Proceedings of the 29th Annual International
Conference of the IEEE EMBS
Cité Internationale, Lyon, France
August 23-26, 2007.

Towards Asynchronous Brain-computer Interfaces: A P300-based Approach with Statistical Models

Haihong Zhang, Chuanchu Wang, Cuntai Guan
Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
Email: {hhzhang, ccwang, ctguan}@i2r.a-star.edu.sg

***Abstract*— Asynchronous control is a critical issue in developing brain-computer interfaces for real-life applications, where the machine should be able to detect the occurrence of a mental command. In this paper we propose a computational approach for robust asynchronous control using the P300 signal, in a variant of oddball paradigm. First, we use Gaussian models in the support vector margin space to describe various types of EEG signals that are present in an asynchronous P300-based BCI. This allows us to derive a probability measure of control state given EEG observations. Second, we devise a recursive algorithm to detect and locate control states in ongoing EEG. Experimental results indicate that our system allows information transfer at approx. 20bit/min at low false alarm rate (1/min).**

I. Introduction

Brain-Computer Interface (BCI) is an emergent multidisciplinary technology which allows a brain to control a computer directly – without relying on normal neuromuscular pathways [1]. Its most important applications are mainly meant for the paralyzed people who are suffering from severe neuromuscular disorders, as BCIs can provide communication, control or rehabilitation tools for the restoration of lost abilities in the patients. Among brain signal acuiqistion measures, of particular interest is the electroencephalogram (EEG) [1], [2], [3]). Recording electrical brain signals from the scalp, it bears advantages like non-invasive, technically less demanding, and widely available at low cost [4], depite downsides like low signal-to-noise ratio, low spatial resolution.

This paper addresses an important issue of *asynchronous control* in EEG-based BCI, which means that the user can send a command through the interface at any time while the machine is able to capture the mental command in ongoing brain waves. On the contrary, conventional systems often assume that the user is always fully engaged in sending certain pre-defined mental commands. Clearly, the capability of asynchronous control is crucial for real applications.

Recent years have seen an increasing research interest for asynchronous control, primarily using motor imagery signals [5], [6], [7]. However, this paper reports the first attempt to build asynchronous control using the P300 signal. It is known that P300-based systems require minimal user training, and provide mental command classification with high accuracy [8].

The P300 is an evoke related potential (ERP) elicited in the brain in response to infrequent/oddball auditory, visual or somatosensory stimuli. Farwell and Donchin [9] first demonstrated the use of P300 for BCI in a so-called *oddball paradigm*. The paradigm can be extended for people with visual impairments, using auditory or tactile stimuli [10]. A great deal of work followed the oddball paradigm while seeking to improve the system performance from various signal processing viewpoints, e.g. [11], [12], [13].

In this paper we propose a computational approach for robust asynchronous control. First, we address how to compute the probability of control state for a given segment of EEG signals. To this end, user-specific Gaussian models using empirical data are employed, which allow us to describe various types of EEG signals in the P300-based BCIs. Based on the statistical models, we derive a probability measure of control/non-control states given observed EEG signals. Second, we devise an algorithm to detect and locate control states in ongoing EEG – not just a given period of EEG. This is important since it's unknown to the machine when and how long the user tries to input a single command through the interface. In essence, our algorithm consists of recursive steps enumerating possible time-windows of control at a time, and determining if a continuous control state occurs throughout one of the windows.

To evaluate the proposed approach, we conduct experiments on five human subjects. The experimental results attest to the efficacy of our approach. For example, on average, a user is able to input information effectively through our interface at approx. 20bit/min in control state, while producing only 1 false alarm per minute in non-control state.

II. Problem Description

A P300-based BCI framework is illustrated in Fig. 1. Unlike previous systems which intensify individual rows and columns, it flashes individual buttons successively in a random order. The timing of intensification is configurable and controlled by the machine, while the concurrent EEG signals are captured by an amplifier plus data aquisition device.

When a button flashes, the machine captures the concurrent readings in EEG. A period (typically 500ms long) of EEG signals following the stimulus is refered to as an *epoch*. A complete cycle in which every button flashes once and only once is referred to as a *round*.

We use N_s to denote the number of buttons on the command display. We use s_i^t to represent an epoch, associated with the intensification of the *i*-th button at the *t*-th round. Thus a complete round will consist of N_s epochs: $S = \{s_i^t\}$, where $i = 1, \ldots, N_s$.

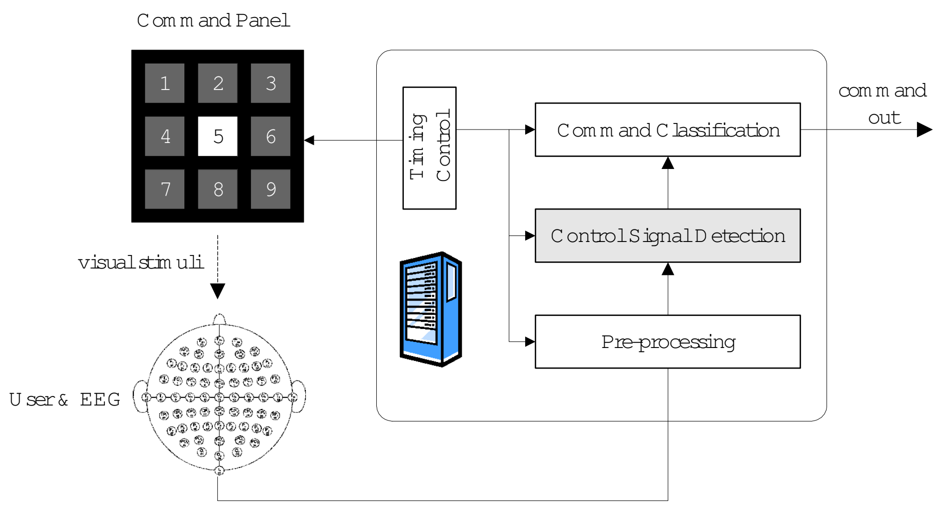

Fig. 1. System Illustration. This work introduces a special component, called "control signal detection", for enabling asynchronous control.

Obviously, control state detection in essence requires an automatical way to infer, from a S or a number of S, if the user is intended to a particular button. In view of low quality of single P300 epochs [14], it's often suggested to make a detection using a few rounds .

Let's now define a user's states and the epoch types. When a user is trying to input a command, he or she is in *control state* and shall pay consistent attention to a particular button. Otherwise, the user is in *non-control state*. In control state, epochs associated with intensifications of target button are *target epochs*, while the other epochs are *non-target epochs*. In non-control state all epochs are *garbage epochs*.

For asynchronous control, we derive our approach in the following steps: 1) Devise the probability model which describes the likelihood that a given period of EEG is a control signal; 2) Devise a method that enumerates all possible time windows at at time and check if the EEG in a time window is a control signal.

III. Models of control/non-control Signals

A. Target Signal and Non-target Signal Models

EEG signals often exhibit considerable variations even in the same type of epochs, which make the direct statistical modeling of the signals very difficult. Instead, in this work we resort to a special space created by support vector machines (SVMs).

SVMs are now a well-known classification method whose principle is to seek maximal margin between two classes. Here we use d to denote the distance from a pattern to the optimal hyperplane.

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

by using a kernel function k (here we uses Gaussian) which corresponds to the inner product in a Reproducing Kernel Hilbert Space. Here $\mathbf{x}_i$ is one of the N support vectors.

As suggested in [15], Gaussian functions may provide a good approximation to SVM scores for signals not present in the training set.

$$\begin{aligned} p(d|\Theta) &= \mathcal{N}(d-\mu_\theta, \sigma_1^2) & (2)\\ p(d|\Phi) &= \mathcal{N}(d-\mu_\phi, \sigma_2^2) & (3)\\ p(d|O) &= \mathcal{N}(d-\mu_o, \sigma_3^2) & (4)\end{aligned}$$

where the parameters can be simply learned from empirical samples using the MAP method.

B. Probability Models for $P(C|\mathbf{S})$

Consider multi-round signal as shown in Fig. **??**. In control state, the user will produce a single row of Θ epochs with the rest being Φ.

Let $P(\Xi, R_i)$ be the probability of control state and the user being attending to the R_i button. By Bayesian rule, it is straightforward to have

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

Assume the signals $S_i^{(j)}$ are independently generated. It follows that

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

where d_{ij} denotes the SVM score of $S_i^{(j)}$ in Fig. (**??**).

$$p(D|\Psi) = \prod_{ij} P(d_{ij}|\Theta') \tag{7}$$

In real implementations, however, direct calculation of the above equations would easily result in overflow because of a number of multiplications of exponentials. To overcome this problem, we turn to use the following itemxxx for the probability measure

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

which, as can be easily seen, is monotorical withxxx the a posteriori probability Eq. 5. To avoid overflow, the first logrithm term on the right side is usually calculated by

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

where M is usualy taken as the average of $-p(D|\Xi,R_i)P(\Xi,R_i)$.

IV. Control Signal Detection

As stated earlier, the decision making aims to differentiate between two cases: the user is intentionally using the interface and being focused on a particular button for n rounds; the user is in idle state or engaged in other things irrelevant to the control task.

When doing the decision, two factors must be taken into consideration. 1. False acceptance rate (F_{AR}). The rate that non-control signals are classified into Case 1 and associated with the user' commands. 2. False rejection rate (F_{RR}). The rate that true control signals are classified into Case 2 and the system does not convey the user's intentions.

Now the asynchronous control procedure can be described as follows.

1) Initialization. Set minimal length L_m. Clear buffer. Set round count $k_r = 0$;
2) Receive a new round of ERP epochs. $k_r = 0$.
3) Proceed if $k_r >= L_m$;otherwise go back to Step 2;
4) Enumerate all possible signal segments $\{S\}$ ending at the last epoch, with length greater than or equal to L_m:
5) Calculate the a posterior probability $P(C|S)$ for each $\{S\}$, and pick up the signal segment S_opt with maximal probability measure P_opt;
6) If $P_opt > \eta$ where η is a preset value, proceed; otherwise go back to Step 2;
7) Carry out classification on S_opt (e.g. with the method in [13]), output the results;
8) Go to Step 1;

V. Experiments

We used a NuAmps device from Neurosoft, Inc. to capture scalp EEG signals. The signals were sampled at 250Hz on 15 selected electrode sites around the central, namely, 'F3', 'Fz', 'F4', 'FC3', 'FCz', 'FC4', 'C3', 'Cz', 'C4', 'CP3', 'CPz', 'CP4', 'P3', 'Pz', 'P4'.

Five healthy subjects participated in the P300 study using the nine-button user-interface as shown in Figure x. In particular, each subject went through three sessions as below

- Session 1 consists of 2 sets of 8 rounds of running when the subject was in control state. And the data was used to train target/non-target ERP classifiers (SVMs).
- Session 2 consists of 1 set of 50 rounds of running when the subject was in control state. The data was used to evaluate the false rejection rate (FRR).
- Session 3 consists of 3 sets of 50 rounds of running when the subject was in non-control state. The data was used to evaluate the false alarm (false acceptance) rate (FAR). In the three sets, the subject was doing three tasks respectively, namely, singing, resting with eye closed, and calculating.

Please see [16] for the button-intensification timing.

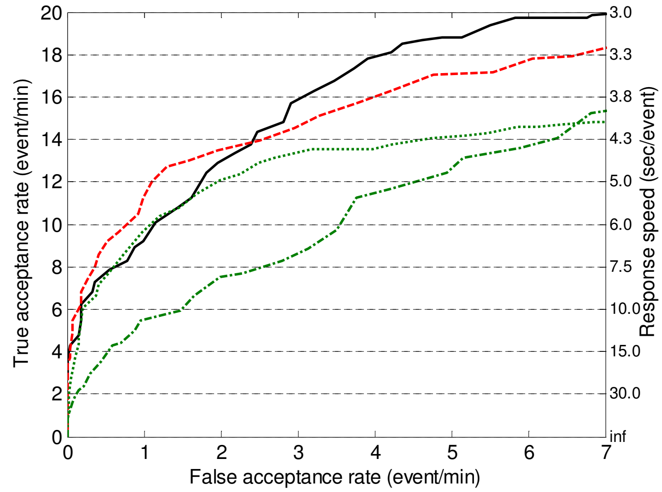

Fig. 2. True acceptance rate versus false acceptance rate

VI. Data analysis

A. Pre-processing

In the pre-processing procedure, we used temporal filtering to remove high frequency noises and very slow wave. Thus, a 5th-order digital Butterworth filter with passband [0.5Hz 15Hz] was applied to the continuous EEG data.

Subsequently, the filter EEG signals were downsampled by a factor of 4 in order to reduce the sample number (thus reducing computational complexity in folowing steps). The downsampled signals were then segmented from 100ms to 500ms after the start of the button intensification in each epoch, and the results were concatenated to form a single vector that represents the epoch.

For the SVM, we used the popular LibSVM toolbox provided by [17] and chose the default setting with Gaussian kernel. At the input of the SVMs, all the vectors were normalized according to the empirical limits learned from the training set.

B. Performance: true acceptance rate and false acceptance rate

We only detected signal length no shorter than 3 rounds, as too few rounds often yield unsatisfactory results. Since we use a threshold in the detection of Control State, the TAR (on control signals) and the (FAR) (on non-control signals) of the system are monotonical functions on the threshold. Thus, we use the popular ROC to measure the performance of detection.

Figure 2 plots the TAR vs FAR curves for the subjects. The vertical axis on the right side plots the response time corresponding to the true acceptance rate. The response time

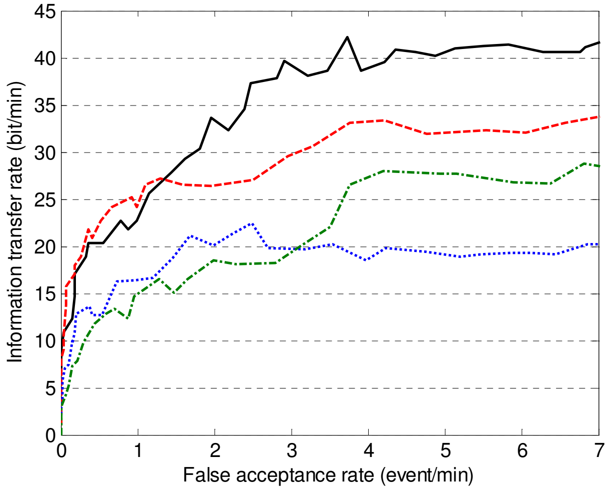

Fig. 3. Information transfer rate

is how long it is expected for a subject to be concentrated on a single choise until the control effort is accepted. On average, for example, the system is able to accept around 8 inputs (among 9 possible buttons), or one needs to be concentrated for 7.5 seconds until an control action is accepted, within one minute at the risk of accepting 1 false signal during the 1 minute's non-control state time.

C. Performance: Information Transfer Rate

The final goal of the ERP-based BCI system is to determine when and which button (command) the subject tries to press (input). The asynchronous control machanism above only addresses the first issue, i.e. the detection of control signals. To complete the system, one needs to plug in a classifier in order to classify the detected signals.

We adopted a simple yet proven method for the classification [13] which picks up the maximal averaged SVM scores among the buttons.

To evalutate the detection plus classification system, we employed a widely-used measure, named information transfer rate which indicates how many bits of information one is able to communicate effectively through the interface.

$$B = n_r \left\{ \log_2 N_s + P \log_2 P + (1-P) \log_2 [(1-P)/(N_s - 1)] \right\} \tag{10}$$

where n_r is the number of rounds per minute, and P is the probability that the target is hit. It can be seen that P is determined by true acceptance rate R_{ta} and recognition accuracy R_r: $P = R_{ta} * R_r$.

The following figure shows the results on each subject.

On average, one is able to communicate 20 bits/min at FAR=1/min, or 30 bits/min at FAR=4/min.

VII. Conclusions

In this paper we proposed a computational approach for asynchronous control using the P300 signal. First, we used Gaussian models in the support vector margin space to describe various types of EEG signals. And we derived a probability measure of control state given EEG observations. Furthermore, we devised a recursive algorithm to detect and locate control states in ongoing EEG. Experimental results indicate that our system allows satisfactory information transfer at low false alarm rate.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113:767–791, 2002.

[2] 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 and Cognition*, 51:326–336, 2003.

[3] T. Ebrabimi, J. M. Vesin, and G. Garcia. Brain-computer interface in multimedia communication. *IEEE Signal Processing Magazine*, 20(1):14–24, 2003.

[4] Brain-computer interface technology: A review of the second international meeting. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):94–109, 2003.

[5] S. G. Mason and G. E. Birch. A brain-controlled switch for asynchronous control applications. *IEEE Transactions on Rehabilitation Engineering*, 47:1297–1307, 2000.

[6] Jose del R. Millan and Josep Mourino. Asynchronous bci and local neural classifiers: An overview of the adaptive brain interface project. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):159–161, 2003.

[7] G. Townsend, B. Graimann, and G. Pfurtscheller. Continuous eeg classification during motor imageryłsimulation of an asynchronous bci. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(2):258–265, 2004.

[8] H. Li, Y. Li, and C. Guan. An effective bci speller based on semi-supervised learning. In *Int. Conf. IEEE Engineering in Medicine and Biology Society*, 2006.

[9] L. A. Farwell and E. Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potential. *Electroencephalography and Clinical Neurophysiology*, 70:510–523, 1988.

[10] 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 Research*, 4(2):77–93, 1996.

[11] H. Serby, E. Yom-Tov, and G. F. Inbar. An improved P300-based brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(1):89–98, 2005.

[12] E. Donchin, K. M. Spencer, and R. Wijesinghe. The mental prosthesis: assessing the speed of a P300-based brain-computer interface. *IEEE Transactions on Rehabilitation Engineering*, 8:174–179, 2000.

[13] T Manoj, C. Guan, and W. Jiankang. Robust classification of eeg signal for brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(1):24–29, 2006.

[14] 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 Transactions on Biomedical Engineering*, 51(6):975–978, 2004.

[15] J.C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[16] C. Wang, C. Guan, and H. Zhang. P300 brain-computer interface design for communication and control applications. In *Int. Conf. IEEE Engineering in Medicine and Biology Society*, 2005.

[17] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.