With this definition, a control pattern and a noncontrol pattern can be illustrated as in Fig. 3(a) and (b), respectively.

Hence, in P300-based asynchronous control, it is essential to detect the occurrence of control patterns like Fig. 3(a) in the ongoing EEG signal. To this end, we take three steps to devise our computational approach.

- 1) Develop the statistical models for three types of EEG epochs (target, nontarget, or noncontrol epoch).
- 2) Develop the probability models for control and noncontrol states for a given multiround EEG.
- 3) Develop an algorithm which detects the occurrence of control state in ongoing EEG.

The three steps are elaborated in the subsequent sections.

IV. PROBABILITY MODELS OF EEG SIGNALS IN ASYNCHRONOUS CONTROL

A. Statistical Epoch Models

Consider a multichannel (m_c) EEG epoch consisting of m_c courses of m_t time samples. To represent the epoch, we form a feature vector \mathbf{x} by concatenating all the time courses as well as their dynamic features [22]. Usually the feature vector \mathbf{x} contains a large number of elements/variables. On the other hand, the available training samples are usually limited in amount. For reliable distribution density function estimation, therefore, it is preferable to first transform the samples into a low dimensional space. And, in this paper, we consider the margin space created by a support vector machine (SVM). The SVM is now a well-known classification method in which the principle is to seek maximal margin between two classes. The basic idea of SVM is illustrated in Fig. 2. Here, the symbol d represents the distance from a sample to the optimal hyperplane. The

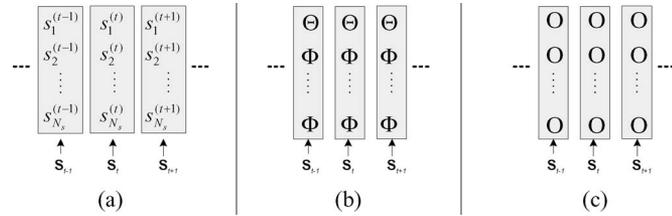


Fig. 3. Matrices of epoch-round structure of EEG signals. (a) Each column represents a round consisting of N_s epochs. (b) In this example, the attended button corresponds to the top row, in which all the epochs are target P300 (Θ). The rest epochs are nontarget P300 (Φ). (c) In a noncontrol pattern, all are noncontrol epochs (O). Epoch-round matrix.

fundamental of the machine lies in Cover's theorem [36] on the separability of patterns: a complex classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space. In SVM, the inner product in the high-dimensional space takes a form of sum of kernel functions, and the distance to the separate hyperplane is given by

$$d = h(\mathbf{x}) = \sum_{i=1}^N a_i k(\mathbf{x}, \mathbf{x}_i) + b \quad (1)$$

where \mathbf{x}_i denotes the i th support vectors. And the parameters $\{a_1, \dots, a_N, b\}$ can be learned from labeled data in the training set [37].

Of particular interest to this research is the modeling of posterior probability of the SVM score (d) for each epoch type. An early study in [38] has suggested that Gaussian functions can provide a good approximation to the distributions of the SVM margin d . Thus, we use the following Gaussian models to describe the conditional probability density of the margin given the epoch type (Θ , Φ , or O)

$$p(d|\Theta) = \mathcal{N}(d - \mu_\theta, \sigma_\theta^2) \quad (2)$$

$$p(d|\Phi) = \mathcal{N}(d - \mu_\phi, \sigma_\phi^2) \quad (3)$$

$$p(d|O) = \mathcal{N}(d - \mu_o, \sigma_o^2) \quad (4)$$

where the parameters can be simply learned from the training samples using the conventional maximum *a posteriori* (MAP) method.

Using Gaussian kernels, the SVM is prone to produce biased distributions of scores for the training set [38]. In particular, the distribution density function for the training set often exhibits two sharp peaks corresponding to SVM margins. A simple solution is to use a separate data set that is independent of the training set, for the estimation of the distribution density functions.

B. Likelihood Model for Control/Noncontrol State

Consider a multiround signal as shown in Fig. 3. In the control state, the brain will generate a single row of Θ (target) epochs with the rest being Φ (nontarget). Let $P(\Xi, R_i)$ be the probability of the control state and the user being attending to the R_i button. By Bayesian rule, it is straightforward to have the

likelihood of the control state (Ξ)

$$\begin{aligned} P(\Xi|D) &= \frac{p(\Xi, D)}{p(D)} = \frac{\sum_{i=1}^{N_s} p(\Xi, R_i, D)}{p(D, \Xi) + p(D, \Psi)} \\ &= \frac{\sum_{i=1}^{N_s} p(D|\Xi, R_i)P(\Xi, R_i)}{\sum_{j=1}^{N_s} p(D|\Xi, R_j)P(\Xi, R_j) + p(D|\Psi)P(\Psi)} \end{aligned} \quad (5)$$

where D denote the ensemble of d_{ij} , d_{ij} being the SVM score of the i th epoch in the j th round (see Fig. 3).

The variables $P(\Xi)$ and $P(\Psi) = 1 - P(\Xi)$ represent the *a priori* probabilities that a subject is in the control/noncontrol state. However, their values shall be determined by the particular application scenario. For example, a user may stay in the control state more often and longer [i.e., larger $P(\Xi)$] when the BCI is used for typewriting than for controlling a TV set. In the current study, we assume an even probability: $P(\Xi) = P(\Psi) = 0.5$. Furthermore, we assume an equal probability of the buttons being attended to: $P(\Xi, R_i) = 1/N_s P(\Xi)$.

The $p(D|\Xi, R_i)$ represents the conditional probability density of an observation D (actually it is the transformed EEG in the support vector machine space) given that the user is attending to the R_i button (thus being in the control state). Now, we assume that given the user's state, the EEG epochs are all independent. Thus, we have

$$p(D|\Xi, R_i) = \prod_j p(d_{ij}|\Theta) \prod_{k,j,k \neq i} p(d_{kj}|\Phi) \quad (6)$$

where $p(d_{kj}|\Phi)$ and $p(d_{kj}|\Theta)$ is from (4).

Similarly, we have the following equation for the conditional probability density of D given that the user is in the noncontrol state (Ψ)

$$p(D|\Psi) = \prod_{ij} p(d_{ij}|O) \quad (7)$$

where $p(d_{kj}|O)$ is from (4).

In real computing, however, a direct calculation of the earlier equations would easily cause overflow, since it involves a number of multiplications of Gaussian distribution functions. To overcome this problem, we resort to using the following log measure for the probability computation

$$L = \log \left[\sum_{i=1}^{N_s} p(D|\Xi, R_i)P(\Xi, R_i) \right] - \log(p(D|\Psi)P(\Psi)). \quad (8)$$

This logarithm function is monotonic with the *a posteriori* probability (5). To avoid overflow, we calculate the first logarithm term on the right-hand side by

$$\log \left[\sum_{i=1}^{N_s} \exp \{ \log(p(D|\Xi, R_i)P(\Xi, R_i)) + M \} \right] - \log(\exp(M)) \quad (9)$$

where M is taken as $\frac{1}{N_s} \sum_{i=1}^{N_s} -p(D|\Xi, R_i)P(\Xi, R_i)$.

V. CONTROL STATE DETECTION FOR ASYNCHRONOUS CONTROL

As stated earlier, asynchronous BCI control requires effective differentiation between two user states: the control state where the user is intentionally using the interface and being focused on a particular button for a few rounds; the noncontrol state where the user is engaged in other things. To this end, we have devised an algorithm, which basically checks each possible time window in the ongoing EEG and computes if one window gives a high likelihood of control pattern. The algorithm only considers time windows no shorter than three rounds to ensure reliable detection.

In the following, we would like to give the basic procedure of the algorithm.

- 1) Initialize the system. Set the round count $k_r = 0$.
- 2) Receive a new round of EEG epochs and set $k_r = k_r + 1$; calculate the SVM scores d_{ik_r} of each epoch [see (1)].
- 3) Proceed if $k_r \geq L_m$ where L_m is the minimal window length taking for detection (usually $L_m \geq 3$); otherwise go back to step 2.
- 4) Enumerate each possible EEG window ending at the present round, with length $l \in \{L_m, \dots, k_r\}$.
 - a) Calculate the *a posteriori* probability $P_l = P(\Xi|D_l)$ [see (5)], where D_l represents the SVM scores for all the epochs in the window of length l .
- 5) Find the maximum of the posterior probability among all possible windows

$$l_m = \underset{l}{\operatorname{argmax}}\{P_l\}. \quad (10)$$

- 6) If $P_{l_m} > \eta$ where η is a preset value, a positive detection is decided, proceed; otherwise go back to step 2.
- 7) Extract the EEG window of length l_m (ending at the present round), and employ a classifier to estimate which command the subject wants to input (e.g., with the method in [22]).
- 8) Reset the round count $k_r = 0$; go back to step 2.

VI. EXPERIMENTS

A. Experiment for Assessing Receiver Operating Characteristics

It is important to design an appropriate experimental setup for studying asynchronous systems, in which one needs to take into consideration more factors than in traditional synchronous ones. As the literature has seen few studies on asynchronous P300-based BCIs, we would like to refer to prior arts of relevant studies on other BCIs using, e.g., motor imagery.

Pfurtscheller's group has endeavored into motor-imagery-based asynchronous BCIs. They reported an asynchronous virtual speller [39], which does not use machine-generated cues for motor imagery classification. However, since it does not consider the "noncontrol" state, that system is unfit for the asynchronous control task which is of concern in the present paper. The same group has actually addressed the "noncontrol"

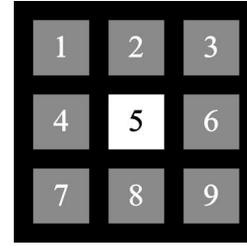


Fig. 4. Nine-button user interface for P300-based BCI. The interface intensifies each button alternately in a random order. Currently, the central button is intensified.

state in [10], where they introduced the ROC curves to evaluate the performance of an asynchronous BCI during imagined movements.

The ROC curves have been widely used in signal detection studies. For example, Birch's BCI group has been employing it for the study of their asynchronous BCI switches [25]. In signal detection, two essential performance indexes are of concern: true positive rate (TPR) and false positive rate (FPR). And the ROC curves depict the relation between the two rates of TRP and FPR. In a real application, one needs to find a particular detection threshold that best fits into the specific task, e.g., to achieve a satisfactory TPR while keeping FPR below a given limit. Therefore, in order to study the system performance in terms of ROC curves, we need to test every threshold and check the resultant TRP and FRP. Obviously, it is impossible by online processing (fixed threshold then). Thus, offline analysis is prevalent, using data from machine-guided data collection sessions [10], [25].

In the present paper, we devised a few machine-guided tasks so as to conduct offline analysis using ROC curves. These tasks were grouped into the following three sessions. Note that the nine-button user interface, as shown in Fig. 4, was employed in this study. Thus, one round of button flashing took 900 ms.

- 1) Session 1 consisted of three sections. In each of the first two sections, the subject attended to one button for eight rounds, paused 2 s (as per the computer's video guide), and moved on to the next button until he/she had gone through all the nine buttons. Hence, each of the two sections contained 72 rounds of EEG, corresponding to 72 epochs of target P300 and 576 epochs of nontarget P300 data. The two sections were both used to train the support vector machine to discriminate between target and nontarget P300 data. The setting of the third section was the same as that of the first two: the monitor was closed and the subject stayed in the noncontrol state. Hence, the data from the third section were used to estimate the distribution of SVM scores of noncontrol epochs.
- 2) In session 2, the subject stayed in the control state and concentrated on one button for 50 rounds, paused 2 s (as per the computer's video guide), and moved on to the next button until he had gone through all the nine buttons. The session was used to evaluate the proposed method in term of TPR.

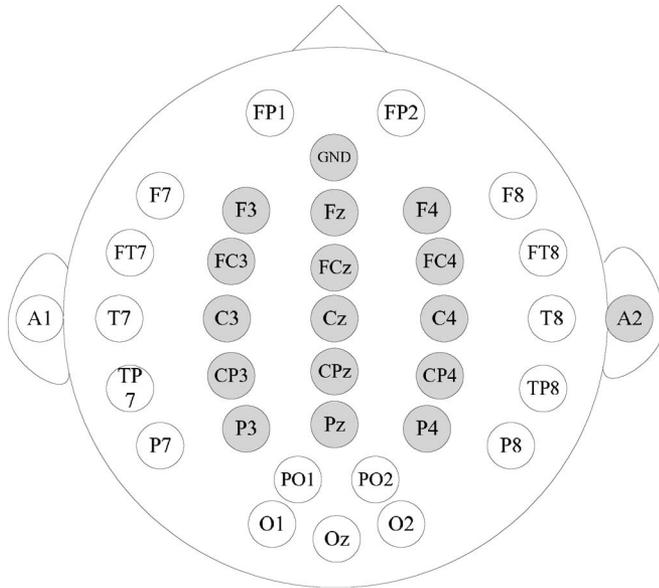


Fig. 5. Topography of EEG sites of the EEE cap, with 15 used channels (plus reference and ground points) in shade.

- 3) Session 3 consisted of three sections. In each of the three sections, the subject stayed in “noncontrol” state, paying no attention to any button nor the computer display. Each section was as long as session 2, thus also containing 50 rounds of EEG for each of the nine buttons. In the first section, the subject was singing a song. In the second section, the subject was relaxed and closed eyes. In the third section, the subject was given a question sheet including a few arithmetic tasks, and needed to finish the tasks quickly. In total, the three sections allowed us to evaluate the method in various noncontrol conditions. And in the data analysis, we combined the three sections together without discriminations.

According to the timing scheme mentioned before, Sessions 1–3 took approximately 4.5, 7, and 21 min, respectively. We allowed the subject to take a short break up to 2 min in between two sessions. So the total data collection on each subject ran for approximately 40 min, excluding the EEG preparation time.

In the experiment we used a NuAmps device from Neuroscan, Inc., to measure the scalp EEG signal. The EEG was recorded from Ag–AgCl electrodes placed at electrode sites in the inferior frontal, central, and parietal regions, including the following: “F3,” “Fz,” “F4,” “FC3,” “FCz,” “FC4,” “C3,” “Cz,” “C4,” “CP3,” “CPz,” “CP4,” “P3,” “Pz,” and “P4.” Fig. 7 shows the locations of the sites, as well as that of the ground and the reference points. The digitizer device worked at a sampling rate of 250 Hz. To ensure accurate recording of stimulus timing, we used a stimulus-generation and data-acquisition software reported in [40]. A *stimulus code* representing a particular button flashing was sent to the EEG hardware via the parallel port, and the EEG machine instead of the computer inserted that stimulus code into a special channel of the ongoing EEG. In this way, the precise time information was recorded, even though there was a delay between a stimulus being generated and the corresponding

EEG signal being received. Throughout the study, we used an interstimulus interval at 100 ms.

Four healthy subjects, all males, between 20 to 45 years, participated in the study. To help the subjects concentrate on the task, we asked them to count the flashing of the target button. Note that no subject screening was conducted, and we used all the four volunteer subjects throughout the study.

B. Experiment for Assessing Online Performance

In the aforementioned experiment design, it is interesting to investigate the online performance of the BCI. To this end, we have implemented the BCI in a real-time EEG processing system, using Visual C++ and C#. And we have designed an online test protocol described later.

In an experiment session, a subject sits comfortably in an armchair while continuously performing alternate control and noncontrol tasks.

- 1) *Two noncontrol tasks: reading and rest.* In the reading task, the subject is reading out some given stories. In the rest task, the subject is having a rest with eyes closed.
- 2) *One control task: inputting a given sequence of 32 digits.* As the nine-button interface is used, the digits from 1 to 8 were randomly selected to compose the sequence, while the digit “9” served as a “backspace” button to correct any input error during online test.

Each session starts with a reading task, followed by an input task, a rest task, and an input task, ending with a reading task. The duration of each task is determined by the subject in doing the online experiments. And there is no break between consecutive tasks. Actually, each of the subjects participating in this study spent 6–7 min in total on the reading tasks, and 5–9 min on the rest task. Because the exact duration of each task is needed to enable accurate computation of ITR and FPR, the actual start/end time points is recorded by the subjects pressing a mouse button. Note that the threshold for control-state detection is determined empirically for each subject using the training data (Section VI-A), for offline FPR to be lower than 1 per minute.

The BCI produces online feedback to the subject by: 1) highlighting detected digits quickly (0.5 s) in red color and 2) outputting the digits into a textbox on the computer monitor. Especially in the control task, the subject needs to use the digit “9” i.e., the “backspace” button, to correct any input error, so as to make the final output digit sequence identical to the given one. Therefore, after each session, the effective communication speed, or ITR, can be derived easily. For example, if a subject uses 5 min to input a correct sequence of 32 digits, the ITR would be $32 \log_2(9)/5 = 20.3$ b/min.

VII. RESULTS

A. Preprocessing and SVM

In the preprocessing procedure, we used temporal filtering to remove high-frequency noises and very slow waves. Thus, a fifth-order digital Butterworth filter with passband [0.5 Hz 15 Hz] was applied to the continuous EEG data.

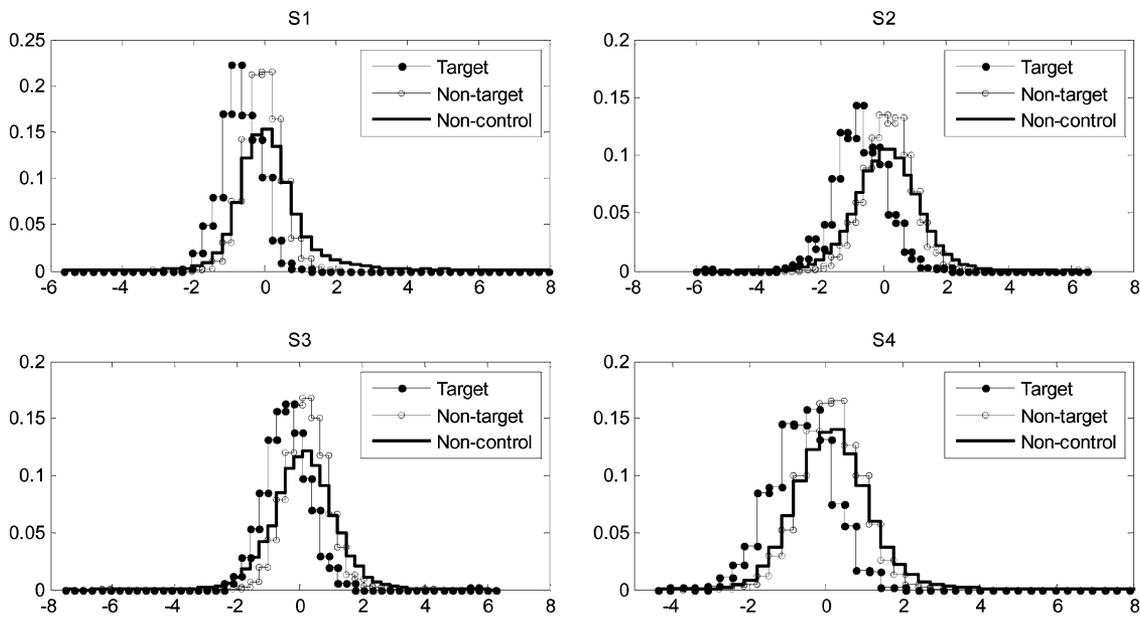


Fig. 6. Histograms of SVM margin scores (d) for target P300, nontarget P300, and noncontrol epochs, for each of the four subjects participating in this study.

Subsequently, the filtered EEG signals were downsampled by a factor of 4 in order to reduce the computational complexity. The downsampled signals were then segmented from 100 to 500 ms after the onset of a button flashing, and the results as well as their dynamic features [22] were concatenated to form a single feature vector that represents the epoch.

We computed the range of values for each variable in the feature vectors of the training samples. Thereafter, we normalized all the feature vectors in both training sets and test sets by mapping them to the range $[-1, 1]$. The normalized feature vectors serve as the input of the SVM, for which we employed the popular LibSVM toolbox with Gaussian kernels provided in [37].

B. Distributions of SVM Scores

We used “session 1” to train a binary SVM to distinguish between two types of signals: the target P300 epochs and nontarget P300 epochs. The SVM was then applied to “session 2” data set and “session 3” data set. Scores from target signals, nontarget signals as well noncontrol signals were collected and the histograms were plotted in Fig. 6.

To see whether the scores conform to Gaussian distributions, we ran a t -test on the three types of scores respectively for each subject. It turned out that the hypothesis was confirmed with 95% confidence level.

Next, we checked the difference among the distributions using the Kolmogorov–Smirnov test on each pair of samples. On all the subjects, the results indicate that we can reject the hypothesis that the distributions are the same, at the 95% confidence level, as all the p -values were all extremely small Table I. This confirms that nontarget signals and noncontrol signals shall be treated with different distribution models.

TABLE I
KOLMOGROVO-SMIRNOV TEST VALUES COMPARING PAIR-WISE DISTRIBUTIONS OF TARGET (TAG), NONTARGET (NTAG), AND NONCONTROL (NC) EPOCHS’ SVM SCORES

| Subject\Pair | TAG vs NTAG | NTAG vs NC | TAG vs NC |
|--------------|-------------|------------|-----------|
| S1 | 7.3e-100 | 8.2e-72 | 5.0e-104 |
| S2 | 1.6e-191 | 1.8e-17 | 1.9e-76 |
| S3 | 4.7e-62 | 8.3e-40 | 9.0e-43 |
| S4 | 2.7e-63 | 7.3e-11 | 3.6e-62 |

C. ROC Analysis

The performance assessment of an asynchronous control system involves two aspects: the capability for detecting the true events when the user is in the control state, and that for rejecting all the signals when the user is in the noncontrol state. For the first aspect, the performance measure termed TPR is often suggested that indicates how many control events the system is able to detect within a time unit, say, 1 min. For the second one, the measure termed FPR, also known as false positive rate, is suggested that indicates how many false events (in the noncontrol state) the system will detect on average within a time unit.

Since we use a threshold on the posterior probability measure (5) for the detection of the control state, both the TPR and the FPR are monotonical functions on the threshold value. Fig. 7 plots the receiver operation characteristic (ROC) curves for each of the four subjects. And it can be seen that when the threshold increases, both the TPR and the FPR drop. And the curves can be used to assess the system performance at various thresholds.

The purpose of the BCI is to determine when and which button (command) the subject tries to select (input). The asynchronous control mechanism above only addresses the first issue (when), i.e., the detection of the control event.

To determine which button the subject wants to select, we adopted a simple yet proven method for the classification [22] that picks up the maximal averaged SVM scores among the

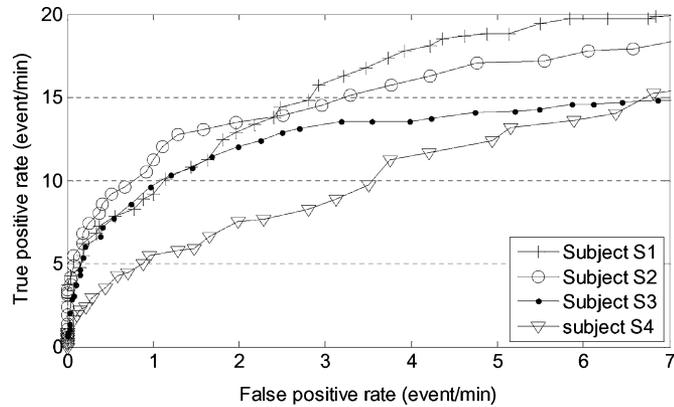


Fig. 7. TPR versus FPR. The TPR is the averaged rate of control events being detected by the asynchronous mechanism. The FPR is the averaged rate of false control events detected during a subject's noncontrol state.

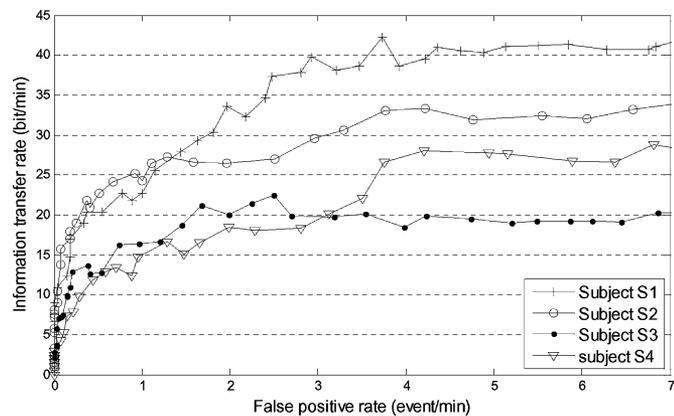


Fig. 8. ITR versus FPR. The ITR represents the capacity of the BCI as a communication channel.

buttons

$$c : \operatorname{argmax}_i \sum_{t \in T} d(i, t) \quad (11)$$

where T is the detected signal time.

To evaluate the detection plus classification system, we employed a widely used measure, named information transfer rate that indicates how many bits of information one is able to communicate effectively through the interface [41].

$$B = n_e \left\{ \log_2 N_s + P \log_2 P + (1 - P) \log_2 \left[\frac{(1 - P)}{(N_s - 1)} \right] \right\} \quad (12)$$

where n_e is the averaged number of event detected per minute and P is the probability that the target is hit (i.e., the control state is detected and the command is successfully recognized.) It can be seen that P is determined by both true positive rate R_{tp} and recognition accuracy R_r : $P = R_{tp} * R_r$.

Fig. 8 depicts the relation between ITR and FPR. Again, it shows that ITR is generally monotonic over FPR for each subject. The right limit of the FPR axis is seven events per minute. As the figure shows, the curves approaches asymptote beyond FPR=3–4 events/min; the right end of each curve actually approximates the ITR without rejection—same as in earlier

TABLE II
AVERAGED INFORMATION TRANSFER RATE VERSUS FALSE POSITIVE RATE

| FPR (event/min) | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------------|------|------|------|------|------|------|------|
| ITR (bit/min) | 19.8 | 24.7 | 27.0 | 29.5 | 29.8 | 29.8 | 31.3 |

TABLE III
ONLINE PERFORMANCE STATISTICS. FOR EACH OF THE FOUR SUBJECTS, AND AVERAGE

| | S1 | S2 | S3 | S4 | Average |
|-----------------|-------|-------|-------|-------|---------|
| ITR (bit/min) | 21.67 | 17.37 | 10.60 | 10.37 | 15.0 |
| FPR (event/min) | 0.20 | 0.71 | 0.39 | 1.54 | 0.71 |

synchronous systems. Hence, the figure indicates that, without considering false positives, one is able to achieve a subject-dependent ITR in the range from 20 to 40 b/min.

Due to finite number of samples, our information transfer rate and false positive rate both take discrete values. Nevertheless, we use (linear) interpolation method to estimate the ITR values at specific FPR values. This allows us to obtain the averaged ITR over FPR. See Table II for the results.

D. Online Experiment Results

The results are summarized in Table III. On average, the present BCI would be able to communicate 15 bit/min in the control state, while producing 0.71 false positive per minute. This can be compared with the offline results of ITR = 19.8 b/min and FPR = 1/min.

VIII. DISCUSSIONS

Similar to other signal detection systems, the criterion for detection-threshold determination is an important issue in the present BCI system. Actually, it is highly dependent on specific applications. For example, controlling a wheelchair demands a lower FPR than a speller due to safety concerns. Because the FPR and the TPR are monotonic functions on the detection threshold, one can easily set the threshold for a given FPR/TPR in the ROC curves (see Fig. 7) according to specific applications. Furthermore, in our implementation, a user can change this threshold online if necessary.

To greatly reduce FPR, one can adapt an additional “lock” mechanism in the asynchronous interface. In particular, this mechanism assigns a “locking/unlocking” function to a dedicated button. In order to lock/unlock the interface, a user needs to select the button once or repeatedly for a few times (n_l). Once the system detects a locking command, it will run in a “standby” mode by ignoring any positive detections on all buttons expect the “locking/unlocking” one. It will return to normal control mode once an unlocking command is detected. False positives may still occur, after a false unlocking happens. Nevertheless, the false unlocking rate can be extremely low. For an n -button interface with FPR = r_{fp} , the rate of false unlocking would be $(r_{fp} \cdot n)^{n_l}$. Take our nine buttons interface, for example, where a decision is made every 0.9 s. If we set FPR = 1/min and $n_l = 2$, the rate of false unlocking would be as low as 1 in 100 h. The asynchronous BCI method is not only applicable to computer applications like virtual speller, but also to the control of assistive

devices such as wheelchairs. We have successfully integrated the BCI control into an automated wheelchair [42]. With the asynchronous BCI control, a user does not need to issue commands to steer the wheelchair all the time. Instead, he just selects among a few predefined paths between different relevant locations. Once a path is selected, the user can rest while the wheelchair is moving using a dedicated path-following controller.

People with higher attention levels usually perform better in P300-based BCIs, since consistent P300 patterns can be extracted. Patients who suffer from neurological disorders may have short concentration span. It makes synchronous BCIs difficult for them to use. This problem can be addressed by our asynchronous mechanism. First, it allows the user to voluntarily switch between the control state and the noncontrol state at any time, making continuous attention unnecessary. Second, the attention level may vary from time to time, even within the same person. Extracting P300 patterns from a fixed number of rounds may not be favorable, as the user may not concentrate throughout all the rounds. On the other hand, the asynchronous mechanism extracts P300 patterns from EEG of varying time lengths. This is relatively more favorable, as the BCI adapts to the attention level, by finding the time window in which the attention is perceived to be the highest.

However, it is still an open problem to detect reliably the control state in patients with severe neurological disorders. This is because visual attention on the BCI may become very difficult. In addition, some patients suffer from shuddering or other involuntary body movements that corrupt the P300 considerably with noise. Research into these problems requires advances in tracking and classification of single-trial P300 as well as effective noise removal techniques.

IX. CONCLUSION

In summary, this paper demonstrated that it is quite possible to use the P300 as an effective asynchronous communication channel. Though it is difficult to reliably detect a single trial P300 because of considerable background noise and variations of the waveforms, we found our probabilistic models and detection algorithm could be used to effectively compute if a subject is concentrated on any button on the screen.

In particular, high-dimensional EEG data (multichannel time sequence) posed a problem to effective P300 modeling. We turned to the support vector margin space, transforming the original data into the univariate space while preserving the discriminative information of interest. Furthermore, we derived a method to compute the likelihood of the subject being in the control state, given the ongoing EEG.

Our experiments with four human subjects have shown that, on average, the BCI can effectively transmit information at a speed of 20–27 b/min (information transfer rate) with a low false positive rate ranging from 1 to 3/min. This indicates that, at a low false positive rate, the asynchronous BCI can achieve comparable information transfer rate to that by synchronous P300-based systems (e.g., 23.75 b/min in [15]). With the asynchronous mechanism, P300 BCIs will provide neurological patients with useful communication and control interfaces.

ACKNOWLEDGMENT

The authors are thankful to the three anonymous referees, as well as the associate editor K. R. Mueller, who coordinated the review, for their constructive comments that considerably improved the paper.

REFERENCES

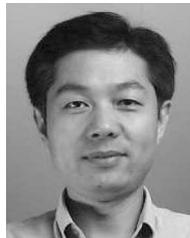
- [1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, pp. 767–791, 2002.
- [2] G. Dornhege, J. Millan, T. Hinterberger, D. McFarland, and K.-R. Müller, Eds., *Toward Brain Computer Interfacing*. Cambridge, MA: MIT Press, 2007.
- [3] E. A. Curran and M. J. Strokes, "Learning to control brain activity: A review of the production and control of EEG components for driving brain-computer interface (BCI) systems," *Brain Cognition*, vol. 51, pp. 326–336, 2003.
- [4] T. M. Vaughan, "Guest Editorial: Brain-computer interface technology: A review of the second international meeting," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 2, pp. 94–109, Jun. 2003.
- [5] T. Ebrabimi, J. M. Vesin, and G. Garcia, "Brain-computer interface in multimedia communication," *IEEE Signal Process. Mag.*, vol. 20, no. 1, pp. 14–24, Jan. 2003.
- [6] B. J. Fisch, *Fisch & Spehlmann's EEG Primer*. Amsterdam, The Netherlands: Elsevier, 1999.
- [7] J. R. Wolpaw, G. E. Loeb, B. Z. Allison, E. Donchin, O. F. do Nascimento, W. J. Heetderks, F. Nijboer, W. G. Shain, and J. N. Turner, "BCI meeting 2005—Workshop on signals and recording methods," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 138–141, Jun. 2006.
- [8] S. G. Mason and G. E. Birch, "A brain-controlled switch for asynchronous control applications," *IEEE Trans. Rehabil. Eng.*, vol. 47, no. 10, pp. 1297–1307, Oct. 2000.
- [9] J. del R. Millan and J. Mourino, "Asynchronous BCI and local neural classifiers: An overview of the adaptive brain interface project," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 2, pp. 159–161, Jun. 2003.
- [10] G. Townsend, B. Graimann, and G. Pfurtscheller, "Continuous EEG classification during motor imagery — Simulation of an asynchronous BCI," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 12, no. 2, pp. 258–265, Jun. 2004.
- [11] E. Donchin and D. B. Smith, "The contingent negative variation and the late positive wave of the average evoked potential," *Electroencephalography Clin. Neurophysiol.*, vol. 29, pp. 201–203, 1970.
- [12] L. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr. Clin. Neurophysiol.*, vol. 70, pp. 510–523, 1988.
- [13] B. Roder, F. Rosler, E. Hennighausen, and F. Nacker, "Event-related potentials during auditory and somatosensory discrimination in sighted and blind human subjects," *Cognitive Brain Res.*, vol. 4, no. 2, pp. 77–93, 1996.
- [14] S. Makeig, T.-P. Jung, A. J. Bell, D. Ghaherani, and T. J. Sejnowski, "Blind separation of auditory event-related brain responses into independent components," *Proc. Nat. Acad. Sci. USA*, vol. 94, pp. 10979–10984, 1997.
- [15] H. Serby, E. Yom-Tov, and G. F. Inbar, "An improved P300—Based brain-computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 1, pp. 89–98, Mar. 2005.
- [16] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao, and F. Yang, "BCI competition 2003—Data set IIb: Enhancing P300 wave detection using ICA-based subspace projections for BCI applications," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1067–1072, Jun. 2004.
- [17] T. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, and T. Sejnowski, "Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects," *Clin. Neurophysiol.*, vol. 111, no. 10, pp. 1745–1758, 2000.
- [18] S. Lemm, G. Curio, Y. Hlushchuk, and K. L. Mueller, "Enhancing the signal-to-noise ratio of ICA-based extracted ERPs," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 4, pp. 601–607, Apr. 2006.
- [19] A. Rakotomamonjy, V. Guigue, G. Mallet, and V. Alvarado, "Ensemble of SVMs for improving brain computer interface P300 speller performances," in *Proc. Int. Conf. Artif. Neural Netw.*, 2005, vol. 1, pp. 45–50.

- [20] M. Kaper, P. Meinicke, U. Grossekhoefer, T. Lingner, and H. Ritter, "Support vector machines for the P300 speller paradigm," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1073–1076, Jun. 2003.
- [21] P. Meinicke, M. Kaper, F. Hoppe, M. Heumann, and H. Rotter, "Improving transfer rates in brain computer interfacing: A case study," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1107–1114.
- [22] T. Manoj, C. Guan, and W. Jiankang, "Robust classification of EEG signal for brain-computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 1, pp. 24–29, Mar. 2006.
- [23] D. Krusienski, E. Sellers, F. Cabestaing, S. Bayouth, D. McFarland, T. Vaughan, and J. Wolpaw, "A comparison of classification techniques for the P300 speller," *J. Neural Eng.*, vol. 3, pp. 299–305, 2006.
- [24] J. F. Borisoff, S. G. Mason, and G. E. Birch, "Brain interface research for asynchronous control applications," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 160–164, Jun. 2006.
- [25] J. F. Borisoff, S. G. Mason, A. Bashashati, and G. E. Birch, "Brain-computer interface design for asynchronous control applications: Improvements to the LF-ASD asynchronous brain switch," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 985–992, Jun. 2004.
- [26] E. Halgren, K. Marinkovic, and P. Chauvel, "Generators of the late cognitive potentials in auditory and visual oddball tasks," *Electroencephalogr. Clin. Neurophysiol.*, vol. 106, pp. 156–164, 1998.
- [27] K. A. Moores, C. R. Clark, J. Hadfield, G. C. Brown, D. J. Taylor, S. P. Fitzgibbon, A. C. Lewis, D. L. Weber, and R. Greenblatt, "Investigating the generators of the scalp recorded visuo-verbal P300 using cortically constrained source localization," *Hum. Brain Mapp.*, vol. 18, pp. 53–57, 2003.
- [28] E. Halgren, P. Baudena, J. Clarke, C. Liegeois, P. Chauvel, and A. Musolino, "Intracerebral potentials to rare target and distractor auditory and visual stimuli. I. Superior temporal plane and parietal lobe," *Electroencephalogr. Clin. Neurophysiol.*, vol. 94, no. 3, pp. 191–220, 1995.
- [29] H. Yamasaki, K. S. Labar, and G. McCarthy, "Dissociable prefrontal brain systems for attention and emotion," *Proc. Natl. Acad. Sci.*, vol. 99, no. 17, pp. 11447–11451, 2002.
- [30] C. Mulert, L. Jager, R. Schmitt, P. Bussfeld, O. Pogarell, H. Moller, G. Juckel, and U. Hegerl, "Integration of fMRI and simultaneous EEG: Towards a comprehensive understanding of localization and time-course of brain activity in target detection," *NeuroImage*, vol. 22, pp. 83–94, 2004.
- [31] C. Guan, M. Thulasidas, and J. Wu, "High performance P300 speller for brain-computer interface," presented at the IEEE Int. Workshop Biomed. Circuits Syst. BioCAS, Singapore, Dec. 2004.
- [32] N. A. Busch, S. Debener, C. Kranczioch, A. K. Engel, and C. S. Herrmann, "Size matters: Effects of stimulus size, duration and eccentricity on the visual gamma-band response," *Clin. Neurophysiol.*, vol. 115, pp. 1810–1820, 2004.
- [33] B. H. Jansen, A. Allam, P. Kota, K. Lachance, A. Osho, and K. Sundaresan, "An exploratory study of factors affecting single trial P300 detection," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 975–978, Jun. 2004.
- [34] E. Donchin, K. M. Spencer, and R. Wijesinghe, "The mental prosthesis: Assessing the speed of a P300-based brain-computer interface," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 2, pp. 174–179, Jun. 2000.
- [35] E. Sellers, A. Kübler, and E. Donchin, "Brain-computer interface research at the University of South Florida Cognitive Psychophysiology Laboratory: The P300 speller," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 221–224, Jun. 2006.
- [36] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-14, no. 3, pp. 326–334, Jun. 1965.
- [37] C.-C. Chang and C.-J. Lin. (2001), LIBSVM: A library for support vector machines [Online]. Software Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [38] J. Platt, "Probabilities for SV machines," in *Advances in Large Margin Classifiers*, A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 2000, pp. 61–74.
- [39] R. Scherer, G. Muller, C. Neuper, B. Graimann, and G. Pfurtscheller, "An asynchronously controlled EEG-based virtual keyboard: improvement of the spelling rate," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 6, pp. 979–984, Jun. 2006.
- [40] C. Wang, C. Guan, and H. Zhang, "P300 brain-computer interface design for communication and control applications," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2005, pp. 5400–5403.
- [41] J. Wolpaw, N. Birbaumer, W. Heetderks, D. McFarland, P. Pckham, G. Schalk, E. Donchin, L. Quatrano, C. Robinson, and T. Vaughan, "Brain-computer interface technology: A review of the first international meeting," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 2, pp. 164–173, Jun. 2000.
- [42] B. Rebsamen *et al.*, "Controlling a wheelchair in a building using thought," *IEEE Intell. Syst.*, vol. 4, no. 2, pp. 18–24, Mar/Apr. 2007.



Haihong Zhang (M'07) received the Ph.D. degree in computer science from the National University of Singapore, Singapore, in 2005.

In 2005, he joined the Institute for Infocomm Research, Singapore, as a Research Fellow, where he is currently a Project Manager for neural signal processing. His current research interests include machine learning, pattern recognition, and brain signal processing for high-performance brain-computer interfaces.



Cuntai Guan (S'91–M'92–SM'03) received the Ph.D. degree in electrical and electronic engineering from Southeast University, China, in 1993.

From 1993 to 1994, he was at the Southeast University, where he was engaged on speech vocoder, speech recognition, and text-to-speech. During 1995, he was a Visiting Scientist at the Centre de Recherche en Informatique de Nancy (CRIN)/Centre National de la Recherche Scientifique (CNRS)-Institut National de Recherche en Informatique et en Automatique (INRIA), France, where he is engaged on key word spotting. From 1996 to 1997, he was with the City University of Hong Kong, Hong Kong, developing robust speech recognition under noisy environment. From 1997 to 1999, he was with the Kent Ridge Digital Laboratories, Singapore, where he was engaged on multilingual, large vocabulary, continuous speech recognition. He was a Research Manager and the R&D Director for five years in industries, focusing on the development of spoken dialogue technologies. In 2003, he established the Neural Signal Processing Group, Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore, where he is currently a Senior Scientist and a Program Manager. His current research interests include e-health, neural signal processing, brain-computer interface, machine learning, pattern classification, and statistical signal processing, with applications to assistive device, rehabilitation, and health monitoring.



Chuanchu Wang (M'07) received the B.Eng. and M.Eng. degrees in communication and electrical engineering from the University of Science and Technology of China, Hefei, China, in 1988 and 1991, respectively.

He is currently a Research Manager at the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. His current research interests include EEG-based brain-computer interface (BCI) and related applications.