

# Adaptive training session for a P300 speller brain-computer interface

Bertrand Rivet<sup>a,\*</sup>, Hubert Cecotti<sup>a</sup>, Margaux Perrin<sup>b,c,d</sup>, Emmanuel Maby<sup>b,c,d</sup>, Jérémie Mattout<sup>b,c,d</sup>

<sup>a</sup>GIPSA-lab, CNRS UMR5216, Grenoble University, Grenoble, France

<sup>b</sup>INSERM, U821, Lyon, 69500, France

<sup>c</sup>Institut Fédératif des Neurosciences, Lyon, 69000, France

<sup>d</sup>Université Lyon 1, Lyon, 69000, France

---

## Abstract

With a Brain-Computer Interface (BCI), it is nowadays possible to achieve a direct pathway between the brain and computers thanks to the analysis of some particular brain activities. The detection of even-related potentials, like the P300 in the oddball paradigm exploited in P300-speller, provides a way to create BCIs by assigning several detected ERP to a command. Due to the noise present in the electroencephalographic signal, the detection of an ERP and its different components requires efficient signal processing and machine learning techniques. As a consequence, a calibration session is needed for training the models, which can be a drawback if its duration is too long. Although the model depends on the subject, the goal is to provide a reliable model for the P300 detection over time. In this study, we propose a new method to evaluate the optimal number of symbols (*i.e.* the number of ERP that shall be detected given a determined target probability) that should be spelt during the calibration process. The goal is to provide a usable system with a minimum calibration duration and such that it can automatically switch between the training and online sessions. The method allows to adaptively adjust the number of training symbols to each subject. The evaluation has been tested on data recorded on 20 healthy subjects. This procedure lets drastically reduced the calibration session: height symbols during the training session reach an initialized system with an average accuracy of 80% after five epochs.

**Keywords:** brain-computer interface, P300 speller, adaptive signal processing, angles between subspace, xDAWN spatial filters, Bayesian Linear Discriminant Analysis

---

## 1. Introduction

Brain-Computer Interfaces (BCI), which aim at providing a direct communication between the user's brain and a computer, have recently become an existing pluridisciplinary field of research, e.g., (Wolpaw et al., 2002). Indeed the BCI field requires research effort in many aspects: signal processing (e.g., signal enhancement), machine learning (e.g., classification), user interface (e.g., ergonomics) and of course neuroscience for understanding the different neural processings. In fact, most of the recent progress in BCI during the last decade has been achieved by coupling knowledge from different research fields.

While BCI is a pluridisciplinary field of research, the evolution of each field does not progress at the same pace over time. The brain responses that are being detected have been known for decades in neuroscience, e.g., (Vidal, 1973). Machine learning has grown a lot in the last two decades. To some extends, some signal processing and classification techniques remain relatively simple but they are efficient, due to the noisy brain signals aspect and Occam's razor principle. It is

the case of Linear Discriminant Analysis (Fisher, 1936), which stays very popular in the field. Moreover, in spite of the recent improvements of BCI, they shall face the hard reality of their early and still pending promises, e.g., pragmatic problems for potential daily user. This paper focuses on some practical aspects for non-invasive BCI.

The BCI performances are both related to the performance of the signal processing techniques that are used for assigning some EEG signal to a command and to the possibility of the user to adapt herself/himself to the system over time. Indeed, advanced machine learning, signal processing and classifier techniques have been widely used for improving BCIs (Bashashati et al., 2007; Lotte et al., 2007; Müller et al., 2008). In spite of these recent improvements in the BCI community, several obstacles remain to fully transfer laboratory demonstrator BCIs to commercial applications. Whereas tuning the different parameters of a BCI in relation to a specific individual can improve the performance of the system, this procedure is time consuming. For training a classifier, a large database containing labeled EEG signal is often needed. To obtain these data, a training session is required where the subject has to follow a calibration procedure which can be long and discouraging.

Several types of potential BCI users can be distinguished. First, BCIs are usually dedicated to people with severe motor disabilities who are unable to communicate through any other means. For these people, a BCI is the only way to communi-

---

\*Corresponding author: Gipsa-lab  
961 rue de la Houille Blanche  
BP 46  
F - 38402 GRENOBLE Cedex  
Phone number: +33(0) 476 826 486  
Fax number: +33(0) 476 574 790  
bertrand.rivet@gipsa-lab.grenoble-inp.fr

cate and the main challenge is to have a functional BCI. While reducing the time of the training session can be a certain advantage, it is not the main objective. For persons who suffer from severe disabilities, like the locked-in syndrome, having a working BCI can still be challenging. Besides, one may add that for some particular disabilities, it would be more advantageous to consider an invasive solution. Second, BCIs can be used by people with disabilities or elderly people who need a device to facilitate their daily life. In this situation, current challenges in BCI are to get a reliable BCI, which could be efficient daily in real condition. For this group of potential users, the time dedicated to the training session and the expected performance shall be well balanced. Third, BCIs can be used by healthy people as a complement to other device in multi tasks situations. In this type of situation, a BCI should ideally be flawless: the calibration step shall be as short as possible. Indeed, once the excitement of controlling something with the mind has passed, the usability stays an important satisfaction criterion. Some potential BCI users, even disabled, are highly demanding in term of performance and usability. The last decade has shown that BCIs can effectively work, they should now ideally become plug'n'play (Cecotti, 2010). The capacity to transfer technologies to commercial applications will be critical for the future of BCI.

On a more practical aspect for BCI users, the calibration step can be a drawback. With a lot of available data, it is possible to train classifiers and to improve the reliability of the detection procedure. However, one may wonder when the signal processing steps show their limits and it becomes useless to pursue the training procedure. Therefore, a challenge is to determine the ideal training session duration for the personalization of a BCI. On one hand, a too short training session would not be enough and would involve the creation of a poorly trained classifier. On the other hand a too long training session would frustrate the user, decrease her/his motivation. The calibration step shall not be considered as the final state of the system adaptation but only one first step for providing an efficient system: it allows to switch automatically between the training session and the online spelling use. For instance, classifiers can be trained in an unsupervised way over time in such fashion that the system adaptability stays invisible to the user. In addition, the user can improve his performance by finding appropriate ways to adapt his/her behavior to the system or to some feedbacks (Martinez et al., 2007; McFarland et al., 2006; Moore, 2003).

In this paper, we focus on the problem of the required time that is needed to train a classifier and provide efficient results for the P300-Speller. In other words, how much time a subject has to spend during an experiment to efficiently determine the features of his/hers. In the P300-Speller, the duration of the training session can be evaluated as a number of characters to spell. The duration for spelling a character is indeed determined by initial setting of the speller like the inter-stimuli interval. This research is part of an ongoing effort in the French BCI community to reduce the time dedicated to the calibration step of the P300-Speller.

The paper is organized as follows: each step of the P300 detection, from the spatial filtering methods till the classification,



Figure 1: Classical P300 interface with the third row intensified.

is described in the second section. The experimental protocol is then detailed in the third section. Finally, the results of the offline classification are presented and discussed in the last section.

## 2. P300 speller and material

### 2.1. P300-Speller brain computer interface

The BCI problem addressed in this paper concerns the P300-speller, which enables people to write a text on a computer (Farwell and Donchin, 1988). It is based on the oddball paradigm: the task is to discriminate between epochs containing a P300 potential evoked by a (rare) target stimulus and epochs associated with the (frequent) non-target stimuli. In the classical P300 BCI, a  $6 \times 6$  matrix containing all the available symbols is displayed on a screen (Fig. 1). To select a symbol, the user has to focus her/his attention on the symbol she/he wants to spell. The stimuli are the intensification of the rows and columns in a random order. Hence, the intensification of the target symbol (i.e. corresponding row and column) is a rare and unexpected event which elicits a P300 potential (i.e. a positive voltage deflection with a latency of about 300ms). Stimulation is organized in blocks of 12 flashes such that each row/column is intensified once per block in a random order. Blocks of 12 intensifications are repeated  $N_{epoch}$  times for each symbol: therefore  $2 \times N_{epoch}$  epochs amount the  $12 \times N_{epoch}$  possible epochs should contain a P300 potential. To identify these epochs allows to determine the target symbol as the intersection of the detected row and column.

### 2.2. Data and pre-processing

Data were recorded from 20 healthy subjects (age =  $26 \pm 5.7$  years, 13 males and 7 females) (Maby et al., 2010). Subjects were wearing an EEG cap with 32 active electrodes (actiCap, Brain Products GmbH, Munich). The electrodes for reference and the ground were placed on the nose and on the forehead, respectively. Data from five sessions have been recorded:

- training session : 50 symbols with 10 epochs (i.e.  $N_{epoch} = 10$ ),
- test session 1: 60 symbols with 3 epochs (i.e.  $N_{epoch} = 3$ ),
- test session 2: 60 symbols with 5 epochs (i.e.  $N_{epoch} = 5$ ),

- test session 3: 60 symbols with 8 epochs (i.e.  $N_{epoch} = 8$ ),
- test session 4: 60 symbols with 10 epochs (i.e.  $N_{epoch} = 10$ ).

For training session, the inter-stimuli interval (ISI) is equal to 170ms. For each test session, four different values of ISI were tested (110ms, 150ms, 190ms, and 230ms) with 15 symbols for each value. It is worth noting that different ISIs may impact the classification accuracy, however several values of ISIs have been used to test the robustness of the proposed algorithm. For all these sessions, the stimulus onsets are known as well as the corresponding intensified row or column. The users have to copy words, as a consequence the target stimuli onsets are known for all the training and test sessions but this information is only used with the training session.

EEG signals were initially sampled at 100Hz. Data pre-processing included a fourth order Butterworth bandpass filter between 1Hz and 20Hz.

### 3. Methods: adaptive training session

#### 3.1. Spatial filters

Recorded EEG signals not only contain the P300 potentials but also ongoing cerebral activity and eventually muscular/ocular artifacts. A usual step to enhance a particular brain pattern is to use spatial filters: e.g., independent component analysis (Comon and Jutten, 2010; Xu et al., 2004), common spatial pattern (Koles, 1991; Blankertz et al., 2008) or more recently xDAWN algorithm (Rivet et al., 2008, 2009). This latter method (Appendix A) is chosen since it has been shown to provide accurate spatial filters. Let  $X^{(i)} \in \mathbb{R}^{N_t \times N_s}$  denote the recorded signals related to the  $i$ -th symbol, where  $N_t$  and  $N_s$  are the number of samples and sensors, respectively. xDAWN algorithm provides spatial filters  $\hat{U}^{(i)} \in \mathbb{R}^{N_s \times N_f}$ , with  $N_f$  the number of spatial filters and  $\mathcal{I}$  the set of training symbol used to train these spatial filters. The enhanced signals  $S^{(i)} \in \mathbb{R}^{N_e \times N_f}$  of the  $i$ -th symbol are thus given by  $S^{(i)} = X^{(i)}\hat{U}^{(i)}$ .

#### 3.2. Classification

The enhanced signals are then segmented in order to identify which row/column elicits a P300. Each segment that represents a row/column includes all samples between 0 and 1s after the beginning of the intensification for all sensors:  $S_{(k,j)}^{(i)} \in \mathbb{R}^{N_e \times N_s}$  is then the enhanced signal of the  $j$ -th repetition of the  $i$ -th symbol related to the row/column indexed  $k$ .  $N_e$  is the number of samples corresponding to 1s (in this study,  $N_e = 100$ ). Finally, the corresponding feature vector  $\mathbf{p}_{(k,j)}^{(i)} \in \mathbb{R}^{(N_e N_s/4)}$  is obtained from the concatenation of all time-course samples in the enhanced signals  $S_{(k,j)}^{(i)}$  which have been decimated by a factor four (i.e. resampled at 25Hz). Among the proposed classifiers that are considered for BCI, Bayesian linear discriminant analysis (BLDA) is chosen since it has been proved to be efficient (Hoffmann et al., 2008). The discriminant vector  $\mathbf{d}^{(i)}$  is estimated from the set of pairs  $\{\mathbf{p}_{(k,j)}^{(i)}, t_{(k,j)}^{(i)}\}_{i \in \mathcal{I}, 1 \leq j \leq N_{epoch}, 1 \leq k \leq 12}$  obtained from the set  $\mathcal{I}$  of symbols in the training database, where

$t_{(k,j)}^{(i)} = 0$  or 1 is the class index associated with the corresponding feature vector, i.e. 1 (resp. 0) if the related intensification is a target (resp. not a target).

#### 3.3. Adaptive training session

The proposed adaptive algorithm aims at estimating the optimal amount of data needed in the training session to learn both the spatial filters  $U$  and the classifier  $\mathbf{d}$ . Starting from one symbol in the training set  $\mathcal{I}$ , the proposed method increases the number of training symbols until the convergence is reached as detailed bellow. As already pointed out, one can use a cross validation technique to estimate the classification accuracy by splitting the data base into two sets. The first one is used to train the spatial filters and the classifier. The second set is used to test the performance. This splitting is then repeated several times as in the K-fold method. However this procedure needs a lot of data to be accurate. The proposed method avoids this drawback by predicting the accuracy without the need to compute it.

Let us suppose that at the  $i$ -th iteration,  $i$  symbols compose the training set, i.e.  $\mathcal{I}_i = \{1, \dots, i\}$ . The training database is thus composed of the set of  $\{X^{(j)}\}_{j \in \mathcal{I}_i}$  for which the target and non-target onsets are known. This set of training data allows to estimate both

- the spatial filters  $\hat{U}^{(\mathcal{I}_i)}$  thanks to the xDAWN algorithm (Section 3.1),
- the discriminative vector  $\mathbf{d}^{(\mathcal{I}_i)}$  thanks to the BLDA (Section 3.2).

The problem is now to know if adding a new symbol in the training session is needed to improve the classification accuracy of the test session. Since BLDA is a linear classifier, then the performance accuracy directly depends on the values of the components of the discriminative vector. As a consequence, if adding a new data in the training set does not change (too much) these values then one can conclude that the convergence is reached. The similitude between the two discriminative vectors  $\mathbf{d}^{(\mathcal{I}_i)}$  and  $\mathbf{d}^{(\mathcal{I}_{i-1})}$  is based on the principal angle between the two subspaces spanned by these vectors. Indeed, reshape vectors  $\mathbf{d}^{(\cdot)} \in \mathbb{R}^{N_t N_s/4}$ , which mixes time and space features, into a matrix  $D^{(\cdot)} \in \mathbb{R}^{N_t/4 \times N_s}$ , which separates the time features along the rows and space features along the columns, defines a subspace  $\mathcal{S}^{(\cdot)}$  of dimension  $N_s$  in  $\mathbb{R}^{N_t/4}$  (as  $N_s \leq N_t/4$ ). The closeness between  $\mathbf{d}^{(\mathcal{I}_i)}$  and  $\mathbf{d}^{(\mathcal{I}_{i-1})}$  is thus defined as the principal angle between  $\mathcal{S}^{(\mathcal{I}_i)}$  and  $\mathcal{S}^{(\mathcal{I}_{i-1})}$  defined by

$$\cos \theta^{(\mathcal{I}_i)} = \max_{\mathbf{u} \in \mathcal{S}^{(\mathcal{I}_i)}} \max_{\mathbf{v} \in \mathcal{S}^{(\mathcal{I}_{i-1})}} \mathbf{u}^T \mathbf{v} \quad (1)$$

as recalled in Appendix B. Because of the variance in the estimation of the discriminative vector  $\mathbf{d}^{(\cdot)}$  due to the few number of data in the training session, one can smooth the evolution of  $\theta^{(\mathcal{I}_i)}$  over the iterations  $i$  thanks to

$$\theta_s^{(i)} = (I + \delta D^T D)^{-1} \theta^{(i)}, \quad (2)$$

where  $\theta^{(i)} = [\theta^{(I_1)}, \dots, \theta^{(I_i)}]^T$ ,  $I \in \mathbb{R}^{i \times i}$  is the identity matrix,  $D \in \mathbb{R}^{(i-1) \times i}$  is the Toeplitz bidiagonal matrix whose first row is  $[-1, 1, 0, \dots, 0, 0]$

$$D = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & -1 & 1 \end{bmatrix}$$

and  $\delta > 0$  is the smoothing parameter. This latter parameter allows to obtain a trade-off between the fidelity to the original data  $\theta^{(i)}$  and the smoothness of the regularized angles  $\theta_s^{(i)}$ . Indeed,  $\theta_s^{(i)}$  is the solution of

$$\theta_s^{(i)} = \arg \min_{\theta} \|\theta^{(i)} - \theta\|_2^2 + \delta \|D\theta\|_2^2,$$

where  $\|\theta^{(i)} - \theta\|_2^2$  represents the fidelity to the data and  $\|D\theta\|_2^2$  the smoothness of the estimated solution, respectively. The convergence is reached when the last value  $\theta_s^{(I_i)}$  of  $\theta_s^{(i)}$  is larger than a threshold  $\eta$ . Finally, the full algorithm to adaptively select the optimal size of the training set is summarized in Algorithm 1.

---

**Algorithm 1** Adaptive training session.

---

{Initialization}

Choose the threshold  $\eta$  and smoothing parameter  $\delta$

$\mathcal{I}_0 = \emptyset$ ,  $i = 0$ ,  $\epsilon = \text{TRUE}$  and  $\mathbf{d}^{(0)} = 0$

{Learning loop}

**while**  $\epsilon$  **do**

Increment the counter:  $i \leftarrow i + 1$

Add a symbol in the training set:  $\mathcal{I}_i = \mathcal{I}_{i-1} \cup i$

Estimate spatial filters  $\hat{U}^{(I_i)}$  from all  $\{X^{(j)}\}_{j \in \mathcal{I}_i}$  (Section 3.1)

Estimate classifier  $\mathbf{d}^{(I_i)}$  (Section 3.2)

Compute angle  $\theta^{(I_i)}$  between  $\mathcal{S}^{(I_i)}$  and  $\mathcal{S}^{(I_{i-1})}$  (1)

Smooth angle evolution  $\theta_s^{(I_i)}$  (2)

**if**  $\theta_s^{(I_i)} > \eta$  **then**

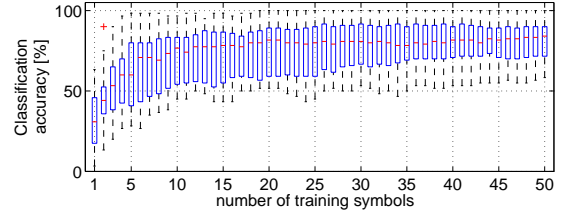
Stop training:  $\epsilon = \text{FALSE}$

**end if**

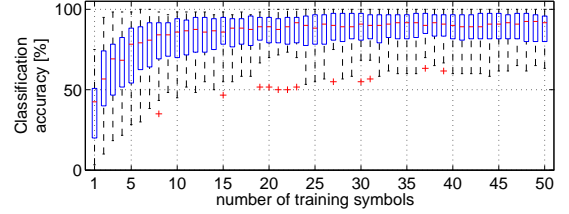
**end while**

---

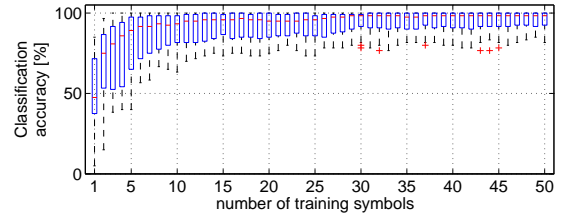
The choice of the two parameters  $\eta$  and  $\delta$  should be related to the following remarks. Choosing a threshold  $\eta$  such that  $\cos \eta$  is close to one ensures that the convergence is reached but increases the number of data in the training session. On the contrary a smaller value of  $\cos \eta$  will speed up the training session but the classification accuracy could be reduced. Secondly, a too large value of the smoothing parameter  $\delta$  only allows small variations of the smooth angles  $\theta_s^{(I_i)}$  and as a consequence requires lot of training data to reach the threshold  $\eta$ . On the contrary, a too small value of  $\delta$  does not smooth enough the variations of  $\theta^{(I_i)}$ , the convergence criterion could then be verified only due to the noise variance, which means that the training dataset could be not large enough to ensure good classification accuracy.



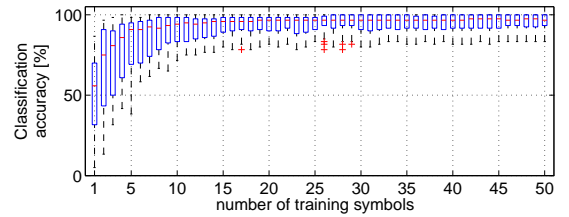
(a) Test session 1:  $N_{epoch} = 3$ .



(b) Test session 2:  $N_{epoch} = 5$ .



(c) Test session 3:  $N_{epoch} = 8$ .



(d) Test session 4:  $N_{epoch} = 10$ .

Figure 2: Classification accuracy in relation to the number of symbols used in the training session.

## 4. Results

In this section, the classification accuracy in relation to the number of symbols in the training data set is presented in Section 4.1. Investigations of the influence of the threshold  $\eta$  and of smoothing parameter  $\delta$  are then presented in Section 4.2 and Section 4.3, respectively.

In this section, the box plots represent distribution of the displayed quantity for all subjects. The central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points the algorithm considers to be not outliers, and the outliers are plotted individually.

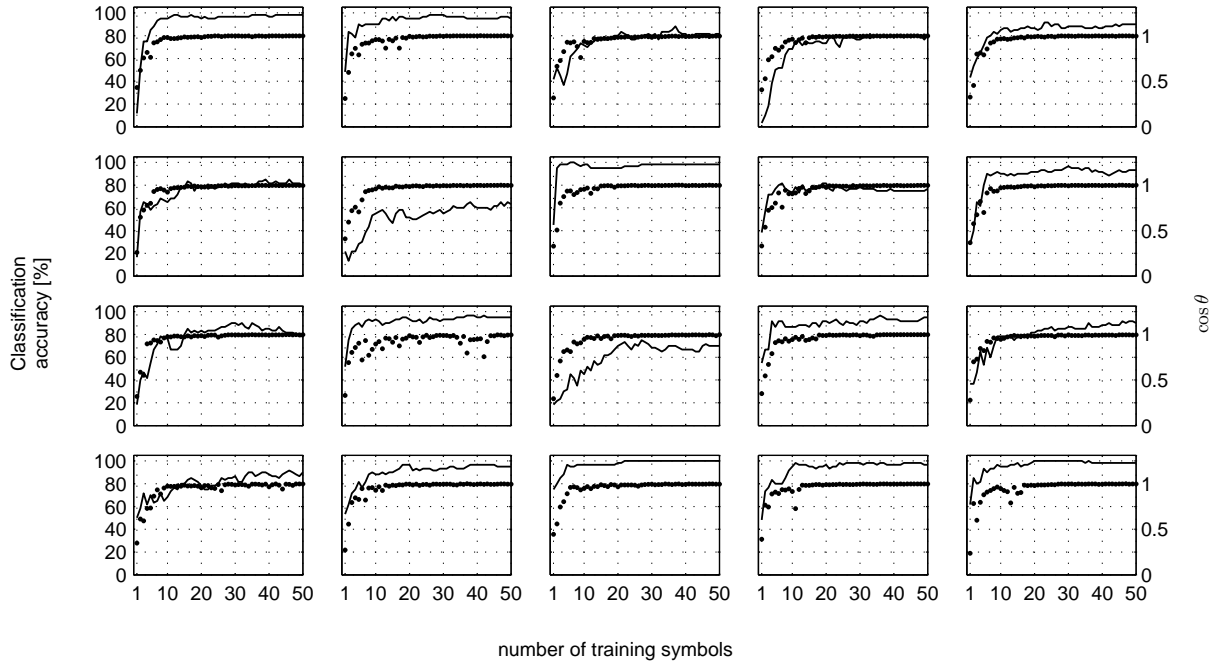


Figure 3: Relation between the classification accuracy achieved with the test session 2 ( $N_{epoch} = 5$ ) and the angle between discriminative vectors  $\mathbf{d}^{(I_i)}$  and  $\mathbf{d}^{(I_{i-1})}$ . For each of the 20 subjects (from bottom left to top right), the continuous line is the classification accuracy (left labels) and the dots are the cosinus of the angle  $\theta^{(I_i)}$  (right labels).

#### 4.1. Number of training symbols

The classification accuracy versus the number of symbols in the training data set is presented on Figure 2. For a fixed number of training symbols, the chronological order during the training session is chosen, i.e., when  $i$  symbols are selected, they correspond to the first  $i$  symbols in the training session. The spatial filters and the classifier are then tested on the four testing session with 3, 5, 8 or 10 epochs, respectively. Obviously, the classification accuracy increases with the number of symbols in the training set. But one can see that increasing too much the number of training symbol do not increase drastically the classification accuracy. Indeed, with the test session 4 if the objective is to achieve at least 80% (resp. 90%) of classification accuracy, a t-test shows that there is no significative difference with more than 9 (resp. 16) training symbols. Moreover, this is confirmed on the test session 2 and an average objective of 80%, a t-test shows that using more than 15 training symbols does not increase significantly the classification accuracy. This figure highlights the possibility to reduce the number of training symbols without a significant breakdown of the classification accuracy. Clearly, the classification accuracy increases with the number of epochs (from Figure 2(a) to Figure 2(d)).

For the 20 subjects, the classification accuracy and the angle  $\theta^{(I_i)}$  between  $\mathbf{d}^{(I_i)}$  and  $\mathbf{d}^{(I_{i-1})}$  in relation to the number of training symbols is presented on Figure 3. This figure shows the high level of correlation between the classification accuracy and the evolution of the angle  $\theta^{(I_i)}$ . For most of the subjects,

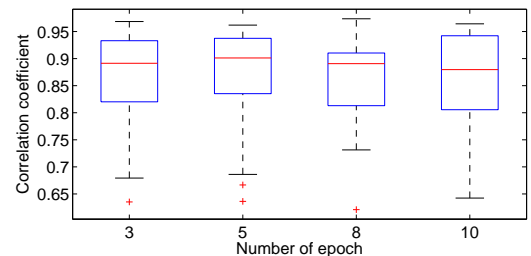


Figure 4: Distribution of the correlation coefficients between the angles  $\theta^{(I_i)}$  and the classification accuracy achieved with the four test session ( $N_{epoch} = 3, 5, 8$  or  $10$ ).

the two curves are very correlated: as a consequence the angles  $\theta^{(I_i)}$  can be used to predict the number of training symbols needed to reach the convergence of the classification accuracy without using a cross validation approach. This is clearly shown on Figure 4 which displays the distributions of the correlation coefficients between the angles  $\theta^{(I_i)}$  and the classification accuracy achieved with the four test sessions. Indeed, the median correlation coefficient is comprised between 0.88 and 0.90. For all the correlation coefficients, the p-value is less than  $1.6e-6$  with a 99.9% confidence interval.

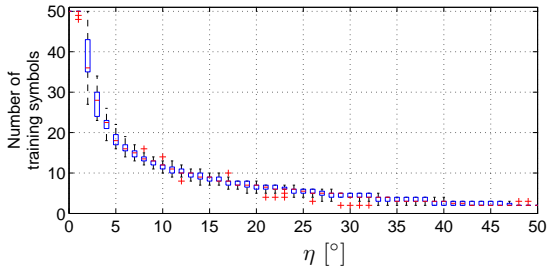


Figure 5: Number of selected symbols in the training session vs the threshold  $\eta$  expressed in degree. Smoothing parameter  $\delta$  is set to 2.

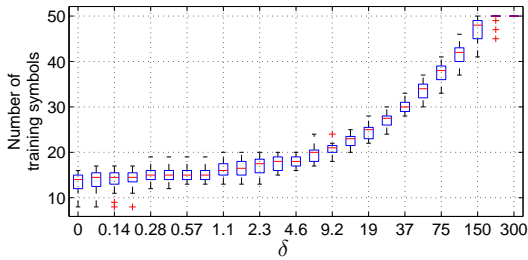


Figure 6: Number of selected symbols in the training session vs the smoothing parameter  $\delta$ . Threshold angle  $\eta$  is set to  $5^\circ$ .

#### 4.2. Influence of threshold $\eta$

Figure 5 presents the number of selected symbols in the training session by the proposed algorithm (Algo. 1) in relation with the threshold  $\eta$ . The number of selected symbols decreases as the threshold  $\eta$  increases. Moreover, the dispersion of the number of selected symbols is quite small as long as the threshold  $\eta$  is larger than  $3^\circ$ . There is thus a direct relationship between the threshold  $\eta$  and the number of the selected symbols. As a consequence, one can estimate the number of selected symbols by choosing the threshold  $\eta$ . Finally, this curve presents a knee between  $\eta = 5^\circ$  and  $\eta = 10^\circ$ , which appears to be a good trade-off between the duration of the training session and the provided classification accuracy in the test sessions.

#### 4.3. Influence of smoothing parameter $\delta$

Figure 6 presents the number of selected symbols in the training session by the proposed method (Algo. 1) in relation with the smoothing parameter  $\delta$ . The number of selected symbols increases with the smoothing parameter. This growth is relatively limited in average as long as the smoothing parameter  $\delta$  is less than 10. However, even a limited smoothing of the angles  $\theta^{(T)}$  ( $\delta$  less than .3) increases the smallest number of selected symbols from 8 to 12. This has a direct impact of the classification accuracy by improving it in average from 67% to 75%.

#### 4.4. Adaptive training

Figure 7 compares the classification accuracy achieved by three different methods of training. ‘Ref’ is the reference method

Test session	Increase of classification accuracy
1 ( $N_{epoch} = 3$ )	$2 < 3 < 4 < 6 < 9 < 10 < 19 < 21 < 50$
2 ( $N_{epoch} = 5$ )	$2 < 3 < 5 < 7 < 9 < 15 < 17 < 26$
3 ( $N_{epoch} = 8$ )	$2 < 3 < 4 < 5 < 8 < 11 < 13 < 36 < 49$
4 ( $N_{epoch} = 10$ )	$2 < 4 < 6 < 7 < 9 < 15 < 25 < 43$

Table 1: Number of necessary symbols in the training database to increase the classification accuracy (CA) of the test session. Column ‘‘Increase of classification accuracy’’ indicates the number of symbols in the training database that are required to increase the CA based on a left-tailed t-test ( $\alpha = 0.01$ ):  $n_1 < n_2$  means that  $n_2$  is the smallest number such that the first  $n_2$  symbols of the training database are necessary to increase the CA compared to the CA achieved with the first  $n_1$  symbols of the training database.

which exploits all the 50 symbols in the training set. ‘Adaptive’ is achieved by the proposed method ( $\eta = 20^\circ$  and  $\delta = 10$ ). For the ‘Fix’ method, the same number of training symbols is used for all subjects which is chosen to be equal to the median of the number of training symbols provided by the ‘Adaptive’ method. With the chosen parameters  $\eta$  and  $\delta$ , this fixed number is equal to 8. In this experiment,  $\eta$  is chosen large to have a short training session and as a consequence,  $\delta$  is also chosen large to avoid a too short training session. It is interesting to note that for almost all the subjects the classification accuracy is similar with the ‘Fix’ (78%) or ‘Adaptive’ methods (80%), while the ‘reference’ achieves 88%. Indeed, the number of selected symbols in the training session is quite equal to 8 for all the subjects. However, the proposed adaptive procedure allows to adjust the training session to the specificity of the subject. For instance, for subject 3, the maximal accuracy is achieved with a few number of training symbols (Fig. 3), the adaptive procedure has reduced the training session to 5 symbols. Moreover, for subject 8 who needs more symbols to reach the convergence (Fig. 3), the adaptive method stops after 10 training symbols. The same remark can be expressed for subject 12. It is worth noting that the subjects, for whom a large number of symbols is selected in the training phase, are generally subjects with the lowest classification accuracy, for instance subjects 8 and 12 (Fig. 3).

Finally, one can show on Table 1 that to add a symbol in the training database until ten symbols increases significantly the CA based on a left-tailed t-test ( $\alpha = 0.01$  and the p-values remain smaller than 0.01). Then, one have to add definitively more symbols in the training database to continue to increase the CA. This last remark shows that the proposed method is well adapted to select automatically the minimum number of necessary symbols in the training database.

## 5. Conclusions

The problem of the optimal training session is still challenging to provide user friendly BCIs. However, while almost all the current studies only used a fixed number of symbols to train the BCI elements, the current paper investigates an adaptive method to adjust the number of training data. Indeed, the classification accuracy highly depends on the signal processing methods and on the classification techniques to enhance and

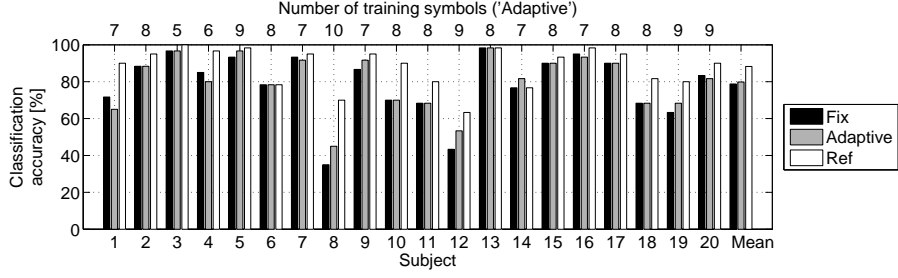


Figure 7: Comparison of classification accuracies achieved by different training procedures. For the ‘adaptive’ method,  $\eta = 20^\circ$  and  $\delta = 10$ . For ‘Fix’ and ‘Ref’ methods, 8 and 50 training symbols are used, respectively. The test session is the second one  $N_{epoch} = 5$ .

to detect the P300 potentials. These steps are of greatest importance to ensure the success of a BCI since a poor accuracy can discourage people to use it. The method proposed in this study to switch automatically between the training session and the online session seems to be very promising. As it is shown in the present work, the angle between two consecutive classifier vectors is suitable to check the convergence of the training session. The proposed method allows to have a good initialization of a P300 speller so that people can start to use the BCI without frustration. Furthermore, the adaptive training session would not only benefit BCI users but also other studies using the P300 detection. Many physiological and psychological studies rely on measurement of the P300 to examine event related potentials, like in combination to behavioral tasks. As the P300 detection has a broad range of uses in clinical research, an adaptive training session would offer a quick evaluation of the P300 characteristics for a particular individual.

Future works will deal with the use of this method in a real context, *i.e.* a real automatic switch between the training session and the online one. Moreover, the next challenge concerns the improvement of this method and its extension to provide an online training so that the BCI can continuously adjust itself to the environment or to the user’s physiology (e.g., tiredness or concentration).

## Appendix A. xDAWN spatial filters

In this appendix, the principle of the xDAWN algorithm is briefly reprised (Rivet et al., 2008, 2009). Its goal is to maximize the signal to signal-plus-noise ratio (SSNR). This method relies on two assumptions: i) signal is made of two typical pattern, one evoked by the targets and one evoked by all stimuli (target and non-target ones); ii) responses evoked by the target stimuli could be enhanced by spatial filtering. Let  $X \in \mathbb{R}^{N_t \times N_s}$  denote the recorded signals, where  $N_t$  and  $N_s$  are the number of samples and sensors, respectively. We denote the pattern synchronized with the target and non-target stimuli by  $A_1 \in \mathbb{R}^{N_t \times N_s}$  and  $A_2 \in \mathbb{R}^{N_t \times N_s}$ . Thus,

$$X = D_1 A_1 + D_2 A_2 + N, \quad (\text{A.1})$$

where  $D_1 \in \mathbb{R}^{N_t \times N_1}$  and  $D_2 \in \mathbb{R}^{N_t \times N_2}$  are Toeplitz matrices whose first column entries are set to zero except for those that

correspond to target stimuli and all stimuli, respectively.  $N_1$  and  $N_2$  indicate the number of samples in responses  $A_1$  and  $A_2$ .  $N \in \mathbb{R}^{N_t \times N_s}$  denotes the residual noise.

The aim of the xDAWN algorithm is to estimate  $N_f$  spatial filters  $U_1 \in \mathbb{R}^{N_s \times N_f}$  so that the SSNR defined by

$$g(U) = \frac{\text{Tr}(U^T \hat{\Sigma}_1 U)}{\text{Tr}(U^T \hat{\Sigma}_X U)} \quad (\text{A.2})$$

is maximum

$$\hat{U}_1 = \arg \max_{U_1} g(U_1) \quad (\text{A.3})$$

where  $\text{Tr}(\cdot)$  is the trace operator and  $\hat{\Sigma}_1 = \hat{A}_1^T D_1^T D_1 \hat{A}_1$ ,  $\hat{\Sigma}_X = X^T X$ . Note that  $\hat{A}_1$  is the least mean square estimation of the unknown target evoked response  $A_1$ . Since  $D_1 A_1$  and  $D_2 A_2$  could overlap,  $\hat{A}_1$  is estimated from

$$\begin{pmatrix} \hat{A}_1 \\ \hat{A}_2 \end{pmatrix} = (D^T D)^{-1} D^T X, \quad (\text{A.4})$$

with  $D = [D_1, D_2]$ ,  $\hat{A}_1$  is thus estimated by  $\hat{A}_1 = B_1^T X$ . To estimate the spatial filters  $\hat{U}_1$ , one computes the generalized eigenvalue decomposition (Golub and Van Loan, 1996) of pair  $(\hat{\Sigma}_1, \hat{\Sigma}_X)$  such that

$$\hat{\Sigma}_1 U = \hat{\Sigma}_X U \Lambda, \quad (\text{A.5})$$

where  $\Lambda$  is a diagonal matrix with entries equal to the generalized eigenvalues  $\lambda_1 \geq \dots \geq \lambda_{N_s}$  and  $U$  is a full matrix whose columns are the corresponding eigenvectors. Finally, the enhanced signals are given by

$$\hat{S}_1 \triangleq X \hat{U}_1 = D_1 A_1 \hat{U}_1 + D_2 A_2 \hat{U}_1 + N \hat{U}_1, \quad (\text{A.6})$$

where  $\hat{U}_1$  is the matrix whose  $N_f$  columns are the eigenvectors associated with the first  $N_f$  largest eigenvalues  $\lambda_i$ .

## Appendix B. Angle between subspaces

This appendix summarizes some useful considerations about angles between subspaces (Golub and Van Loan, 1996). Let  $\mathcal{S}_1$  and  $\mathcal{S}_2$  be two subspaces in  $\mathbb{R}^m$ . The principal angles  $\theta_k$  between subspaces  $\mathcal{S}_1$  and  $\mathcal{S}_2$  are defined by

$$\cos(\theta_k) = \max_{\mathbf{u} \in \mathcal{S}_1} \max_{\mathbf{v} \in \mathcal{S}_2} \mathbf{u}^T \mathbf{v} = \mathbf{u}_k^T \mathbf{v}_k, \quad (\text{B.1})$$

subject to  $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$ ,  $\mathbf{u}^T \mathbf{u}_i = 0$ , and  $\mathbf{v}^T \mathbf{v}_i = 0$ ,  $\forall i = \{1, \dots, k-1\}$ . Let  $Q_1 \in \mathbf{R}^{m \times p}$  and  $Q_2 \in \mathbf{R}^{m \times q}$  be two orthonormal basis of  $\mathcal{S}_1$  and  $\mathcal{S}_2$ , respectively. Assume that  $p \geq q$ . The angles between subspaces can be efficiently obtained by the singular value decomposition of  $Q_1^T Q_2$ :

$$Y^T(Q_1^T Q_2)Z = \text{diag}(\lambda_1, \dots, \lambda_q).$$

The principal angles and vectors are then obtained by  $\cos(\theta_k) = \lambda_k$ ,  $k = \{1, \dots, q\}$  and  $[\mathbf{u}_1, \dots, \mathbf{u}_p] = Q_1 Y$ ,  $[\mathbf{v}_1, \dots, \mathbf{v}_q] = Q_2 Z$ , respectively.

Notice that  $\theta_k$  is such that  $0 \leq \cos \theta_k \leq 1$ ,  $\cos \theta_k = 0$  means that the two related subspaces are orthogonal and  $\cos \theta_k = 1$  means that there is an intersection of the two subspaces.

## Acknowledgements

This work has been supported by the French National Research Agency (ANR) through the DEFIS program (project Co-Adapt ANR-09-EMER-002).

Bashashati, A., Fatourech, M., Ward, R. K., Birch, G. E., 2007. A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. *Journal of Neural Engineering* 4, 32–57.

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., Müller, K.-R., 2008. Optimizing spatial filters for robust eeg single-trial analysis. *IEEE Signal Processing Magazine* 25 (1), 41–56.

Cecotti, H., 2010. A self-paced and calibration-less SSVEP based brain-computer interface speller. *IEEE Trans. on Neural Systems and Rehab. Eng.* 18, 127–133.

Comon, P., Jutten, C. (Eds.), 2010. *Handbook of Blind Source Separation Independent Component Analysis and Applications*. Academic Press.

Farwell, L. A., Donchin, E., December 1988. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology* 70 (6), 510–523.

Fisher, R., 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7, 179–188.

Golub, G. H., Van Loan, C. F., 1996. *Matrix Computation*, 3rd Edition. Johns Hopkins University Press.

Hoffmann, U., Vesin, J.-M., Ebrahimi, T., Diserens, K., Jan. 2008. An efficient P300-based brain-computer interface for disabled subjects. *Journal of Neuroscience Methods* 167 (1), 115–125.

Koles, Z., December 1991. The quantitative extraction and topographic mapping of the abnormal components in the clinical eeg. *Electroencephalography and Clinical Neurophysiology* 79 (6), 440–447.

Lotte, F., Congedo, M., Lecuyer, A., Lamarche, F., Arnaldi, B., 2007. A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering* 4, R1–R13.

Maby, E., Gibert, G., Aguera, P.-E., Perrin, M., Bertrand, O., Mattout, J., 2010. The OpenViBE P300-speller scenario: a thorough online evaluation. In: *Human Brain Mapping Conference*.

Martinez, P., Bakardjian, H., Cichocki, A., 2007. Fully online multicommand brain-computer interface with visual neurofeedback using SSVEP paradigm. *Computational Intelligence and Neuroscience*.

McFarland, D. J., Krusienski, D. J., Wolpaw, J. R., 2006. Brain-computer interface signal processing at the wadsworth center: mu and sensorimotor beta rhythms. *Progress in Brain Research*, chapt. 26 159, 411–419.

Moore, M. M., 2003. Real-world applications for brain-computer interface technology. *IEEE Trans Neural Systems and Rehabilitation Engineering* 11 (2), 162–165.

Müller, K.-R., Tangermann, M., Dornhege, G., Krauledat, M., Curio, G., Blankertz, B., 2008. Machine learning for real-time single-trial EEG-analysis: From brain-computer interfacing to mental state monitoring. *J Neurosci Methods* 167 (1), 82–90.

Rivet, B., Souloumiac, A., Attina, V., Gibert, G., August 2009. xDAWN algorithm to enhance evoked potentials: Application to brain-computer interface. *IEEE Transactions on Biomedical Engineering* 56 (8), 2035–2043.

Rivet, B., Souloumiac, A., Gibert, G., Attina, V., August 2008. "P300 speller" Brain-Computer Interface: Enhancement of P300 evoked potential by spatial filters. In: *Proc. European Signal Processing Conference (EUSIPCO)*. Lausanne, Switzerland.

Vidal, J. J., June 1973. Toward direct brain-computer communication. *Annual reviewing of Biophysics and bioengineering* 2, 157–180.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., Vaughan, T. M., June 2002. Brain-computer interfaces for communication and control. *Clinical Neurophysiology* 113 (6), 767–791.

Xu, N., Gao, X., Hong, B., Miao, X., Gao, S., Yang, F., June 2004. BCI Competition 2003–Data Set IIb: Enhancing P300 Wave Detection Using ICA-Based Subspace Projections for BCI Applications. *IEEE Transactions on Biomedical Engineering* 51 (6), 1067–1072.