Documenting, modelling and exploiting P300 amplitude changes due to variable target delays in Donchin's speller

Luca Citi *, **Riccardo Poli, and Caterina Cinel**

Brain-Computer Interfaces Lab, School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK

E-mail: lciti@neurostat.mit.edu, rpoli@essex.ac.uk, ccinel@essex.ac.uk

Abstract. The P300 is an endogenous Event-Related Potential (ERP) which is naturally elicited by rare and significant external stimuli. P300s are used increasingly frequently in Brain Computer Interfaces (BCI) because users of ERP-based BCIs need no special training. However, P300 waves are hard to detect and, therefore, multiple target stimulus presentations are needed before an interface can make a reliable decision. While significant improvements have been made in the detection of P300s, no particular attention has been paid to the variability in shape and timing of P300 waves in BCIs. In this paper we start filling this gap by documenting, modelling and exploiting a modulation in the amplitude of P300s related to the number of non-targets preceding a target in a Donchin speller. The basic idea in our approach is to use an appropriately weighted average of the responses produced by a classifier during multiple stimulus presentations, instead of the traditional plain average. This makes it possible to weigh more heavily events that are likely to be more informative, thereby increasing the accuracy of classification. The optimal weights are determined through a mathematical model that precisely estimates the accuracy of our speller as well as the expected performance improvement w.r.t. the traditional approach. Tests with two independent datasets show that our approach provides a marked statistically significant improvement in accuracy over the top performing algorithm presented in the literature to date. The method and the theoretical models we propose are general and can easily be used in other P300-based BCIs with minimal changes.

Submitted to: *J. Neural Eng.*

* Luca Citi is now with the department of Anesthesia, Massachusetts General Hospital / Harvard Medical School, Boston, MA 02114 USA, and with the Brain and Cognitive Sciences department, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

1. Introduction

Brain-Computer Interfaces (BCIs) measure specific (intentionally and unintentionally induced) brain activity signals and translate them into device control signals [1, 2].

Many factors limit the performance of BCIs. These include: the natural variability and noise in the brain signals measured; the limitations of recording devices, of the processing methods that extract signal features and of the algorithms that translate these features into device commands; the quality of the feedback provided to the user; the lack of motivation, the tiredness, the limited degree of understanding, the age variations, the handedness, etc. of users; the combination of mental tasks used; and the natural limitations of the human perceptual system [3].

While amplitude and shape variations in brain waves are often a hindrance for BCIs, in some cases they carry information which can be exploited to improve performance, irrespective of whether a physiological explanation for such variations is available [3, 4]. Some of these variations are hard to document because standard averages may severely misrepresent what really goes on in the brain on a trial by trial basis [5–8], although more sensitive averaging techniques have recently become available [9, 10].

In this paper, we will document, theoretically model and exploit a source of waveform variations in P300 waves, namely, their modulation caused by variations in the interval between target-stimulus presentations. We illustrate our ideas using the Donchin speller, although the benefits of the approach can be accrued also in other P300-based systems.

The paper is organised as follows. In section 2 we provide some background on P300 waves and on the factors that may affect their characteristics, with particular attention paid to effects associated with the timing of stimulus presentation. In the same section we also introduce the Donchin speller paradigm. In section 3, we document the effects of the number of non-targets preceding a target on the amplitude and shape of P300s. In section 4, we consider a typical two-stage implementation of Donchin's speller, where one stage processes the responses to single stimuli while the other integrates such responses and decides which character to enter. We then derive mathematical models of the two stages, which, together, express the speller's accuracy, and generalise such models to make them applicable to a variety of ERP-based BCIs. Section 5 extends the model of the first stage of Donchin speller so as to consider the modulations of P300s associated with stimulus history documented in section 3. In section 6, we propose and model a new algorithm for the second stage of the speller that exploits such effect making it possible to compare its performance to that of the standard speller. The analysis (corroborated by Monte Carlo simulations) indicates that our algorithm is superior to the top performing Donchin speller from the literature. In section 7 we verify the predictions of our mathematical models by using two independent datasets: one from BCI Competition III and 12 subjects tested in our labs. Results show that the new system is statistically significantly superior to the original. We discuss and position our results in section 8. Finally, we present some conclusions and indicate promising avenues for future work in section 9.

2. Background

2.1. P300 waves and factors affecting their characteristics

Among the different approaches used in BCI studies, those based on event related potentials (ERPs) [11–13] are rather appealing since, at least in principle, they offer a relatively high bit-rate and require no user training. ERPs are relatively well defined shape-wise variations of the ongoing EEG which are elicited by a stimulus and are temporally linked to it. ERPs include an exogenous response, due to the primary processing of the stimulus, as well as an endogenous response, which is a reflection of higher cognitive processing induced by the stimulus [14].

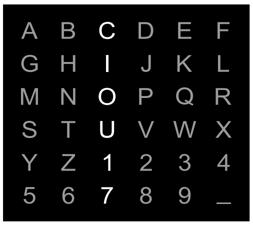
The P300 wave is an endogenous ERP with a latency range of approximately 300 to 600 ms and which is elicited by rare and/or significant stimuli (visual, auditory, or somatosensory). Effectively, the presence of P300 ERPs depends on whether or not a user attends a rare, deviant or target stimulus. This is what makes it possible to use them, in BCI systems, to determine user intent. For an overview of the cognitive theory and the neurophysiological origin of the P300 see [15].

The characteristics of the P300, particularly its amplitude and latency, depend on several factors [16]. Some factors, such as food intake, fatigue or use of drugs, are related to the psychophysical state of the subject [17]. Other factors, such as the number of stimuli, their size and their relative position, are related to the spacial layout of the stimuli [18–20]. Finally, there are several important factors related to the timing and sequence of stimuli. Since these are the focus of this paper, we will devote the rest of this section to summarise what is known about such factors.

Several studies have reported that P300 amplitude increases as target probability decreases (see [14] for a review). The P300 amplitude seems to also be positively correlated with the inter-stimulus interval (ISI) or the stimulus onset asynchrony (SOA) (e.g., see [21–23]). Other studies [24, 25] note that, despite the P300 being clearly affected by target-stimulus probability and ISI, each of these factors also varies the average target-to-target interval (TTI) and hypothesise that TTI is the true factor underlying the P300 amplitude effects attributed to target probability, sequence length, and ISI. The P300 is also sensitive to the order of non-target and target stimuli, most likely because this temporarily modifies target-stimulus probabilities and TTIs. Indeed, there is a positive correlation between P300 amplitude and the number of non-target stimuli preceding a target (e.g., [26, 27]).

In order to avoid the overlap between two consecutive epochs, in psychophysiology the effects discussed above have always been studied with SOAs of approximately one second or more. However, such long SOAs would drastically limit the information transfer rate in BCIs, where SOAs of less than 300 ms are much more common. Also in these cases there is a reduction of the P300 amplitude when two targets are close in a sequence of stimuli. However, this is more likely the result of phenomena like repetition blindness or attentional blink [3] than the TTI effect documented with slower protocols.

The influence of the timing and sequence of stimuli preceding a target on the P300 could be partially explained as the result of "recovery cycle" limitations in the mechanisms responsible for the generation of this ERP [24]. The smaller potentials produced when a target is presented shortly after another target might simply be the result of the brain not having yet reacquired the necessary resources to produce large ERPs.



A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	1	2	3	4
5	6	7	8	9	_

Figure 1. The matrix of characters used in the Donchin speller.

Some ascribe such effects to an inability to consistently generate P300s in the presence of short TTIs [28], suggesting that a smaller average amplitude might be the result of an increase in the fraction of responses to target stimuli that do not present a P300 rather than a modulation of the P300 amplitude in all responses.

In summary, the literature indicates that the order and timing of stimuli have measurable effects of P300s. In an effort to capture and exploit some such effects to improve BCI performance, in this paper we will focus on the *number of non-targets preceding a target stimulus*. Note that we make no claims regarding whether or not such a number is the most primitive factor affecting P300 amplitudes. All we need for our method to improve performance is that P300's amplitude be correlated with this number, which is the case for Donchin's speller, as we will show in Section 3.

2.2. Donchin ERP-based speller

Donchin and Farwell [11] designed a speller based on the P300 ERP, which is now in widespread use. In this speller, users are presented with a 6 by 6 matrix of characters (see figure 1), the rows and columns of which are highlighted for a short period and one at a time. The sequence of flashes is generated randomly without replacement, i.e., a row or column is not allowed to flash again until all other rows and columns have flashed. The users' task is to focus their attention on the character they want to input. Thus, the flashes of the row and column containing the desired character represent target stimuli. As they are relatively rare, they elicit a P300 wave. In principle, a BCI could then identify the row and column of the target character, thereby inferring it, by looking for the largest P300-like responses in the EEG signals acquired immediately after the flashing of all the rows and columns, respectively. Unfortunately, noise makes reliably recognising P300s in single trials very difficult. However, by asking users to focus their attention on the character of interest for multiple blocks of 12 flashes, and averaging either the signal epochs related to each row and column or the corresponding outputs of a P300 classifier, it is possible to infer the target character with acceptable accuracy [29].

3. Stimulus history effects in the Donchin speller

Evidence of the shape of P300s being modulated by the temporal distance between consecutive targets in the Donchin speller has recently been reported [30–32]. In this work, we will take this one step further by modelling and exploiting this phenomenon in order to improve the performance of the speller. Before we do so, however, in this section we want to present experimental evidence further corroborating the results of [30–32]. Related experimental data will be used in Section 5.1 to corroborate our model.

We studied the training set of the two subjects of dataset II from the BCI competition III [33] (results with 12 further subjects will be presented in Section 7.2). Data were collected with the Donchin speller protocol described above, using a SOA of 175 ms. For each subject, the training set consisted of 85 blocks. Each block corresponds to a subject focusing on one character for 15 repetitions of the sequence of the flashes of the 6 rows and 6 columns in the display. Further details on the data can be found in [33].

We bandpass-filtered the signals in the band 0.15 - 5 Hz (HPF: 1600-tap FIR; LPF: 960-tap FIR) to reduce exogenous components at the stimulus presentation frequency (5.7 Hz) and higher. One-second epochs following each flash were extracted. Therefore, for each subject, a total of 15300 (85×15×12) epochs were available, of which 2550 (85×15×2) corresponded to targets.

The set of epochs was partitioned according to the number, h (for "history"), of non-targets presented between the previous target and the stimulus corresponding to the current epoch. For example, for the sequence ...TNNNT..., the second target (T) is assigned to partition t3 because it is preceded by three non-targets (N).

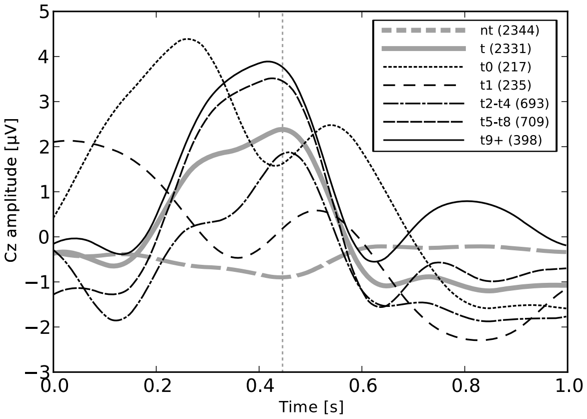
Outlier, e.g., epochs with unusually large deflections due to movements or eye-blinks, were removed using the following procedure. For each set of epochs, the first quartile, $\mathrm{q}_1(i)$, and third quartile, $\mathrm{q}_3(i)$, at each sample, i, were found. Then an acceptance "strip" was defined as the time-varying interval $[\mathrm{q}_1(i) - 1.5\,\Delta\mathrm{q}(i), \mathrm{q}_3(i) + 1.5\,\Delta\mathrm{q}(i)]$ where $\Delta(i) = \mathrm{q}_3(i) - \mathrm{q}_1(i)$ is the interquartile range. Responses falling outside the acceptance strip for more than one tenth of the epoch were rejected. For each partition, the mean, $\mathrm{m}(i)$, and the standard error, $\mathrm{ste}(i)$, of the responses for each class were finally evaluated using the remaining epochs.

Figure 2 shows the average responses obtained from the epoch-partitioning and artefact-rejection procedures described above. The results suggest that despite Donchin's speller being characterised by a fast SOA, significant modulations of the P300 amplitude related to the number, h, of non-targets preceding a stimulus are present. The most significant effect is visible around the peak of the "t" line, that represents the average P300. Indeed, if we focus on around 450 ms for subject A and 350 ms for B, we see that the P300 is hardly visible in the average associated with the "t0" partition which represents two targets in a row. The same is true for the average of partition "t1". The averages for other partitions show a progressive increase in the P300 amplitude as the number of non-targets separating a target from the previous target grows.

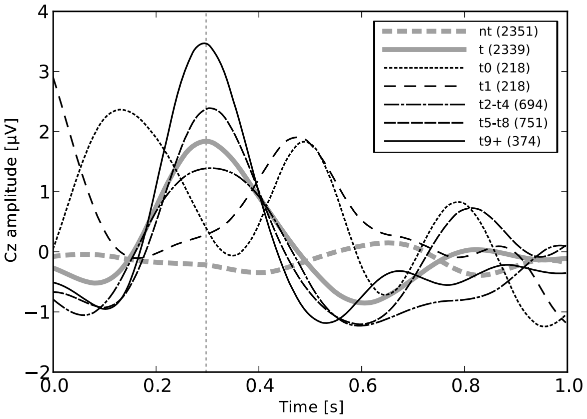
The effect is particularly clear in figure 3 where we plotted P300 amplitudes vs. h for subjects A and B at 450 ms and 350 ms, respectively. Note that, because of the randomisation within each block of 12 stimulus presentations, it is possible, albeit infrequent, for a target stimulus in a block to be followed by up to 20 non-target stimuli before a target stimulus in the following block is presented.

4. Modelling the accuracy of a Donchin speller

An objective of this paper is to model and exploit the P300 modulations documented in the previous section to improve the accuracy of Donchin's speller. As we will see in Section 6, simple modifications to the internal structure of the speller can achieve



Subject A



Subject B

Figure 2. Modulation of the amplitude and shape of P300s as the number of non-targets preceding the current stimulus, h, varies for the two subjects in the BCI competition III Donchin's speller dataset. The thin lines are averages of ERPs recorded in channel Cz for the epochs partitioned according to h (lines "t0" to "t9+"). The thick line labelled "t" is the average of all targets; a vertical dashed line marks the time where this reaches its peak in each plot. The line "nt" is the average of an equally-sized random sample of non-targets. In all cases the number of epochs averaged (after artefact rejection) is shown in brackets.

this. However, before we can proceed with this, we need to describe and model the internal operations of typical Donchin spellers. We do this in this section.

4.1. Algorithm of a Donchin speller

In a Donchin speller, inferring the character the user wants to input is a two-stage process (figure 4). The ERPs produced in the 1-second epochs following the flashing of the rows and columns of characters in the display are acquired across all EEG channels and time steps, resulting in a sequence of 2–D arrays (far left of the figure).

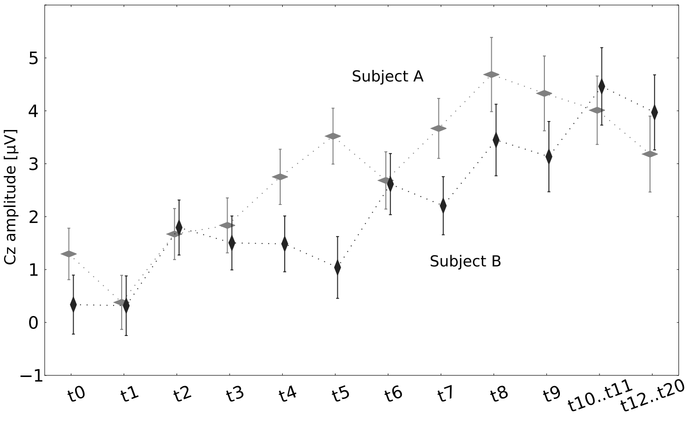


Figure 3. A cross section of the data in figure 2 taken at the time when the average target ERP (line "t" in figure 2) peaks. Whiskers represent standard errors. Note that the groups of partitions "t2-t4", "t5-t8" and "t9+" from figure 2 have been split more finely for a more detailed analysis.

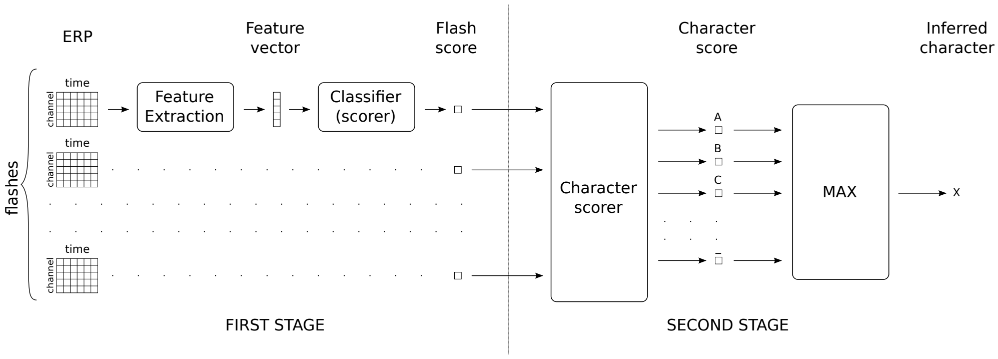


Figure 4. Block diagram of typical implementations of Donchin's speller (see text for details).

In the first stage, each such array is processed by a feature extractor/classifier whose output is expected to contain useful information as to whether the corresponding flash represented a target or a non-target (flash score). In the second stage, an algorithm combines the flash scores for different epochs, assigns a score to each character (character score) and chooses the character with the highest score as its classification for the data. Below we provide details on how the two stages are implemented in current ERP-based BCIs.

As shown in figure 4(left), the first stage often includes a feature extractor/selector that creates a low-pass filtered and sub-sampled version of the epoch following a flash and selects a subset of informative samples and channels. Let us call x_i the resulting vector of features characterising the i-th epoch. Feature vectors are typically processed by a classifier that performs a rotation and a projection of the feature space in the direction that maximises some form of separability measure between the target and non-target classes. Frequently the classifier is used in an unconventional way: it is trained as if it were a binary classifier with a two-class output, but it is then used without applying any thresholding as a continuous scoring function, f. Let $y_i = f(x_i)$

be the flash score produced by such a function in response to feature vector x_i.

As shown in figure 4(right), in its simplest form the second stage can be performed in two steps. The first involves averaging the scores associated with the flashing of each row and column independently and then computing a score for each character by adding up the averages of the corresponding row and column. The second step simply requires choosing the character with the maximum character score [34].

In formulae, in the standard approach, the character at the intersection of row r and column c is given a score

$$\bar{S}_{r,c} = \frac{1}{J} \sum_{\rho \in \mathcal{R}_r} y_\rho + \frac{1}{J} \sum_{\chi \in \mathcal{C}_c} y_\chi \ , \tag{1}$$

where J is the number of repetitions of the sequence of stimuli, y_ρ and y_χ are the outputs of the flash scoring function f in the presence of feature vectors x_ρ and x_χ, respectively, while $\mathcal{R}_r$ and $\mathcal{C}_c$ are sets of indices of the stimuli where row r and column c flashed, respectively. The classifier then chooses the character at position

$$(\hat{r}, \hat{c}) = \arg\max_{(r,c)} \bar{S}_{r,c} \ , \tag{2}$$

in the character grid.

4.2. Model of the accuracy of a Donchin speller

From the description provided in the previous section, it is clear that the accuracy of Donchin's speller depends the performance of the two main stages shown in figure 4. In this section we will construct a probabilistic model that will allow us to compute the overall performance of the speller. We will do so by modelling its second stage in detail, while we will coarse-grain on the internal mechanisms of the first by only modelling its output, i.e., the flash scores. We do this because later in the paper we will modify the second stage so as to exploit target history effects and this will require only knowing and adapting the details of the model of the second stage of the standard speller.

Before we start, we need to model the sequence of stimuli, since the output of a speller and its performance are necessarily functions of the visual stimuli as well as, naturally, of the user's intentions. We model the sequence of $J \times 12$ flashes presented to the subject as stochastic vector $L = (L_1, L_2, L_3, ...)$. Each component L_i of the vector is itself a stochastic variable drawn from the set of outcomes $\{\mathsf{R}_1, \ldots, \mathsf{R}_6, \mathsf{C}_1, \ldots, \mathsf{C}_6\}$ where R_r represents the event "the r-th row flashed", while C_c represents the event "the c-th column flashed". Let us denote with ℓ an instantiation of L and with ℓ_i the instantiation of the element/variable L_i.

The sequence of stimuli needs to be treated as a stochastic variable rather than a constant because of the randomness in the sequence generation process described in section 2.2. Thus, in different repetitions of the experiment different sequences of stimuli will likely be presented to the user. This will happen, for example, if the user needs to enter different letters or the same letter multiple times.

4.2.1. Modelling Stage 1 ERPs are modulated by the nature and type of the stimuli eliciting them. However, even in the presence of identical external stimuli, ERPs are typically affected by a great deal of noise and variability. As a result the feature vectors, x_i, and the flash scores, y_i, produced in the first stage of Donchin's speller should be treated as variable, or, more specifically, as stochastic functions of the variables L_i introduced above.

In principle to fully model the first stage of the speller which corresponds to the process of assigning a score to each flash, one would need to identify the joint probability distribution of the variables y_1, y_2, ... as a function of L. However, this is very hard to do. Therefore, here we will introduce some simplifying assumptions, which, as we will show later, still produce accurate models. Firstly, we will assume that all the scores y_i that are associated with the flashing of a row or column containing the target character are independent and identically distributed, and that the same is true for the scores produced in response of non-target flashes. Secondly, if we denote by Y_{nt} any stochastic variable that represents the flash score in the presence of a non-target and by Y_{t} any variable representing the flash score in the presence of a target, we further assume that such variables have different means, which we will denote with α_{nt} and α_{t}, respectively. Thirdly, we assume that, other than for the mean, such variables have the same distribution, i.e., they can be represented as

$$Y_{\mathrm{nt}} = \alpha_{\mathrm{nt}} + \mathring{Y} \tag{3}$$

$$Y_{\mathrm{t}} = \alpha_{\mathrm{t}} + \mathring{Y} \tag{4}$$

for some assignment of the parameters α_{t} and α_{nt}, where $\mathring{Y}$ is a zero-mean stochastic variable. Finally, we assume that the variable $\mathring{Y}$ is normally distributed with zero mean, i.e., $\mathring{Y} \sim N(0, \sigma^2)$ for some assignment of σ, and we will interpret such variable as the noise in the flash scores.

Naturally, exactly which variables y_i in a sequence are of type Y_{t} and which are of type Y_{nt} depends on whether the i-th flash is that of a row/column containing the target character or not, respectively.

4.2.2. Modelling Stage 2 Based on the observations above, it is clear that the character scores, $\bar{S}_{r,c}$, computed in the second stage of the speller via (1), are averages of stochastic variables (the flash scores, y_i) and, thus, are stochastic themselves. Randomness in the $\bar{S}_{r,c}$ is not only due to the y_i's but also to the fact that the sets of indices $\mathcal{R}_r$ and $\mathcal{C}_c$ in (1) are themselves stochastic. In fact, as one can immediately see, if we formalise their definition, i.e., $\mathcal{R}_r = \{k \mid L_k = \mathsf{R}_r\}$ and $\mathcal{C}_c = \{k \mid L_k = \mathsf{C}_c\}$, they are functions of the stimulus sequence L.

An important consequence of the 36 character scores $\bar{S}_{r,c}$ being stochastic variables is that one cannot say which character has the maximum value: in principle in different instantiations of the 36 scores $\bar{S}_{r,c}$ different characters could present the maximum score. If the target character is at position $(\hat{r}, \hat{c})$, the returned character will be correct if and only if

$$\bar{S}_{\hat{r},\hat{c}} = \max_{r,c} \bar{S}_{r,c} \ , \tag{5}$$

ignoring the zero-probability event of a tie. The objective of this section, computing the *accuracy of the speller*, which we will denote with A, can, therefore, be recast as determining the probability of (5) being the case. In formulae:

$$A = \Pr\left(\bar{S}_{\hat{r},\hat{c}} = \max_{r,c} \bar{S}_{r,c}\right) . \tag{6}$$

Let us start by rewriting (1) in terms of stochastic variables:

$$\bar{S}_{r,c} = \frac{1}{J} \sum_{\rho \in \mathcal{R}_r} Y_\rho + \frac{1}{J} \sum_{\chi \in \mathcal{C}_c} Y_\chi \ , \tag{7}$$

where Y_ρ and Y_χ now are stochastic score variables. This, in turn, can be rewritten as

$$\bar{S}_{r,c} = \bar{U}_r + \bar{V}_c \tag{8}$$

where $\bar{U}_r$ is the average of the J variables Y_ρ for $\rho \in \mathcal{R}_r$, while $\bar{V}_c$ is the average of the J variables Y_χ for $\chi \in \mathcal{C}_c$. Note that both term in (8) are stochastic variables.

The random variable $\bar{U}_r$ has a distribution that depends on whether or not $r = \hat{r}$. Specifically, if $r = \hat{r}$, the variables Y_ρ in (7) are all distributed like Y_t. Therefore, $\bar{U}_{\hat{r}}$ is the average of J independent instances of Y_t and, so, it is a normal variable $\bar{U}_{\hat{r}} \sim N(\mu_{\bar{U}_{\hat{r}}}, \sigma^2_{\bar{U}_{\hat{r}}})$ with $\mu_{\bar{U}_{\hat{r}}} = \alpha_\mathrm{t}$ and $\sigma^2_{\bar{U}_{\hat{r}}} = \sigma^2/J$. Similarly, if $r \neq \hat{r}$, the variables Y_ρ in (7) are all distributed like Y_nt. Thus, if we denote with $*$ any non target row, $\bar{U}_*$ is a normal variable $\bar{U}_* \sim N(\mu_{\bar{U}_*}, \sigma^2_{\bar{U}_*})$ with $\mu_{\bar{U}_*} = \alpha_\mathrm{nt}$ and $\sigma^2_{\bar{U}_*} = \sigma^2/J$.

Using similar arguments, it is easy to show that the variables $\bar{V}_{\hat{c}}$ and $\bar{V}_*$ are distributed like $\bar{U}_{\hat{r}}$ and $\bar{U}_*$, respectively.

Because of the independence of the variables Y_i and the geometry of the speller's grid, for any $r_1 \neq r_2$ the set of Y_i averaged in $\bar{U}_{r_1}$ and the set of the Y_i averaged in $\bar{U}_{r_2}$ are disjoint. The same is true of the $\bar{V}_c$ variables. In other words, the 6 variables $\bar{U}_r$ and the 6 variables $\bar{V}_c$ are all mutually independent. Therefore,

$$\bar{S}_{\hat{r},\hat{c}} = \max_{r,c} \bar{S}_{r,c} \iff \bar{U}_{\hat{r}} > \bar{U}_\mathrm{max} = \max_{r\neq\hat{r}} \bar{U}_r \;\wedge\; \bar{V}_{\hat{c}} > \bar{V}_\mathrm{max} = \max_{c\neq\hat{c}} \bar{V}_c \tag{9}$$

and, from (6),

$$A = \Pr\left(\bar{S}_{\hat{r},\hat{c}} = \max_{r,c} \bar{S}_{r,c}\right) = \Pr\left(\bar{U}_{\hat{r}} > \bar{U}_\mathrm{max}\right)\, \Pr\left(\bar{V}_{\hat{c}} > \bar{V}_\mathrm{max}\right). \tag{10}$$

Hence, to make further progress in our modelling effort, we need to compute $\Pr\left(\bar{U}_{\hat{r}} > \bar{U}_\mathrm{max}\right)$ and $\Pr\left(\bar{V}_{\hat{c}} > \bar{V}_\mathrm{max}\right)$.

Let us start with $\Pr\left(\bar{U}_{\hat{r}} > \bar{U}_\mathrm{max}\right)$. Having already computed the distribution for $\bar{U}_{\hat{r}}$, first, we need to compute the distribution of $\bar{U}_\mathrm{max}$. To do so, we exploit the fact that any number greater than the maximum of multiple variables is greater than any individual variable. Therefore, for independent variables, the cumulative distribution function of their maximum is the product of the cumulative distribution functions of the original distributions. That is

$$\mathrm{F}_{\bar{U}_\mathrm{max}}(x) = \Pr\left(\bar{U}_\mathrm{max} < x\right) = \prod_{r\neq\hat{r}} \Pr\left(\bar{U}_r < x\right) = \left[\mathrm{F}_{\bar{U}_*}(x)\right]^5 ,$$

where the symbol $\mathrm{F}_.(x)$ represents cumulative distribution functions.

We are now in a position to compute $\Pr\left(\bar{U}_{\hat{r}} > \bar{U}_\mathrm{max}\right)$. In general, given two independent random variables X and Y, the probability of $X > Y$ is

$$\begin{aligned}\Pr(X > Y) &= \int_\mathbb{R} \int_{-\infty}^{x} \mathrm{f}_X(x)\, \mathrm{f}_Y(y)\; \mathrm{d}y\; \mathrm{d}x \\ &= \int_\mathbb{R} \mathrm{f}_X(x)\, \mathrm{F}_Y(x)\; \mathrm{d}x\ .\end{aligned}$$

Thus,

$$\begin{aligned}&\Pr\left(\bar{U}_{\hat{r}} > \bar{U}_\mathrm{max}\right) = \\ &\int_\mathbb{R} \frac{1}{\sigma_{\bar{U}_{\hat{r}}}} \mathrm{normpdf}\left(\frac{x - \mu_{\bar{U}_{\hat{r}}}}{\sigma_{\bar{U}_{\hat{r}}}}\right) \left[\mathrm{normcdf}\left(\frac{x - \mu_{\bar{U}_*}}{\sigma_{\bar{U}_*}}\right)\right]^5 \mathrm{d}x\ ,\end{aligned} \tag{11}$$

where

$$\text{normpdf}\,(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$
$$\text{normcdf}\,(x) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right) .$$

Using the same approach one can easily compute $\Pr\left(\bar{V}_{\hat{c}} > \bar{V}_{\max}\right)$. By substituting that result and (11) into (10), we finally obtain the following expression for the *accuracy of Donchin's speller*:

$$A = \left\{ \int_{\mathbb{R}} \frac{1}{\sigma_{\bar{U}_{\hat{r}}}} \text{normpdf}\left(\frac{x - \mu_{\bar{U}_{\hat{r}}}}{\sigma_{\bar{U}_{\hat{r}}}}\right) \left[\text{normcdf}\left(\frac{x - \mu_{\bar{U}_*}}{\sigma_{\bar{U}_*}}\right)\right]^5 \mathrm{d}x \right\}^2 . \quad (12)$$

This expresses the accuracy of the speller as a function of the characteristics of the score distributions for targets and non-targets obtained from the first stage of the speller. Therefore, in order to use this formula to measure the performance of a particular realisation of Donchin's speller, one needs to empirically identify the means and standard deviation of the outputs of the flash scoring block for that implementation and then feed them into (12). We will postpone this parameter-identification phase until section 6, where we will develop a more accurate model of the first stage which will take target history effects into account.

4.3. Generalising the model

While the system described and modelled above is the most typical incarnation of Donchin's speller, also other types of ERP-based speller have been considered in the literature. By using procedures similar to the one we followed to obtain (12), we analysed a variety of such systems and modelled their accuracy. The spellers we considered are listed in the first column of table 1. These include Donchin spellers with different matrix sizes and different features for the distributions of flash scores for targets and non-targets, a sequential speller, and a generic multi-choice selection BCI.

Interestingly, we found that the accuracy of all such systems is represented by the following general equation:

$$A = \left\{ \int_{\mathbb{R}} \text{normpdf}\,(\xi) \ \left[\text{normcdf}\,(\beta\,\xi + \gamma)\right]^\eta \,\mathrm{d}\xi \right\}^\lambda . \quad (13)$$

For example, it is easy to derive (12) from (13) by performing the substitution $x = \sigma_{\bar{U}_{\hat{r}}}\,\xi + \mu_{\bar{U}_{\hat{r}}}$ and with an appropriate choice of the constants β, γ, η and λ (indicated as case (a) in table 1). The values of the constants necessary to derive the accuracy of other spellers are shown in the table. Other more complex BCI systems could similarly be modelled.

5. Modelling *h*-related effects

In section 4.2.1 we introduced a simple model of the first stage of Donchin's speller, which, in conjunction with a model of the second stage, was then used to derive an expression for the accuracy of the speller. The focus of this paper is really the second stage of the speller. However, the accuracy of the second stage, and, hence, of the speller, can only be evaluated if the statistical properties of the scores produced by the

Table 1. Parameters to be used in (13) to model the accuracy of different ERP-based brain-computer interfaces

	Type of BCI	β	γ	η	λ	notes
a	Donchin speller choosing the character at the intersection between the row and column with the maximum average output (e.g., [34]);	1	$\frac{\alpha_{\rm t}-\alpha_{\rm nt}}{\sigma}\sqrt{J}$	5	2	$\alpha_{\rm t}$ and $\alpha_{\rm nt}$ are the average flash scores for target and non-target epochs, respectively; σ^2 is the variance of flash scores for both target and non-target epochs
b	Same as in (a) in the hypothesis of heteroscedasticity	$\frac{\sigma_{\rm t}}{\sigma_{\rm nt}}$	$\frac{\alpha_{\rm t}-\alpha_{\rm nt}}{\sigma_{\rm nt}}\sqrt{J}$	5	2	$\alpha_{\rm t}$ and $\alpha_{\rm nt}$ are as in (a); $\sigma^2_{\rm t}$ and $\sigma^2_{\rm nt}$ are the variances of the flash scores for target and non-target epochs, respectively
c	Same as in (a), but with a generic $M \times M$ matrix of characters	1	$\frac{\alpha_{\rm t}-\alpha_{\rm nt}}{\sigma}\sqrt{J}$	$M-1$	2	$\alpha_{\rm t}$, $\alpha_{\rm nt}$, σ^2 is as in (a)
d	Single character (sequential) Donchin speller; selecting the character with the highest average score (e.g., [35])	1	$\frac{\alpha_{\rm t}-\alpha_{\rm nt}}{\sigma}\sqrt{J}$	35	1	$\alpha_{\rm t}$, $\alpha_{\rm nt}$, σ^2 is as in (a)
e	Selection of one item out of a set of $N_{\rm C}$ choices (e.g., [36])	1	$\frac{\alpha_{\rm t}-\alpha_{\rm nt}}{\sigma}\sqrt{J}$	$N_{\rm C}-1$	1	$\alpha_{\rm t}$, $\alpha_{\rm nt}$, σ^2 is as in (a)
f	Donchin speller with h-related effects; using (16) as a scoring function	$\frac{\sigma_{\bar{U}''_{\hat{r}}}}{\sigma_{\bar{U}''_{*}}}$	$\frac{\mu_{\bar{U}''_{\hat{r}}}-\mu_{\bar{U}''_{*}}}{\sigma_{\bar{U}''_{*}}}$	5	2	$\mu_{\bar{U}''_{\hat{r}}}$, $\sigma_{\bar{U}''_{\hat{r}}}$, $\mu_{\bar{U}''_{*}}$ and $\sigma_{\bar{U}''_{*}}$ are derived in Appendix A
g	Donchin speller with h-related effects; using (1) as a scoring function (equivalent to the approach in [34])	$\frac{\sigma_{\bar{U}''_{\hat{r}}}}{\sigma_{\bar{U}''_{*}}}$	$\frac{\mu_{\bar{U}''_{\hat{r}}}-\mu_{\bar{U}''_{*}}}{\sigma_{\bar{U}''_{*}}}$	5	2	$\mu_{\bar{U}''_{\hat{r}}}$, $\sigma_{\bar{U}''_{\hat{r}}}$, $\mu_{\bar{U}''_{*}}$, $\sigma_{\bar{U}''_{*}}$ are as in (f), after setting $w(h) = 1\ \forall h$

first stage are fully specified. In this section we want to extend the model of the first stage developed above so as take the modulations of P300s associated with stimulus history documented in section 3 into account. In section 6 we will propose and model a new algorithm for the second stage of the speller which will exploit such effects.

5.1. Model of the first stage in presence of h-related effects

So far we have modelled the flash scores produced by the first stage of the speller as a normally distributed stochastic variable whose mean depends exclusively on whether or not the flash is a target or a non-target. As discussed above, however, h strongly affects the shape and amplitude of P300s and, so, it is reasonable to expect that this will also affect flash scores for target stimuli.

To take this into account, we propose to modify (4) by creating a separate version of it for each value of h to which significant P300-amplitude changes can be ascribed. In other words we replace it with the following:

$$Y_{{\rm t}h} = \alpha_{{\rm t}h} + \mathring{Y} \qquad (14)$$

where $h = 0, 1, ..., h_{\max}$ and the means $\alpha_{{\rm t}h}$ account for the dependence of target flash scores on h. Based on the evidence gathered in section 3, we chose $h_{\max} = 9$. All epochs with $h \geq 9$ were artificially assigned to class $h = 9$, which we will term "9+" in the following to make it easier to remember its contents. We, therefore, obtained ten versions of (14).

While we have no evidence that h significantly affects also the shape and amplitude of the ERPs produced in the presence of *non-targets* in Donchin speller, in principle, this might happen with particular choices for the shape and timing of the stimuli or in other types of BCI. These situations could easily be modelled by generalising (3) as we did for the targets, e.g., by considering a model of the form

$$Y_{\mathrm{nt}h} = \alpha_{\mathrm{nt}h} + \overset{\circ}{Y} \tag{15}$$

In appendix Appendix D we will look at possible causes of TTI-related modulations of flash scores in the presence of non-targets and we will explain how the theory and methods developed in the paper would be affected by the use of (15). However, as we will show at the end of section 5.3, this level of sophistication is unnecessary for Donchin speller. So, we will continue to use the simpler (3) to model non-target scores.

5.2. Model identification

Based on the choices we made in the previous section, in order to model the first stage of the speller in the presence of h related effects we need to identify 12 parameters: the variance, σ^2, of the noise variable, $\overset{\circ}{Y}$, in (3) and (14); the α_{nt} parameter in (3); and the ten $\alpha_{\mathrm{t}h}$ parameters in (14). Before we can proceed with this task, however, we need to choose a particular implementation of the first stage of the speller.

For the purpose of demonstrating the ideas, we decided to use an existing high-performance approach. More specifically, we borrowed the first stage from the work of Rakotomamonjy and Guigue [29] which won the BCI Competition III for the Donchin speller. Namely, we used an approach they called "Ensemble SVM without channel selection" which is easy to implement and outperforms other alternatives when using only 5 repetitions of the 12 stimuli to classify a character. In this approach, an ensemble of classifiers is used. The datasets are split into several subsets and a linear support vector machine (SVM) classifier is trained on each of them. When classifying unseen test data, the outputs of all classifiers are averaged to give a flash score representing the "targetness" of the flash.

We analysed the distribution of the flash scores produced by this approach on the BCI Competition data used also in section 3. As in [29], we divided the 85 data blocks of each subject into 17 subsets of 5 blocks each. To obtain realistic generalisation statistics and avoid overfitting problems, we applied a leave-one-out approach estimating the output of the ensemble classifier on a subset of data by averaging the output of 16 classifiers which were not trained using that subset. By repeating this estimation process for each of the 17 subsets we obtained a reliable estimate of the output of the ensemble on the full dataset of each subject.

We partitioned the data into two classes based on whether the corresponding input epochs represented targets (t) or non-targets (nt). We represented the class of a flash as $\theta \in \{\mathrm{t}, \mathrm{nt}\}$. Then, we divided each class into subclasses according to h, for $h \in \{0, \ldots, h_{\max}\}$. We, therefore, obtained a total of $2 \times (h_{\max} + 1) = 20$ partitions, each represented by a tuple (θ, h).

We computed the 12 parameters of the model of the first stage described in section 5.1 as follows. First, we evaluated the mean and variance of the flash scores associated with each partition (θ, h) using the 5%-truncated mean, $a_{\theta h}$, and 5%-truncated variance, $s^2_{\theta h}$. These are robust measures of central tendency and spread

Table 2. Parameters for the model in (3) and (14) estimated from the datasets of the two subjects in the BCI Competition III dataset.

Subject	$\alpha_{\mathrm{t}0}$	$\alpha_{\mathrm{t}1}$	$\alpha_{\mathrm{t}2}$	$\alpha_{\mathrm{t}3}$	$\alpha_{\mathrm{t}4}$	$\alpha_{\mathrm{t}5}$	$\alpha_{\mathrm{t}6}$	$\alpha_{\mathrm{t}7}$	$\alpha_{\mathrm{t}8}$	$\alpha_{\mathrm{t}9+}$	α_{nt}	σ
A	-1.183	-1.188	-0.936	-0.867	-0.730	-0.566	-0.669	-0.588	-0.494	-0.534	-1.846	0.982
B	-0.824	-0.783	-0.428	-0.490	-0.313	-0.417	-0.165	-0.212	-0.139	-0.198	-1.719	0.876

which are less sensitive to outliers than their non-truncated counterparts. We then computed the average of the variances of the partitions

$$s^2 = \frac{1}{2\,(h_{\max}+1)} \sum_{\theta} \sum_{h} s_{\theta h}^2$$

and the average of the means of the partitions representing non-targets

$$a_{\mathrm{nt}} = \frac{1}{h_{\max}+1} \sum_{h} a_{\mathrm{nt}h} \; .$$

Finally, we estimated the model's parameters as follows: $\sigma^2 = s^2$, $\alpha_{\mathrm{nt}} = a_{\mathrm{nt}}$ and $\alpha_{\mathrm{t}h} = a_{\mathrm{t}h}$ for $h = 0, ..., h_{\max}$.

We performed this model identification process independently for the two subjects in the dataset II of the BCI Competition III. The resulting parameters are reported in table 2. Note how in both cases $\alpha_{\mathrm{t}h}$ markedly increases as h increases, as also confirmed by linear regression. For example, the difference in flash score between non-targets and targets for $h = 8$ is twice as much as for $h = 0$. As the classification's accuracy strongly depends on the distance between the two classes, it is clear that epochs with small h (i.e., one target presented shortly after the previous one) represent an unfavourable condition for the speller.

5.3. Model validation

Having defined the model for the first stage and determined its parameters, we need to assess whether this model fits the actual flash scores resulting from the datasets.

We used a Kolmogorov-Smirnov test to check whether we could reject the hypothesis that the flash scores in each partition could be the result of sampling from the corresponding distributions in (3) and (14) with the parameters in table 2. Results indicated that we cannot reject this hypothesis for 19 out of 20 partitions (we discuss the possible reasons for the exception, partition nt1, later in this section). This confirms the substantial correctness of our assumption that the variables in (3) and (14) are normally distributed and all have the same variance.

To visually evaluate the goodness of the fit between our models and the actual flash scores produced by the ensemble SVM, we show the QQ-plots [37] associated to each (θ, h) partition of each subject in figure 5. They confirm that model fits very well the data in the key part of the distributions, i.e., between the 5th and the 95th percentiles.

It is natural to ask why the Kolmogorov-Smirnov test indicated that the data in partition nt1 are unlikely to be the result of sampling from the stochastic variable $Y_{\mathrm{nt}} = \alpha_{\mathrm{nt}} + \overset{\circ}{Y}$ (i.e., (3)). Obviously, with the standard confidence level of 5% which we adopted we should expect a statistical test to give incorrect answers on average once in 20 applications simply because of stochastic effects. However, this does not appear to be the main reason for this effect. The most likely cause is that the mean score for

Subject A

Subject B

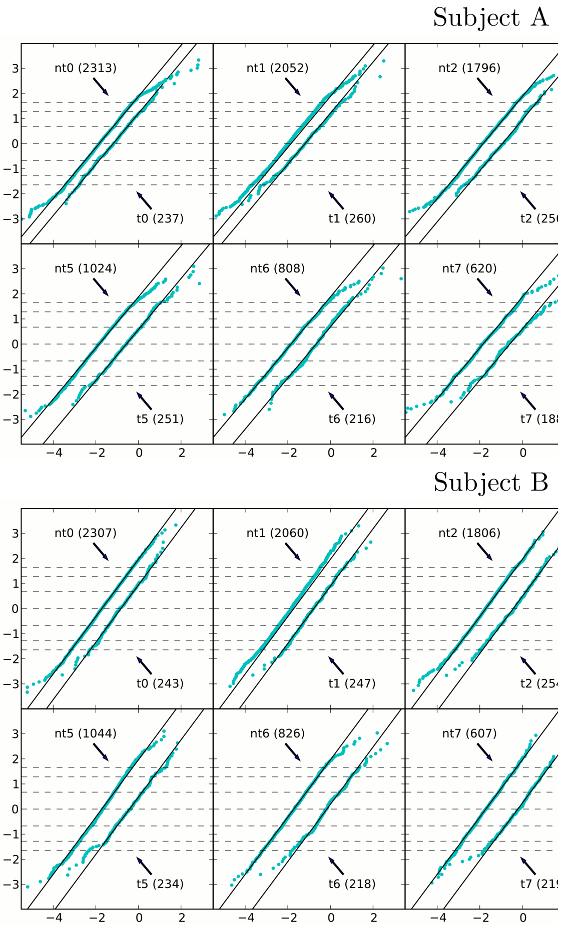


Figure 5. QQ-plots for the flash scores in the (θ, h) partitions for subjects A and B in the BCI Competition dataset. For each partition and type of flash the quantiles corresponding to the samples in the partition are plotted against the corresponding quantiles of a standardized normal distribution. The dark thin lines represent the QQ-plots we would expect to see if the scores exactly followed a normal distributions with the parameters listed in table 2. The dots represent actual data points. The horizontal dashed lines represent the 5th, 10th, 25th, 50th, 75th, 90th, 95th percentiles for the standardized normal distribution. So, data points below the bottom dashed line or above the topmost dashed line are in the tails of the distribution.

the epochs in partition nt1 is slightly smaller than for other non-target partitions, as shown in figure 6 which plots the average score for targets and non-targets for different h values. In Appendix D we provide a possible explanation for this anomaly of nt1.

Irrespective of the causes of the minor anomaly of partition nt1, it is clear from the plots in figure 6 that the assumption that non-target score averages are not substantially influenced by h is essentially correct for Donchin's speller. Thus, the adoption of a more sophisticated model (such as (15)), over the simpler (3), is unnecessary.

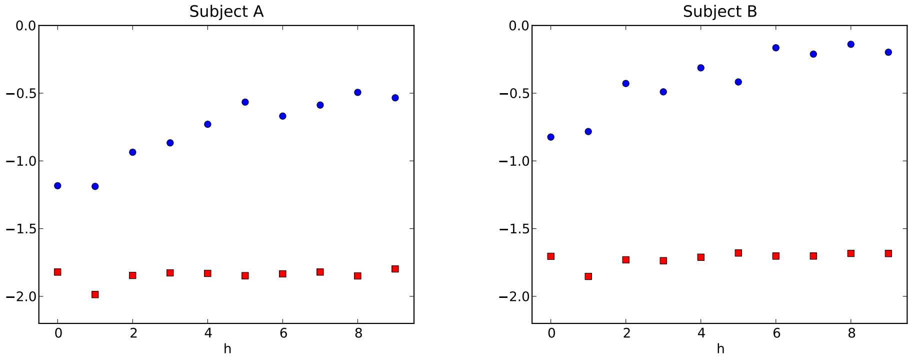


Figure 6. Average flash scores for targets (circles) and non-targets (squares) as a function of h. The target data correspond to the values in table 2.

6. Exploiting h-related effects

In this section we will propose a new classification algorithm for Donchin's speller that exploits the flash-score modulations modelled and documented in section 5. This uses a modified second processing stage (where the system combines the flash scores into character scores and makes a final decision) which, in turn, will require some modification to the model of the accuracy of the speller presented in section 4.2.2.

6.1. A new character-scoring algorithm

Let us start by introducing some useful notation. Let B be a set of integers representing the indices of some events in L and let $\mathrm{h}(i, B)$ be a function that returns the number of flashes between the i-th flash and the last event in B preceding such a flash, i.e., $\mathrm{h}(i, B) = i - \max[B \cap \{1, 2, \ldots, i-1\}] - 1$. We want to apply this function to $B = \mathcal{R}_r \cup \mathcal{C}_c$, which represents the set of indices of events where either the row or the column of the character at position (r, c) in the character-grid of Donchin's speller flashed.

Let us denote by $\hat{r}$ and $\hat{c}$ the row and column containing the target character, respectively. Then, if the i-th flash is a target, $\mathrm{h}(i, \mathcal{R}_{\hat{r}} \cup \mathcal{C}_{\hat{c}})$ gives the value of h for the i-th flash, i.e., the number of non-targets preceding the target flash i. Note, however, that in general the coordinates of the target character are unknown to the speller. So, the function $\mathrm{h}(i, \mathcal{R}_r \cup \mathcal{C}_c)$ should not be interpreted as representing the true h of the i-th flash, but as its *h value on the hypothesis that the target character is at position* (r, c). To reiterate, only when scoring the target character $(\hat{r}, \hat{c})$, the function h gives the true h value of the i-th flash.

To account for the effects of h, we propose to modify the scoring formula in (1) by weighing flash scores based on the values returned by the function $\mathrm{h}(i, \mathcal{R}_r \cup \mathcal{C}_c)$. The scoring function we used is as follows:

$$\begin{aligned}\bar{S}_{r,c} = &\frac{1}{J} \sum_{\rho \in \mathcal{R}_r} w(\mathrm{h}(\rho, \mathcal{R}_r \cup \mathcal{C}_c))\,(y_\rho - b) \\ &+ \frac{1}{J} \sum_{\chi \in \mathcal{C}_c} w(\mathrm{h}(\chi, \mathcal{R}_r \cup \mathcal{C}_c))\,(y_\chi - b)\end{aligned} \qquad (16)$$

where w is a function that determines the weight to be used when combining flash scores. Note that we also added a parameter, b, which allows us to change the bias of the first stage effectively determining the balance between sensitivity and specificity of the classifier. It is easy to see that (1) is a special case of (16) obtained with the substitutions $w \equiv 1$ and $b = 0$.

In the version of Donchin's speller considered in this paper, the codomain of the function $\mathrm{h}(i, \mathcal{R}_r \cup \mathcal{C}_c)$ is the set of integers $\{0, 1, \cdots, 20\}$ (see discussion at the end of section 3). So, the algorithm only evaluates the function w on points in that set. Thus, to fully specify the behaviour of the algorithm in (16) we need to know the value of the parameter b and the (codomain) values taken by the function w for domain values in the set $\{0, 1, \cdots, 20\}$. In other words, our scoring technique has a total of 22 degrees of freedom which need to be fixed before the algorithm can be used. We will do this in section 6.3 by using the theoretical model of (16) developed in the next section.

6.2. Modelling the accuracy of the new algorithm

In section 4.2 we modelled the accuracy of the standard Donchin speller disregarding the fact that flash scores are affected by h. We now need to modify that model to consider both the influence of h on target-flash scores (which we studied in section 5) and also the use of the weighted scoring function in (16).

The procedure we used to derive the new model follows approximately the same strategy as in section 4.2: we need to define and study two stochastic variables representing the weighted sum of row and column flash scores, respectively, and then work out with which probability the row and column containing the target will have a higher score than any other row and column.

Since the calculations involved are quite laborious, we provide the detailed treatment in Appendix A. Here it is sufficient to note that the final model of the accuracy of the speller in the presence of h related effects and using a weighted character-scoring function (16) is also a member of the family defined in (13), as was the case for the ordinary speller, although, naturally, the parameters β and γ are quite different in the two cases (see table 1, rows (a) and (f)).

6.3. Model identification/Algorithm optimisation

In principle, the probabilistic model of the accuracy of a Donchin speller controlled by the scoring function (16) developed in the previous section and in Appendix A gives us the opportunity of optimising the algorithm's parameters in such a way as to maximise performance. However, in practice, optimally identifying the 22 parameters of the algorithm (and of the corresponding model) from empirical data without incurring in overfitting problems would require very extensive datasets, which are rarely available in BCI. Below we describe two workarounds for this problem and the reasons for our particular choices.

In all the data available to us, it is clear that the effect of h on the P300 saturates for large enough values of h (e.g., see figures 2 and 3). So, our first step to reduce the number of degrees of freedom of the algorithm was to clip the values returned by the function $\mathrm{h}(i, \mathcal{R}_r \cup \mathcal{C}_c)$ to an upper value of $h_{\max} = 9$, thereby reducing the domain values of the function w in (16), and, thus, its degrees of freedom, to 10 from the original 21, without significantly affecting performance. Our second step was to adopt a particularly simple form for w which has only three degrees of freedom, thereby making a reliable model identification possible even from relatively small datasets. More specifically, we chose w to be a linear splinedefined by the knots (abscissas) $[0, 3, 6, h_{\max}]$ and the corresponding ordinate values $[1, 1 + c_1, 1 + c_2, 1 + c_3]$. With these choices, optimising (16) requires finding appropriate values for a parameter vector $\boldsymbol{q} = [c_1, c_2, c_3, b] \in \mathbb{R}^4$.

We chose this particular form for w for two further reasons. Firstly, our representation of w ensures that the standard Donchin speller is part of our search space. It is easy to see this by using the assignment $\boldsymbol{q} = \boldsymbol{0}$, which corresponds to setting $b = 0$ and all the weights in (16) to 1. In this case, (16) collapses down to (1) and the model of the accuracy of the speller developed in the previous section gives us the accuracy of an ordinary Donchin speller in the presence of h related effects, which, however, are not exploited when scoring characters (a situation depicted in row (g) of table 1). So, if for whatever reason using a weighted scoring function is not a good choice for a particular BCI, an optimisation algorithm can easily discover this and fall back on the standard scoring method. The second reason for considering w functions in the neighbourhood of the constant function $w \equiv 1$ is a technical one. As discussed in Appendix A.3, with this choice the estimation errors produced by our model due to simplifying assumptions are minimum. So, the optimal parameters for the algorithm obtained by via our analytic approach are more likely to be close to the real global optimum for the speller.

6.4. Performance differences between the standard speller and our speller

Having constructed theoretical models of the accuracy of the standard speller and our speller, it is easy to see whether, with an appropriate choice of the parameter vector $\boldsymbol{q}$, the use of (16) can improve performance w.r.t. using (1). All one has to do to study the relative performance of the two spellers is to evaluate the difference between the values taken by (13) when using the parameters in rows (f) and (g) of table 1, respectively.

For space limitations, here we cannot provide full details on such a comparison: even with the constraints imposed on the new algorithm in section 6.3, we still have a 5–dimensional parameter space: the 4 components of $\boldsymbol{q}$ plus the repetition number, J. However, below we will look at performance differences in a representative subset of the parameter space.

The contour plots on the left of figure 7 show the difference between the accuracy (percentage of correctly spelled characters) obtained when using a weighted scoring function and the accuracy of the standard Donchin's speller for the two subjects in dataset II of the BCI Competition III. The figure represents the subset of parameter space where $J = 5$, $c_3 = c_2$ and $b = b_0 \overset{\text{def}}{=} \frac{1}{2}(\alpha_{\text{nt}} + \frac{1}{h_{\max}+1}\sum_h \alpha_{th})$, which corresponds to ensuring that the average output for a balanced data set containing the same number of targets and non-targets is zero. With these simplifications we are left with only two degrees of freedom, c_1 and c_2, and we can, therefore, visualise accuracy differences using a plot. Specifically, in figure 7(left, top and bottom) we studied how the performance of the two spellers differed as the parameters c_1 and c_2 took all possible values in the set $\{-0.3, -0.2, \ldots, 2.0\}$.

Interestingly, we see that there are choices for c_1 and c_2, where the use of a weighted scoring function worsens performance. However, an appropriate choice of c_1 and c_2 can produce significant performance improvements: up to approximately 4% for subject A and up to approximately 1% for subject B. Even bigger improvements can be obtained if one allows (as we will in section 7) the independent optimisation of the 4 components of $\boldsymbol{q}$ and the repetition number, J.

6.5. Monte Carlo validation

In section 7 we will indirectly corroborate the veracity of the performance predictions made in the previous section by our model by testing the weighted scoring method experimentally on two independent test sets. However, before we do that, we would like to more directly and independently validate the model of our second stage and confirm its predictions. In this section we will do this via Monte Carlo simulations.

To keep the effort of this expensive form of validation under control and for an easier visualisation of results, we again considered a subset of the 5–dimensional parameter space of the system. Namely, we set $J = 5$, $c_3 = c_2$ and $b = b_0$ (as we did in the previous section) and ran a Monte Carlo simulation for each possible pair $(c_1, c_2) \in \{-0.3, -0.2, \ldots, 2.0\}^2$.

The simulation was performed as follows. For each (c_1, c_2) we generated 100 million simulated sequences of stimuli using the following algorithm:

(i) Select a target character out of the possible 36 (e.g., “A”, at position $(1, 1)$, in the top left corner of the display);

(ii) Generate a vector ℓ containing $J \times 12$ components. The vector is constructed by concatenating J random permutations of $\{\mathsf{R}_1, \ldots, \mathsf{R}_6, \mathsf{C}_1, \ldots, \mathsf{C}_6\}$ (e.g., $\ell = \mathsf{C}_1, \mathsf{R}_3, \mathsf{R}_5, \mathsf{R}_2, \mathsf{C}_2, \mathsf{R}_1, \mathsf{C}_6, \mathsf{C}_4, \mathsf{R}_6, \mathsf{C}_3, \mathsf{R}_4, \mathsf{C}_5, \mathsf{R}_1, \ldots$);

(iii) Scan ℓ, element by element, and label each element, ℓ_i, with the corresponding (θ, h) partition, i.e., according to whether ℓ_i is the target row or column and to the number of non-targets preceding it in ℓ (e.g., the labelling of the example sequence shown above when the target is “A” is: $\mathrm{t}h_{\max}$, nt0, nt1, nt2, nt3, t4, nt0, nt1, nt2, nt3, nt4, nt5, t6, ...);

(iv) Create a vector $\boldsymbol{y}$ of size $J \times 12$. The i-th element of $\boldsymbol{y}$ is obtained by randomly sampling from the set of flash scores associated with the partition assigned to element ℓ_i in the previous step;

(v) Apply the scoring function in (16) to $\boldsymbol{y}$ and ℓ (the latter is used to find r and c) and check whether the character with highest $\bar{S}_{r,c}$ is the target character.

(vi) Perform the same as in (v) with the non-weighted scoring function in (1).

By counting how many times the procedures in (v) and (vi) correctly classify a character over a large number of repetitions and across all characters, one can precisely estimate the accuracy of the speller with the weighted and non-weighted scoring functions, respectively. Comparing the results one can see the performance difference associated with each particular choice of (c_1, c_2).

Figure 7(right, top and bottom) shows the results of these Monte Carlo simulations as contour plots, side by side with the results estimated through our analytical model, for our two subjects. Let us analyse these results.

Firstly, we should note the striking similarity between the contour plots obtained from the model and those resulting from Monte Carlo simulations. Apart from minor numerical differences, for each subject the shape and number of the contour lines match very closely across plots. Secondly, for both subjects, Monte Carlo simulations confirm that by appropriately weighting the flash scores, i.e., using (16) instead of (1), we can improve the accuracy of the speller considerably. Thirdly, the peaks of the

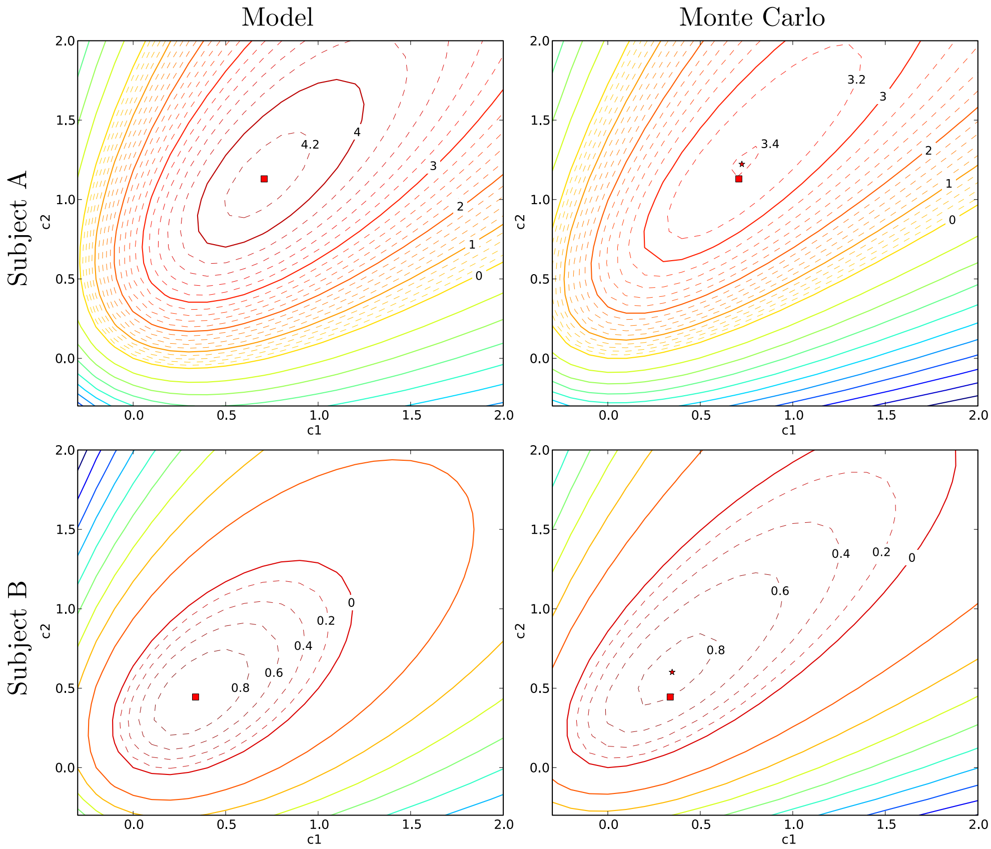


Figure 7. Contour plots of the difference between the accuracy (expressed as percentage of correctly spelled characters) obtained with a weighted scoring function in (16) and the accuracy of the original method. The improvement is plotted as a function of c_1 and c_2, the parameters that modulate the weighting function $w(h)$. On the left we show the improvement estimated using the mathematical model as described in section 6.4. On the right we show the result obtained through the Monte Carlo simulation described in section 6.5. A square represents the point (c_2, c_1) of maximum improvement according to the model, while a star is the maximum according to the Monte Carlo simulation. The similarity between the left and right plots is striking, as well as the closeness of the maxima found by the two approaches, suggesting the substantial correctness of our mathematical model.

performance improvement in the (c_1, c_2) plane predicted with the two methods are also extremely close. All of this strongly corroborates the validity of our models.†

† In principle, the Monte Carlo technique we have used to validate our model would itself need to be validated. We have not done a direct validation of this because of the technical difficulties and need for large datasets associated with such a validation. However, one should not infer from this that nothing has been proved in this section. The chances of two radically distinct and independent approaches to modelling a system as complex as Donchin's speller giving almost identical results and, yet, being both incorrect, are very low. Also, as we will see in section 7.1, optimising $\boldsymbol{q}$ based on the predictions of our model and then testing the system on independent data gives accuracy improvements almost identical to those predicted by the model. So, in reality, Monte Carlo simulations, our theoretical model and the testing of our scoring algorithm on unseen data corroborate one another.

Table 3. Comparison of the accuracy (%) of the speller when using the optimised weighted scoring ("opt $\boldsymbol{q}$") w.r.t. the non-weighted version ("$\boldsymbol{q} = \mathbf{0}$").

Subject	Algorithm	J				
		1–3	4–6	7–9	10–12	13–15
A	opt $\boldsymbol{q}$	35.3	71.7	84.7	93.3	98.0
	$\boldsymbol{q} = \mathbf{0}$	36.3	67.3	82.7	89.7	96.7
B	opt $\boldsymbol{q}$	58.0	80.7	92.0	96.0	96.7
	$\boldsymbol{q} = \mathbf{0}$	57.0	78.7	90.0	95.7	96.7

7. Testing the generalisation of weighted scoring functions

Having corroborated the model presented in section 6.2, in this section we will use it to optimise weighted scoring functions so as to improve the accuracy of the speller in two sets of experiments. Tests will be carried out with independent datasets which will not be used in the setting of any of the parameters of the algorithm.

In sections 6.4 and 6.5, for analysis and visualisation purposes, we restricted the exploration of the parameter space to only two degrees of freedom. However, since the model computes estimates of the accuracy of the speller very efficiently, we can evaluate many tentative parameter settings. Therefore, in this section, for each subject we will allow the exploration of the full 5-dimensional parameter space defined by the parameter J and the four components of $\boldsymbol{q}$.

7.1. Tests with an independent test set from the BCI Competition III

For each subject in dataset II from BCI Competition III and for each value of J, we determined the vector $\boldsymbol{q} = [c_1, c_2, c_3, b]$ which resulted, according to the model, in the best accuracy for the speller. The parameters of the algorithm/model were determined only using the training data from dataset II. Then the optimal weighting function was tested on the test set of the BCI competition. To reiterate, this set was never used during any phase of the design of the system, nor for the training of the classifier or the weighting function. So, results with this test give a true indication of the generalisation capabilities of the approach.

Table 3 and figure 8 show the results of our tests, reporting the accuracy of the speller on the test set for the two subjects when using the optimised weighted scoring function. We also report the results for the case $\boldsymbol{q} = \mathbf{0}$, i.e., when using the standard character scoring method, which corresponds to the original method in [29]. Let us analyse the results.

For very small or very large values of J the improvement provided by a weighted scoring function is small or even slightly negative. For short sequences ($J < 3$), this is probably ascribable to the fact that some of the approximations in the model become less accurate when J is very small (see Appendix A.2). Note that this does not imply that a weighted scoring function would not help with small values of J: it simply means that, for this particular case, the parameters the model suggest for the scoring algorithm are sub-optimal. The weighting has also a small effect for long sequences, where the performance of both scoring techniques reaches 95% ($J \geq 13$ for A and $J \geq 11$ for B) and, so, there is little room for improvement.

However, for intermediate values of J, the new method performs markedly better than the method in [29], i.e., the best performing Donchin speller published to date. More specifically, in the range of J values where the accuracy of the speller is between

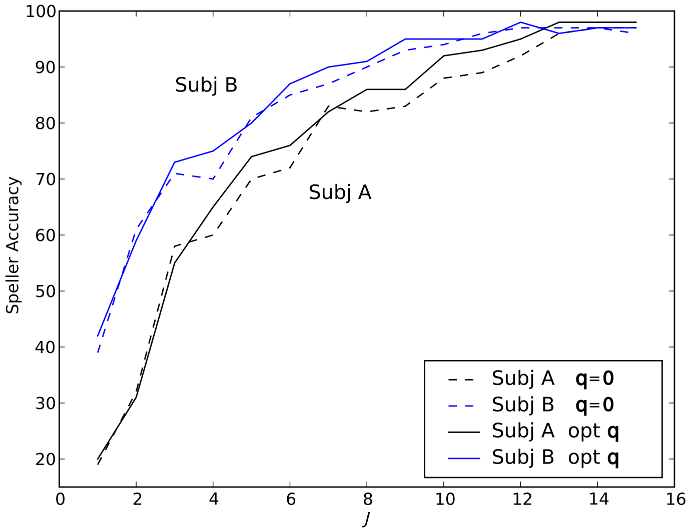


Figure 8. Accuracy (percentage of correctly identified characters) of the speller on a test set for the two subjects in the BCI Competition III dataset when using the optimised weighting function ("opt $\mathbf{q} = 0$") and with the original non-weighted algorithm ("$\mathbf{q} = 0$") by Rakotomamonjy and Guigue [29].

70% and 95%, there is an improvement of around 3% on average. This is particularly important because 70–95% is the accuracy range for which the usability of a speller is highest. A speller with an accuracy below 70% can be frustrating for the user [38] and can prevent recipients from understanding messages. Instead, an accuracy exceeding 95% only marginally improves the understandability of messages while markedly increasing the time needed to communicate, which can be equally frustrating.

7.2. Test with a further twelve independently acquired subjects

In order to confirm the applicability and benefits of the new approach on a larger and independent group of subjects, we tested the algorithm within our BCI lab on 12 further subjects. In this section we report the details and the results of these experiments.

Each participant was asked to spell a total of 20 characters. Target characters were randomly chosen before the beginning of the session. Each row and column of the standard 6×6 matrix of characters was randomly intensified without replacement for 100 ms with a gap of 50 ms, leading to a SOA of 150 ms. During the spelling of a character, each row and column flashed 10 times, for a total of 120 flashes, between targets and non-targets. During that period, subjects were asked to focus on the target character and to mentally count the number of times it was highlighted.

Subjects were seated comfortably with their neck supported by a C-shaped inflatable travel pillow to reduce muscular artefacts. The eyes were at approximately 80 cm from a 22" LCD screen with 60 Hz refresh rate. Data were collected from 64 electrode sites using a BioSemi ActiveTwo EEG system. The EEG channels were referenced to the mean of the electrodes placed on either earlobe. Data were sampled at 2048 Hz, filtered and then downsampled by a factor of 8.

Cross validation was used to assess the accuracy of the new algorithm as well as

Table 4. Comparison of the accuracy (%) of the speller when using the optimized weighted scoring ("opt $\boldsymbol{q}$") w.r.t. the non-weighted version ("$\boldsymbol{q} = \mathbf{0}$") with the 12 subjects tested within our lab.

Subject	Algorithm	J									
		1	2	3	4	5	6	7	8	9	10
1	opt $\boldsymbol{q}$	45	85	100	95	100	100	100	100	100	100
	$\boldsymbol{q} = \mathbf{0}$	50	85	90	95	100	100	100	100	100	100
2	opt $\boldsymbol{q}$	80	100	95	100	100	100	100	100	100	100
	$\boldsymbol{q} = \mathbf{0}$	70	100	95	100	100	100	100	100	100	100
3	opt $\boldsymbol{q}$	40	70	90	95	95	95	100	95	95	95
	$\boldsymbol{q} = \mathbf{0}$	45	65	85	95	90	95	95	95	95	95
4	opt $\boldsymbol{q}$	55	95	95	100	95	100	100	100	100	100
	$\boldsymbol{q} = \mathbf{0}$	60	85	95	100	95	100	100	100	100	100
5	opt $\boldsymbol{q}$	65	60	85	80	90	90	100	100	100	95
	$\boldsymbol{q} = \mathbf{0}$	55	60	85	85	95	90	100	100	100	95
6	opt $\boldsymbol{q}$	70	90	100	100	100	100	100	100	100	100
	$\boldsymbol{q} = \mathbf{0}$	70	80	95	100	100	100	100	100	100	100
7	opt $\boldsymbol{q}$	10	20	40	65	85	85	80	85	90	95
	$\boldsymbol{q} = \mathbf{0}$	15	15	40	60	75	70	80	90	85	85
8	opt $\boldsymbol{q}$	55	90	95	95	100	95	100	100	95	100
	$\boldsymbol{q} = \mathbf{0}$	55	90	95	95	100	95	100	100	95	100
9	opt $\boldsymbol{q}$	65	90	95	100	100	100	100	100	100	100
	$\boldsymbol{q} = \mathbf{0}$	70	80	95	100	100	100	100	100	100	100
10	opt $\boldsymbol{q}$	35	65	90	90	95	95	95	95	100	100
	$\boldsymbol{q} = \mathbf{0}$	35	60	90	90	95	95	95	95	100	100
11	opt $\boldsymbol{q}$	45	65	80	90	95	95	100	100	100	100
	$\boldsymbol{q} = \mathbf{0}$	45	65	75	90	95	95	100	100	100	100
12	opt $\boldsymbol{q}$	35	75	90	95	95	95	100	100	100	100
	$\boldsymbol{q} = \mathbf{0}$	45	80	85	95	95	95	95	100	100	100

of the standard, non-weighted version of the speller. More specifically, the dataset of each subject was split into 10 subsets, each including the data related to two target characters. One dataset was then used as a test set while the remaining nine formed the training set. Using the software in [39], one ensemble classifier was built by training one SVM on each of the 9 subsets of the training set. Then, using the same leave-one out approach as in section 5.1, we found the parameters α_{th}, α_{nt}, and σ for the model in (3) and (14). Given these parameters, we used the analytical model in section 6.2, to find the optimal vector of parameters $\boldsymbol{q}$. We then used the ensemble classifier and the weighted scoring function associated to $\boldsymbol{q}$ to classify the test subset. For reference, we did the same with the non-weighted scoring function. This procedure was repeated using, in turn, each one of the 10 subsets as test set and the remaining nine as training set. The results were then combined so as to obtain an average accuracy measure across the 20 characters acquired with each subject.

Table 4 reports the results of this procedure for the 12 subjects and for 10 different values of the number of repetitions, J. In total there are 120 accuracy figures in the table. These are multiples of 5% because each subject was tested on 20 characters which tends to make small performance differences difficult to resolve. Nonetheless, our new algorithm performs better than the standard speller in 21 cases and when it does so the accuracy improvement is approximately 7.4%, i.e., between one and two characters more out of 20. Conversely, only in 10 cases the standard speller beats our algorithm and when it does so the performance difference is 5.5%, corresponding to approximately one character. Additionally, 6 out of these 10 cases are for $J = 1$, where we already know from the previous section that our model becomes less reliable and so our algorithm is likely to be sub-optimally parametrised. So, overall results are quite encouraging.

Table 5. Accuracy of the speller for the two algorithms choosing the minimum number of repetitions for which either one reaches an accuracy of 80%.

		Subject											
		1	2	3	4	5	6	7	8	9	10	11	12
J such that acc $\geq$80%		2	1	3	2	3	2	5	2	2	3	3	3
algorithm	opt $\boldsymbol{q}$	85	80	90	95	85	90	85	90	90	90	80	90
	$\boldsymbol{q} = \boldsymbol{0}$	85	70	85	85	85	80	75	90	80	90	75	85

As the accuracy figures for different values of J are strongly correlated, we could not run tests to verify the statistical significance of the data in table 4 using more than one value of J per subject. However, it seemed inappropriate to choose a particular value of J, since different algorithms may work best for different J's and also the optimal value of J (in terms of usability — more on this below) appears to be subject dependent. Therefore, for statistical testing we decided to take, for each subject, the value of J for which either method reached at least an accuracy of 80% which provides a reasonable trade-off between accuracy and speed (see discussion at the end of section 7.1). Table 5 reports the value J chosen with this method and the corresponding accuracy of the two algorithms, for each subject. Note that the data in this table are a subset of the data in table 4.

We ran a paired t-test on the data from table 5. The test confirmed that the accuracy improvement of the weighted scoring algorithm w.r.t. the non-weighted one is statistically significant (p=0.014, t=2.93) with a 95%-confidence interval for the improvement of 1.14–8.03%. In other words, we obtained significant improvements in accuracy with our own subjects as well. They are compatible with the values estimated by our probabilistic model, the Monte Carlo simulations and with the values obtained on the BCI competition test set.

All this evidence points clearly in one direction: significant benefits can be accrued by considering and exploiting the modulations in P300s induced by stimulus history.

8. Discussion

Amplitude and shape variations in brain waves are often considered harmful for BCIs. However, as we suggested in [3, 4], in some cases they carry information which can be exploited to improve performance.

Our system embraces this philosophy, by considering, modelling and exploiting TTI-related P300 modulations. This is done at two levels. In system's second stage (the character scorer), we use different weighting factors for the responses of the first stage to flashes. The weights are based on flash latencies with respect to some earlier reference event, thereby making it possible for the system to adapt to or exploit history effects on brain waves. We also use a model which takes TTI-related P300 modulations into account to optimally set the weights and biases of such a system from a training set of data. The model explicitly represents the modulations in the flash scores produced by the first stage in the presence of TTI variations, translates them into corresponding character scores and integrates the results into a formula which gives the probability of correct classification as a function of the system's parameters.

The good results obtained with this approach suggest that the modulations in P300 related to differences in TTI are not necessarily a problem within the Donchin speller paradigm, provided one takes such variations into account when designing the system, as we did, instead of averaging over them.

Naturally, this is not the only way of improving performance in a Donchin speller. Another alternative is, for example, to modify the protocol in such a way to reduce TTI-related modulations in ERPs. For instance, [40] suggested that performance improvements could be obtained by using stimuli where the letters being flashed were not all in the same row or column. Groups of flashing letters were chosen in such a way as to maximise the distance between the representation of the outputs of the first stage (over multiple repetitions of a complete sequence of flashes) for different characters on the assumption that the first-stage classifier was 100% correct. This particular choice provides the system with error correction capabilities which may help in the recognition of characters. The benefits of this idea were particularly clear when using a new type of stimuli where each letter of the matrix was placed in a grey rectangle with either horizontal or vertical orientation and letters were activated by switching the orientation of the rectangle between these two states. It was suggested that these stimuli may generate psycho-physiological responses which are less susceptible to TTI modulations than the original flashes. It would be interesting to combine such stimuli with our method to see if the two techniques can work synergistically to further improve performance.

An important question regarding performance is whether this is assessed with online or offline data and analyses. It is clear that the ultimate goal should be to evaluate performance by testing a system online, i.e., in exactly the same conditions experienced by the intended users of the system. However, after weighing the benefits and the obstacles associated with testing our system online we decided to postpone an online comparison to a follow-up paper and perform an offline analysis at this stage. The main reason is that with an offline analysis we can perform a comparison of the two algorithms on the exact same datasets, which in turn allows the use of a paired t-test to establish the statistical significance or otherwise of performance differences. This test has a greater statistical power than an unpaired test, thereby making it possible to make statistically sound inferences with much fewer subjects. Testing the two algorithms with online systems would have added a noise factor due to the two algorithms classifying different EEG signals and would have prevented us from using of a paired test.

Recent work [41, 42] has shown that the performance of P300-based spellers is significantly affected by whether users gaze at target letters in addition to focusing their attention on them (overt attention) or gaze at a fixation element of the screen and use covert attention to focus on target letters. Performance is significantly higher in the overt attention case. This is most likely because, in addition to the P300 and other endogenous components being modulated, also visual exogenous ERPs are modulated by overt attention (contrarily to what happens with covert attention). As a result, in the case of overt attention classifiers can exploit extra information when making decisions. As suggested in [42], this implies that resulting BCIs are not truly independent and, thus, future work should attempt to create systems which decouple attention from gaze. In such systems it will be even more important to use strategies such as the methods suggested in this paper to make the best use of the information available in endogenous ERPs, including their modulations due to stimulus history.

Finally, we would like to mention, that in recent work BCIs that rely on P300s evoked through different sensory modalities such as spacial auditory [43] or tactile [44] have been proposed. There is no reason to believe the techniques presented in the paper to best exploit visual ERPs could not be applied also to ERPs generated using stimuli presented in a different modality.

9. Conclusions

In this paper we first documented, modelled and then exploited, within the context of the Donchin speller, a modulation in the amplitude of P300s associated with variations in the number of non-targets preceding a target. In particular, we specialised the system through the use of a new character-scoring function which, with minimal effort, allows it take such modulations of the P300 account.

We mathematically assessed the potential of the approach by means of a statistical model based on few reasonable assumptions, which we later verified. Validation by Monte Carlo simulations showed that the model is accurate.

Testing the new approach on unseen data revealed an average improvement in accuracy of between 2 and 4% over the best classification algorithm for Donchin speller in the literature [29] at the range of sequence lengths which is most suited for practical use of a speller.

The model and method we developed are quite general and can be applied to a wide spectrum of ERP-based BCIs, including the Donchin speller with different matrix sizes, a sequential speller, a generic multi-choice selection BCI, and many others. In future research we intend to explore the possibility of obtaining similar improvements within other BCI paradigms based on P300s, including our own BCI Mouse [4].

Acknowledgments

The authors would like to thank Mathew Salvaris for his valuable comments and help. We would also like to thank Franciso Sepulveda for his help in revising the manuscript.

This work was supported by the Engineering and Physical Sciences Research Council (UK) under grant EP/F033818/1 “Analogue Evolutionary Brain Computer Interfaces”.

Appendix A. Modelling the accuracy of the speller based on weighted scoring

In our model of the first stage of Donchin speller in the presence of history effects, we assumed that the distribution of flash scores follows (3) for not targets (see section 4.2.1) and follows (14) for targets (see section 5.1), respectively. In this appendix we will formally derive a model of the second stage so as to assess the accuracy of the speller when making decisions after a sequence of J stimulus repetitions using the scoring function in (16).

Appendix A.1. Scoring characters

Let us start by considering a single repetition of 12 flashes — one for each row and column — within a longer sequence of repetitions (we will extend the treatment to the multi-repetition case later). Then (16) can be rewritten in statistical terms as

$$\begin{aligned} S_{r,c} = {} & w(\mathrm{h}(\rho, \mathcal{R}_r \cup \mathcal{C}_c))\,(Y_\rho - b) \\ & + w(\mathrm{h}(\chi, \mathcal{R}_r \cup \mathcal{C}_c))\,(Y_\chi - b) \end{aligned} \tag{A.1}$$

where ρ and χ satisfy $L_\rho = \mathsf{R}_r$ and $L_\chi = \mathsf{C}_c$, respectively. In (A.1) the randomness is due to L (through ρ, χ, $\mathcal{R}_r$ and $\mathcal{C}_c$) and to the random variables Y_i, for $i \in \{\rho, \chi\}$.

If the target character is at position $(\hat{r}, \hat{c})$, according to (3) and (14) the variables Y_i can be expressed as:

$$Y_i = \begin{cases} \alpha_{\mathrm{nt}} + \mathring{Y} & \text{if } i \notin \mathcal{R}_{\hat{r}} \cup \mathcal{C}_{\hat{c}} \ , \\ \alpha_{\mathrm{t}\ \mathrm{h}(\mathrm{i}, \mathcal{R}_{\hat{\mathrm{r}}} \cup \mathcal{C}_{\hat{\mathrm{c}}})} + \mathring{Y} & \text{otherwise} \ , \end{cases} \tag{A.2}$$

where the function h was defined at the beginning of section 6.1. Note that the variable $\mathring{Y}$ is independently instantiated for each i.

To avoid confusion, in the following, every time two independently instantiated variables appear in the same expression, we will use different Roman superscripts to highlight the fact that they are variables with the same distribution, but not the same random variable.

Using (A.2), (A.1) becomes

$$\begin{aligned} S_{r,c} = {} & w(\mathrm{h}(\rho, \mathcal{R}_r \cup \mathcal{C}_c)) \left\{ [r = \hat{r}]\, \alpha^b_{\mathrm{t}\ \mathrm{h}(\rho, \mathcal{R}_{\hat{\mathrm{r}}} \cup \mathcal{C}_{\hat{\mathrm{c}}})} + [r \neq \hat{r}]\, \alpha^b_{\mathrm{nt}} + \mathring{Y}^I \right\} \\ & + w(\mathrm{h}(\chi, \mathcal{R}_r \cup \mathcal{C}_c)) \left\{ [c = \hat{c}]\, \alpha^b_{\mathrm{t}\ \mathrm{h}(\chi, \mathcal{R}_{\hat{\mathrm{r}}} \cup \mathcal{C}_{\hat{\mathrm{c}}})} + [c \neq \hat{c}]\, \alpha^b_{\mathrm{nt}} + \mathring{Y}^{II} \right\} \ , \end{aligned} \tag{A.3}$$

where $[\dots]$ are Iverson brackets (i.e., they return 1 if the condition in brackets is satisfied, and 0 otherwise), and we used the definitions $\alpha^b_{t\ \mathrm{h}(.)} \overset{\mathrm{def}}{=} \alpha_{t\ \mathrm{h}(.)} - b$ and $\alpha^b_{\mathrm{nt}} \overset{\mathrm{def}}{=} \alpha_{\mathrm{nt}} - b$ for conciseness.

Because of the variability of ρ, χ, $\mathcal{R}_r$, $\mathcal{C}_c$, $\mathcal{R}_{\hat{r}}$ and $\mathcal{C}_{\hat{c}}$, the functions $\mathrm{h}(\rho, \mathcal{R}_r \cup \mathcal{C}_c))$ and $\mathrm{h}(\chi, \mathcal{R}_r \cup \mathcal{C}_c))$ are stochastic. For notional convenience we represent them with a new stochastic variable, H, which we need to characterise. To simplify our treatment, we make use of the assumption that the number of flashes occurring between the flashing of two particular rows or columns of interest is independently and identically distributed across epochs. This is a first order approximation because it ignores the constraint that each row and column flash exactly once within each group of 12 flashes. Under this assumption, however, the H variables associated with different ρ, χ, r, c become i.i.d.

Using H, (A.3) simplifies. However it takes different forms depending on whether we are scoring a target character, a "half-target" (i.e., a character belonging to either the same row or the same column of the target, but not both) or a non-target. Namely:

$$S_{\hat{r},\hat{c}} = \mathrm{w}(H^I)\ \left[\alpha^b_{\mathrm{t}\ \mathrm{H^I}} + \mathring{Y}^I\right] + \mathrm{w}(H^{II})\ \left[\alpha^b_{\mathrm{t}\ \mathrm{H^{II}}} + \mathring{Y}^{II}\right] \qquad \text{(target)} \tag{A.4}$$

$$S_{\hat{r},*} = \mathrm{w}(H^{III,*})\ \left[\alpha^b_{\mathrm{t}\ \mathrm{H^I}} + \mathring{Y}^I\right] + \mathrm{w}(H^{IV,*})\ \left[\alpha^b_{\mathrm{nt}} + \mathring{Y}^{III,*}\right] \qquad \text{(row half-target)} \tag{A.5}$$

$$S_{*,\hat{c}} = \mathrm{w}(H^{V,*})\ \left[\alpha^b_{\mathrm{nt}} + \mathring{Y}^{IV,*}\right] + \mathrm{w}(H^{VI,*})\ \left[\alpha^b_{\mathrm{t}\ \mathrm{H^{II}}} + \mathring{Y}^{II}\right] \qquad \text{(col. half-target)} \tag{A.6}$$

$$S_{*,*} = \mathrm{w}(H^{VII,*})\ \left[\alpha^b_{\mathrm{nt}} + \mathring{Y}^{V,*}\right] + \mathrm{w}(H^{VIII,*})\ \left[\alpha^b_{\mathrm{nt}} + \mathring{Y}^{VI,*}\right] \qquad \text{(non-target)} \tag{A.7}$$

where the star symbol stands for any non target line, i.e., any $r \neq \hat{r}$ if used as a row index, or any $c \neq \hat{c}$ if used as a column index.‡

‡ Note that the variables resulting from the replacement of a star with a row or column index are different variables with the same distribution. Also, note that during online use of a classifier the true target is unknown. In our approach, when scoring a generic character (r, c), h is computed under the assumption that r and c were targets. However, only if (r, c) is the true target character $(\hat{r}, \hat{c})$, the variables controlling the weight, H^I and H^{II}, match the ones determining the average amplitude; in all other cases the flash score is weighted with a "wrong" h.

The target character $(\hat{r}, \hat{c})$ will be correctly identified if and only if the stochastic variable $S_{\hat{r},\hat{c}}$ takes a value that is greater than the maximum taken by the variables in the set $\{S_{\hat{r},*}\} \cup \{S_{*,\hat{c}}\} \cup \{S_{*,*}\}$. We want to compute the probability of this happening.

This task is quite difficult. In order to simplify the treatment we will make an approximation which leads to slightly over-estimating the accuracy of the speller. Namely, we will assume that misclassifications can only occur because the score of one of the 10 most likely competitors, the "half-targets", is higher than the value taken by $S_{\hat{r},\hat{c}}$. In other words, we neglect the possibility of one of the 25 non-targets (corresponding to (A.7)) causing a misclassification.§ We will therefore drop (A.7) hereafter.

Equations (A.4)–(A.6) can be rewritten as

$$S_{\hat{r},\hat{c}} = U'_{\hat{r}} + V'_{\hat{c}} \ , \tag{A.8}$$

$$S_{\hat{r},*} = \frac{\mathrm{w}(H^{III,*})}{\mathrm{w}(H^{I})} \, U'_{\hat{r}} + V'_{*} \ , \tag{A.9}$$

$$S_{*,\hat{c}} = U'_{*} + \frac{\mathrm{w}(H^{VI,*})}{\mathrm{w}(H^{II})} \, V'_{\hat{c}} \ , \tag{A.10}$$

where

$$U'_{\hat{r}} = \mathrm{w}(H^{I}) \, \left[\alpha^{b}_{\mathrm{t\ H^{I}}} + \mathring{Y}^{I}\right] \ , \tag{A.11}$$

$$U'_{*} = \mathrm{w}(H^{V,*}) \, \left[\alpha^{b}_{\mathrm{nt}} + \mathring{Y}^{IV,*}\right] \ , \tag{A.12}$$

$$V'_{\hat{c}} = \mathrm{w}(H^{II}) \, \left[\alpha^{b}_{\mathrm{t\ H^{II}}} + \mathring{Y}^{II}\right] \ , \tag{A.13}$$

$$V'_{*} = \mathrm{w}(H^{IV,*}) \, \left[\alpha^{b}_{\mathrm{nt}} + \mathring{Y}^{III,*}\right] \tag{A.14}$$

are stochastic variables.

When a sequence of J repetitions is used to score characters and make decisions, the average, $\bar{S}_{\hat{r},\hat{c}}$, of J instantiations of $S_{\hat{r},\hat{c}}$ has to be compared with the corresponding half-target averages $\bar{S}_{\hat{r},*}$ and $\bar{S}_{*,\hat{c}}$. These can be expressed by trivially generalising (A.8)–(A.10) as follows:

$$\bar{S}_{\hat{r},\hat{c}} = \bar{U}'_{\hat{r}} + \bar{V}'_{\hat{c}} \ , \tag{A.15}$$

$$\bar{S}_{\hat{r},*} = \overline{\frac{\mathrm{w}(H^{III,*})}{\mathrm{w}(H^{I})} \, U'_{\hat{r}}} + \bar{V}'_{*} \ , \tag{A.16}$$

$$\bar{S}_{*,\hat{c}} = \bar{U}'_{*} + \overline{\frac{\mathrm{w}(H^{VI,*})}{\mathrm{w}(H^{II})} \, V'_{\hat{c}}} \ , \tag{A.17}$$

where a horizontal bar over a variable or an expression represents a new stochastic variable which is the average of J independent instances of that variable or expression.

The purpose of $\mathrm{w}(h)$ in our scoring equation is to slightly modulate the effect different values of Y have on the average, according to their h. So we will choose weights that are distributed around 1. Under this restriction, it is safe to assume that

§ This simplified model would be mistaken only if there is a non-target whose score is higher than the target's, which in turn is higher than all the "half-targets". However, this is *impossible* if $\mathrm{w}(h) = 1 \ \forall h$. So, considering only the half-targets introduces no approximation in the calculations of the speller's accuracy when applied to the traditional non-weighted scoring scheme in (1). Also, the likelihood of non-targets having scores larger than $S_{\hat{r},\hat{c}}$ is very low for values of $\mathrm{w}(h)$ reasonably close to 1 (we will show later that this is indeed the case for our experiment).

the value $r_{\max}$ such that $\bar{U}'_{r_{\max}} = \bar{U}'_{\max} = \max\{\bar{U}'_*\}$ also satisfies $\bar{S}_{r_{\max},\hat{c}} = \max\{\bar{S}_{*,\hat{c}}\}$. In other words, in (A.17) we allow the second term to partly contribute to the value of the score but we assume that its value is never so big to change which one of the non-target rows has the maximum score.

Under this assumption

$$\bar{S}_{\hat{r},\hat{c}} > \max\{\bar{S}_{*,\hat{c}}\} \quad \Longleftrightarrow \quad \bar{U}'_{\hat{r}} + \overline{\Delta U} > \bar{U}'_{\max} \tag{A.18}$$

where

$$\Delta U = V'_{\hat{c}} \left(1 - \frac{\mathrm{w}(H^{VI,\, r_{\max}})}{\mathrm{w}(H^{II})}\right) = \left[\mathrm{w}(H^{II}) - \mathrm{w}(H^{VI,\, r_{\max}})\right] \left[\alpha^b_{\mathrm{t\ H^{II}}} + \mathring{Y}^{II}\right] .$$

Similar expressions can be obtained for the columns and the V variables.

Combining these results we obtain:

$$\bar{S}_{\hat{r},\hat{c}} > \max\left\{ \{\bar{S}_{\hat{r},*}\} \cup \{\bar{S}_{*,\hat{c}}\} \right\} \quad \Longleftrightarrow \quad \bar{U}''_{\hat{r}} > \bar{U}''_{\max} \quad \wedge \quad \bar{V}''_{\hat{c}} > \bar{V}''_{\max} \tag{A.19}$$

where $\bar{U}''_{\hat{r}} = \bar{U}'_{\hat{r}} + \overline{\Delta U}$, while $\bar{U}''_{\max} = \bar{U}'_{\max}$ (and likewise for the V variables).

Appendix A.2. Probability distributions

We now need to find the distributions of $\bar{U}''_{\hat{r}}$, $\bar{U}''_{\max}$, $\bar{V}''_{\hat{r}}$, and $\bar{V}''_{\max}$. In the following we will concentrate on the U variables but results also apply to the V variables because their distributions are the same.

The variables $\bar{U}''_{\hat{r}}$ and $\bar{U}''_{\max}$ depend on $\bar{U}'_{\hat{r}}$, $\bar{U}'_*$, and $\overline{\Delta U}$. These, in turn, depend on $\bar{U}'_*$ and $\bar{U}'_{\hat{r}}$, which are averages of U'_* and $U'_{\hat{r}}$ variables. So, let us start by finding the distributions of U'_* and $U'_{\hat{r}}$.

Using the theorem in Appendix B, we can express the cumulative distribution functions (cdf) of U'_* and $U'_{\hat{r}}$ as:

$$\mathrm{F}_{U'_{\hat{r}}}(x) = \sum_h p_h \,\mathrm{normcdf}\left(\frac{x - \mathrm{w}(h)\,\alpha^b_{\mathrm{t}\ h}}{\mathrm{w}(h)\,\sigma}\right) \tag{A.20}$$

$$\mathrm{F}_{U'_*}(x) = \sum_h p_h \,\mathrm{normcdf}\left(\frac{x - \mathrm{w}(h)\,\alpha^b_{\mathrm{nt}}}{\mathrm{w}(h)\,\sigma}\right) . \tag{A.21}$$

The corresponding probability density functions (pdf) are:

$$\mathrm{f}_{U'_{\hat{r}}}(x) = \sum_h p_h \frac{1}{\mathrm{w}(h)\,\sigma} \,\mathrm{normpdf}\left(\frac{x - \mathrm{w}(h)\,\alpha^b_{\mathrm{t}\ h}}{\mathrm{w}(h)\,\sigma}\right) \tag{A.22}$$

$$\mathrm{f}_{U'_*}(x) = \sum_h p_h \,\frac{1}{\mathrm{w}(h)\,\sigma} \,\mathrm{normpdf}\left(\frac{x - \mathrm{w}(h)\,\alpha^b_{\mathrm{nt}}}{\mathrm{w}(h)\,\sigma}\right) . \tag{A.23}$$

Using the theorem in Appendix C, from (A.20) and (A.21) we obtain:

$$\begin{aligned}
\mu_{U'_{\hat{r}}} &= \sum_h p_h \mathrm{w}(h)\,\alpha^b_{\mathrm{t}\ h} \\
\sigma^2_{U'_{\hat{r}}} &= \sum_h p_h (\mathrm{w}(h)^2\sigma^2 + (\mathrm{w}(h)\alpha^b_{\mathrm{t}\ h} - \mu_{U'_{\hat{r}}})^2) \\
\mu_{U'_*} &= \sum_h p_h \mathrm{w}(h)\,\alpha^b_{\mathrm{nt}} \\
\sigma^2_{U'_*} &= \sum_h p_h \left(\mathrm{w}(h)^2\sigma^2 + (\mathrm{w}(h)\alpha^b_{\mathrm{nt}} - \mu_{U'_*})^2\right) .
\end{aligned}$$

We now want to find the distributions of the averages of J instances of U'_* and $U'_{\hat{r}}$, namely $\bar{U}'_*$ and $\bar{U}'_{\hat{r}}$, respectively. We will do so by using the central limit theorem.

According to the central limit theorem, the distribution of the average of a sufficiently large number of i.i.d. random variables can be considered as normal and its mean and variance can be estimated from the mean and variance of the distribution of the original variables. In our case the number of terms averaged ranges from a few to up to 15. These numbers may look too small for the application of the central limit theorem. However, results on the rate of convergence of the central limit theorem based on pseudo-moments (e.g., [45]) indicate that a faster convergence can be obtained when the original distribution is close to a normal distribution, which is the case in our model.

From (A.22) and (A.23), it follows that the probability density functions of $U'_{\hat{r}}$ and U'_* are sums of scaled and shifted Gaussians. If the differences among the values of $\mathrm{w}(h)$ as well as the differences among the values of $\alpha^b_{\mathrm{t}\,h}$ are small compared to σ and the values of $\mathrm{w}(h)$ and $\alpha^b_{\mathrm{t}\,h}$ are of the same order of magnitude as σ, the Gaussians in (A.22) and (A.23) are close and merge into a bell-like distribution which closely resembles a Gaussian. Hence, the central limit theorem gives reasonably accurate approximations even for $J \leq 15$.

Using the central limit theorem, $\bar{U}'_{\hat{r}}$ and $\bar{U}'_*$ are normally distributed with parameters: $\mu_{\bar{U}'_{\hat{r}}} = \mu_{U'_{\hat{r}}}$, $\sigma^2_{\bar{U}'_{\hat{r}}} = \sigma^2_{U'_{\hat{r}}}/J$, $\mu_{\bar{U}'_*} = \mu_{U'_*}$, $\sigma^2_{\bar{U}'_*} = \sigma^2_{U'_*}/J$.

Through a procedure similar to that in Appendix B, we find that the variable ΔU has the following cdf:

$$\mathrm{F}_{\Delta U}(x) = \sum_{h_1}\sum_{h_2} p_{h_1}\, p_{h_2} \;\mathrm{normcdf}\left(\frac{x - [\mathrm{w}(h_1) - \mathrm{w}(h_2)]\;\alpha^b_{\mathrm{t\,h_1}}}{[\mathrm{w}(h_1) - \mathrm{w}(h_2)]\;\sigma}\right) .$$

Its mean and variance are

$$\mu_{\Delta U} = \sum_{h_1}\sum_{h_2} p_{h_1}\, p_{h_2}\; [\mathrm{w}(h_1) - \mathrm{w}(h_2)]\;\alpha^b_{\mathrm{t\,h_1}}$$

$$\sigma^2_{\Delta U} = \sum_{h_1}\sum_{h_2} p_{h_1}\, p_{h_2}\; \Big\{ [\mathrm{w}(h_1) - \mathrm{w}(h_2)]^2\,\sigma^2 + \big\{[\mathrm{w}(h_1) - \mathrm{w}(h_2)]\;\alpha^b_{\mathrm{t\,h_1}} - \mu_{\Delta U}\big\}^2 \Big\} .$$

The considerations made earlier about the central limit theorem apply also to ΔU. Thus, we find that $\overline{\Delta U}$ is approximately Gaussian with parameters $\mu_{\overline{\Delta U}} = \mu_{\Delta U}$ and $\sigma^2_{\overline{\Delta U}} = \sigma^2_{\Delta U}/J$.

As $\bar{U}'_{\hat{r}}$ and $\overline{\Delta U}$ are independent and normally distributed, then also $\bar{U}''_{\hat{r}} = \bar{U}'_{\hat{r}} + \overline{\Delta U}$ is normally distributed. Its parameters are: $\mu_{\bar{U}''_{\hat{r}}} = \mu_{\bar{U}'_{\hat{r}}} + \mu_{\overline{\Delta U}}$ and $\sigma^2_{\bar{U}''_{\hat{r}}} = \sigma^2_{\bar{U}'_{\hat{r}}} + \sigma^2_{\overline{\Delta U}}$.

For the non targets, the distribution of $\bar{U}''_{\max} = \bar{U}'_{\max} = \max\{\bar{U}'_*\}$ can be found using the same procedure as in section 4.2 obtaining $\mathrm{F}_{\bar{U}''_{\max}}(x) = \left[\mathrm{F}_{\bar{U}'_*}(x)\right]^5$.

Appendix A.3. Model of Donchin speller accuracy in the presence of h-related effects

We can find the probability that $\bar{U}''_{\hat{r}} > \bar{U}''_{\max}$ using the same procedure as in section 4.2 obtaining the equivalent of (11).

The very last step is to derive the probability of the conjunction on the right in (A.19): $\Pr\left(\bar{U}''_{\hat{r}} > \bar{U}''_{\max} \wedge \bar{V}''_{\hat{c}} > \bar{V}''_{\max}\right)$. Even if the statement (A.19) is formally identical to (9) with $\bar{U}''_{\hat{r}}$ replacing $\bar{U}_{\hat{r}}$ and $\bar{U}''_{\hat{r}}$ replacing $\bar{U}_{\hat{r}}$, we cannot directly write the equivalent of (10) because $\bar{U}''_{\hat{r}}$ and $\bar{V}''_{\hat{c}}$ are not independent. In fact, they both depend on both $\bar{U}'_{\hat{r}}$ and $\bar{V}'_{\hat{c}}$. Their dependence is through ΔU and ΔV whose means

$\mu_{\Delta U}$ and $\mu_{\Delta V}$ (as shown in Appendix A.2) contain the factor $[\mathrm{w}(h_1) - \mathrm{w}(h_2)]\ \alpha^{b}_{\mathrm{t\ h_1}}$. However, if we restrict the search space to values of $\mathrm{w}(h)$ and b (which affects $\alpha^{b}_{\mathrm{t\ h}}$) such that $\mathrm{w}(h)$ is never too different from 1 and $\alpha^{b}_{\mathrm{t\ h}}$ is not excessively large, we can assume the independence of $\bar{U}''_{\hat{r}}$ and $\bar{V}''_{\hat{c}}$.

Under this assumption we can approximate $\Pr\left(\bar{U}''_{\hat{r}} > \bar{U}''_{\max} \wedge \bar{V}''_{\hat{c}} > \bar{V}''_{\max}\right)$ as $\Pr\left(\bar{U}''_{\hat{r}} > \bar{U}''_{\max}\right)^2$ obtaining:

$$\begin{aligned} A = \quad & \Pr\left(\bar{S}_{\hat{r},\hat{c}} > \max\left\{\, \{\bar{S}_{\hat{r},*}\} \cup \{\bar{S}_{*,\hat{c}}\}\, \right\}\right) = \Pr\left(\bar{U}''_{\hat{r}} > \bar{U}''_{\max} \wedge \bar{V}''_{\hat{c}} > \bar{V}''_{\max}\right) = \\ & \left\{ \int_{\mathbb{R}} \frac{1}{\sigma_{\bar{U}''_{\hat{r}}}} \mathrm{normpdf}\left(\frac{x - \mu_{\bar{U}''_{\hat{r}}}}{\sigma_{\bar{U}''_{\hat{r}}}}\right) \left[\mathrm{normcdf}\left(\frac{x - \mu_{\bar{U}''_{*}}}{\sigma_{\bar{U}''_{*}}}\right)\right]^5 \mathrm{d}x \right\}^2 . \end{aligned} \tag{A.24}$$

This equation models the accuracy of the Donchin speller in the presence of h-related effects, when (16) is used to score the characters. It is a special case of (13) with $\beta = \frac{\sigma_{\bar{U}''_{\hat{r}}}}{\sigma_{\bar{U}''_{*}}}$, $\gamma = \frac{\mu_{\bar{U}''_{\hat{r}}} - \mu_{\bar{U}''_{*}}}{\sigma_{\bar{U}''_{*}}}$, $\eta = 5$ and $\lambda = 2$ (case (f) in table 1).

Appendix B. Probability distribution of a function of a discrete and a continuous random variable

Theorem. *Let us consider two independent random variables, H and Y, and the function $\phi(h, y)$. Let H be a discrete variable with values in $\mathcal{H} = \{h_1, \ldots, h_N\}$ and probability masses $p_i = \Pr(H = h_i)$. Let Y be a continuous variable with cumulative distribution function $\mathrm{F}_Y(y)$. If $\phi_i(y) = \phi(h_i, y)$ is strictly increasing in y for all $i \in \{1, \ldots, N\}$, then the cumulative distribution function of $U = \phi(H, Y)$ is $\mathrm{F}_U(u) = \sum_i p_i\, \mathrm{F}_Y(\phi_i^{-1}(u))$.*

Proof. Given a generic proposition $\mathcal{P}$ which depends on the variables H and Y, the probability of $\mathcal{P}$ being true is given by

$$\Pr(\mathcal{P}) = \iint_{\Gamma} \mathrm{f}_{H,Y}(h, y)\, dh\, dy\ ,$$

where $\mathrm{f}_{H,Y}(h, y)$ is the joint probability density function of H and Y and Γ is the region where $\mathcal{P}$ holds.

Therefore, $\mathrm{F}_U(u) = \Pr(U \leq u)$ can be expressed as

$$\mathrm{F}_U(u) = \Pr(\phi(H, Y) \leq u) = \iint_{\Gamma} \mathrm{f}_{H,Y}(h, y)\, dh\, dy\ , \tag{B.1}$$

where Γ is the region of the h-y plane where $\phi(h, y) \leq u$.

The independence of H and Y, and the fact that H is discrete, imply that

$$\mathrm{f}_{H,Y}(h, y) = \mathrm{f}_Y(y) \sum_i p_i\, \delta(h_i - h)\ ,$$

where $\delta(x)$ is the Dirac delta function and $\mathrm{f}_Y(y)$ is the probability density function of Y.

Using Iverson brackets (see Appendix A.1) we can rewrite (B.1) as

$$\begin{aligned}\mathrm{F}_U(u) &= \int_{\mathbb{R}}\int_{\mathbb{R}} [\phi(h,y) \le u] \sum_i p_i\, \delta(h_i - h)\, \mathrm{f}_Y(y)\, dx\, dy \\ &= \sum_i p_i \int_{\mathbb{R}} \delta(h_i - h) \int_{\mathbb{R}} [\phi(h,y) \le u]\, \mathrm{f}_Y(y)\, dy\, dh\ .\end{aligned} \tag{B.2}$$

The outer integral performs the convolution between the Dirac function and the inner integral. As

$$\int_{\mathbb{R}} \delta(t-\tau)\, \psi(\tau)\, d\tau = \psi(t)\ ,$$

(B.2) is equivalent to

$$\mathrm{F}_U(u) = \sum_i p_i \int_{\mathbb{R}} [\phi(h_i,y) \le u]\, \mathrm{f}_Y(y)\, dy\ . \tag{B.3}$$

Each h_i corresponds to a function $\phi_i(y) = \phi(h_i, y)$. As $\phi_i(y)$ is strictly increasing, its inverse exists and is increasing, too. Therefore,

$$\phi(h_i, y) \le u \quad \iff \quad y \le \phi_i^{-1}(u)\ .$$

Then, (B.3) becomes

$$\mathrm{F}_U(u) = \sum_i p_i \int_{\mathbb{R}} [y \le \phi_i^{-1}(u)]\, \mathrm{f}_Y(y)\, dy = \sum_i p_i\, \mathrm{F}_Y(\phi_i^{-1}(u)),$$

which proves the theorem. □

Appendix C. Mean and variance of a variable whose probability density function (pdf) is the weighted sum of several pdfs

Theorem. *Let us consider a random variable, X, with probability density function*

$$\mathrm{f}(x) = \sum_i p_i\, \mathrm{f}_i(x)\ .$$

Let μ_i be the mean and σ_i^2 the variance of a variable whose pdf *is* $\mathrm{f}_i(x)$*. Then, the mean μ and the variance σ^2 of X are:*

$$\mu = \sum_i p_i\, \mu_i \tag{C.1}$$

$$\sigma^2 = \sum_i p_i\, [\sigma_i^2 + (\mu_i - \mu)^2]\, . \tag{C.2}$$

Proof. From the definition of mean it follows that

$$\mu = \int_{\mathbb{R}} x\, \mathrm{f}(x)\, dx = \int_{\mathbb{R}} x \sum_i p_i\, \mathrm{f}_i(x)\, dx = \sum_i p_i \int_{\mathbb{R}} x \mathrm{f}_i(x)\, dx = \sum_i p_i\, \mu_i\ ,$$

which proves (C.1).

From the definition of variance, we have

$$\begin{aligned}
\sigma^2 &= \int_{\mathbb{R}} (x-\mu)^2\, \mathrm{f}(x)\, dx = \int_{\mathbb{R}} (x-\mu)^2 \sum_i p_i\, \mathrm{f}_i(x)\, dx \\
&= \sum_i p_i \int_{\mathbb{R}} (x-\mu)^2\, \mathrm{f}_i(x)\, dx = \sum_i p_i \int_{\mathbb{R}} (x-\mu_i+\mu_i-\mu)^2\, \mathrm{f}_i(x)\, dx \\
&= \sum_i p_i \left[\int_{\mathbb{R}} (x-\mu_i)^2\, \mathrm{f}_i(x)\, dx + 2\,(\mu_i-\mu) \int_{\mathbb{R}} (x-\mu_i)\, \mathrm{f}_i(x)\, dx + (\mu_i-\mu)^2 \int_{\mathbb{R}} \mathrm{f}_i(x)\, dx \right] \\
&= \sum_i p_i \left[\sigma_i^2 + 0 + (\mu_i-\mu)^2 \right],
\end{aligned}$$

which proves (C.2). □

Appendix D. Explaining and modelling TTI-related modulations of flash scores produced by non-target stimuli

In section 5.3 we found that the average flash score for ERPs in partition nt01 (which gathers the responses to the second non-target following a target) was lower than for other non-target partitions. In this appendix we look at possible causes of TTI-related modulations of flash scores in the presence of non-targets and how the theory and methods developed in the paper would be affected by the use of (15).

When stimuli are presented at a sufficiently fast pace, the P300 produced in response to a target stimulus will appear within the epochs corresponding to the presentation of one or more non-targets following that target thereby deforming ERPs. Naturally, not all such deformations will influence a flash scorer. However, some may do so.

The SVM classifier used in the first stage of our Donchin speller weighs the contribution of the different EEG samples in order to maximise the distance between targets and non-targets. For simplicity let us think of it as a kind of matched filter looking for P300-like shapes, i.e., positive deflections of the EEG w.r.t. the baseline in a time window of around 300–500 ms after stimulus presentation. With the standard stimulus presentation rate of Donchin's speller, for the epochs in the nt01 partition, the P300 induced by the preceding target (the target epoch that started $2 \times \mathrm{SOA} = 350\,\mathrm{ms}$ before an nt01 epoch) shifts positively the baseline of the nt01 epochs thereby significantly deforming the ERP and making it less likely to produce a high flash score. Figure D1 illustrates the idea using a simple synthetic signal and an elementary classifier. We believe that this is the most likely reason for nt01 having more negative scores than other non-target partitions in our tests.

Of course, when using a different timing for stimuli, one might find that other non-target partitions are affected by this effect. The effect may also depend on the particular choice of stimulus paradigm and flash scoring algorithm. So, we cannot exclude that in different P300 based BCIs a more marked TTI modulation of non-target ERPs might be observed. For these reasons it is interesting to consider how the theory and algorithms presented in this paper would be modified if we wanted to also model and exploit such modulations.

Building a model where the score distribution of the non-targets depends on h would be conceptually straightforward and would pose no new challenges. The model for non-targets could be expressed using (15). Apart from small formal changes in the

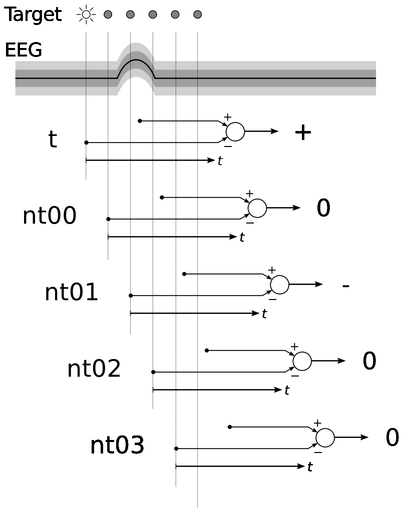


Figure D1. Possible explanation for nt01 being more negative than the other non-targets. For simplicity, let us imagine no ERPs are produced by non-target stimuli. When a target flash is presented to a subject, we observe a positive deflection of the EEG signal. This positive deflection occurs approximately 300–500 ms after the beginning of the epoch starting when the target event occurs. For simplicity let us imagine that our flash scorer simply returns the difference between the EEG at time 400 ms and the EEG at time 0 ms (relative to the start of the epoch). With our stylised EEG signal, this algorithm tends to give a positive score to target epochs and a zero score to non-target ones. In the case of nt01, however, the EEG at time 0 ms in an epoch corresponds to the EEG at time 2 × SOA = 350 ms of the target epoch acquired two flashes earlier. This causes the voltage at 0 ms to be significantly higher than that at 400 ms. In the presence of such an unusual non-target ERP, our simple difference-based scorer will give nt01 epochs an unusually low score.

formulae, the main substantial difference is that (A.5) and (A.6) should be rewritten as

$$S_{\hat{r},*} = \mathrm{w}(H^{III,*}) \left[\alpha^b_{\mathrm{t}\ \mathrm{H^I}} + \mathring{Y}^{I}\right] + \mathrm{w}(H^{IV,*}) \left[\alpha^b_{\mathrm{nt}\ H^{IX}} + \mathring{Y}^{III,*}\right] \quad \text{(D.1)}$$

and

$$S_{*,\hat{c}} = \mathrm{w}(H^{V,*}) \left[\alpha^b_{\mathrm{nt}\ H^{X}} + \mathring{Y}^{IV,*}\right] + \mathrm{w}(H^{VI,*}) \left[\alpha^b_{\mathrm{t}\ \mathrm{H^{II}}} + \mathring{Y}^{II}\right]\ , \quad \text{(D.2)}$$

where the variables H^{IX} and H^{X} determine which values of $\alpha_{\mathrm{nt\ h}}$ to use. Please note that, unlike the target case where the same H selects both the value of $\alpha_{\mathrm{t\ h}}$ and $w(h)$, in this case the pair of variables $H^{IV,*}$ and H^{IX} and the pair of variables $H^{V,*}$ and H^{X} are different i.i.d. variables.

The second term in (D.1) was called V'_* in Appendix A. Earlier its mean, $\mu_{V'_*}$, was found to be $\mathrm{E}[\mathrm{w}(H^{IV,*})]\ \alpha^b_{\mathrm{nt}}$. When using (D.1), $\mu_{V'_*} = \mathrm{E}[\mathrm{w}(H^{IV,*})\ \alpha^b_{\mathrm{nt}\ H^{IX}}]$. But, as $H^{IV,*}$ and H^{IX} are independent, this reduces to the product $\mathrm{E}[\mathrm{w}(H^{IV,*})] \times \mathrm{E}[\alpha^b_{\mathrm{nt}\ H^{IX}}]$, where the second factor is the average non-target score among the partitions, i.e., α^b_{nt}. In other words, even with a more sophisticated model of non-target scores, $\mu_{V'_*} = \mathrm{E}[\mathrm{w}(H^{IV,*})]\ \alpha^b_{\mathrm{nt}}$. The same applies to the first term in (D.2). This shows that a model that assumes the dependence of the non-target score distribution on h would make the mathematical formulation more complicated without giving any substantial advantage.

References

[1] Dornhege G, del R Millán J, Hinterberger T, McFarland D J and Müller K R (eds) 2007 *Toward Brain-Computer Interfacing* (The MIT Press)

[2] Allison B Z, Wolpaw E W and Wolpaw J R 2007 Brain-computer interface systems: progress and prospects. *Expert review of medical devices* **4** 463–474

[3] Cinel C, Poli R and Citi L 2004 Possible sources of perceptual errors in P300-based speller paradigm *Biomedizinische technik* **49**(1) 39–40 Proceedings of 2nd International BCI workshop and Training Course

[4] Citi L, Poli R, Cinel C and Sepulveda F 2008 P300-based BCI mouse with genetically-optimized analogue control *IEEE transactions on neural systems and rehabilitation engineering* **16** 51–61

[5] Hansen J C 1983 Separation of overlapping waveforms having known temporal distributions. *Journal of neuroscience methods* **9** 127–139

[6] Zhang J 1998 Decomposing stimulus and response component waveforms in ERP *Journal of neuroscience methods* **80** 49–63

[7] Spencer K M 2004 Averaging, detection and classification of single-trial ERPs *Event-related potentials. A method handbook* ed Handy T C (MIT Press) chap 10

[8] Luck S J 2005 *An introduction to the event-related potential technique* (Cambridge, Massachusetts: MIT Press)

[9] Poli R, Cinel C, Citi L and Sepulveda F 2010 Reaction-time binning: a simple method for increasing the resolving power of ERP averages *Psychophysiology* **47**(3) 467–485

[10] Citi L, Poli R and Cinel C 2009 High-significance averages of event-related potential via genetic programming *Genetic Programming Theory and Practice VII* Genetic and Evolutionary Computation ed Riolo R L, O'Reilly U M and McConaghy T (Ann Arbor: Springer) pp 135–157

[11] Farwell L A and Donchin E 1988 Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical neurophysiology* **70** 510–523

[12] Allison B Z, McFarland D J, Schalk G, Zheng S D, Jackson M M and Wolpaw J R 2008 Towards an independent brain-computer interface using steady state visual evoked potentials. *Clinical neurophysiology* **119** 399–408

[13] Hong B, Guo F, Liu T, Gao X and Gao S 2009 N200-speller using motion-onset visual response. *Clinical neurophysiology* **120** 1658–1666

[14] Donchin E and Coles M G H 1988 Is the P300 a manifestation of context updating? *Behavioral and brain sciences* **11** 355–372

[15] Polich J 2004 Neuropsychology of P3a and P3b: A theoretical overview *Brainwaves and mind: recent developments* ed Moore N C and Arikan K (Kjellberg Inc.) pp 15–29

[16] Polich J 2007 Updating P300: an integrative theory of P3a and P3b. *Clinical neurophysiology* **118** 2128–2148

[17] Polich J and Kok A 1995 Cognitive and biological determinants of P300: an integrative review. *Biological psychology* **41** 103–146

[18] Hagen G F, Gatherwright J R, Lopez B A and Polich J 2006 P3a from visual stimuli: task difficulty effects. *International journal of psychophysiology* **59** 8–14

[19] Sellers E W, Krusienski D J, McFarland D J, Vaughan T M and Wolpaw J R 2006 A P300 event-related potential brain-computer interface (BCI): the effects of matrix size and inter stimulus interval on performance. *Biological psychology* **73** 242–252

[20] Allison B Z and Pineda J A 2003 ERPs evoked by different matrix sizes: implications for a brain computer interface (BCI) system. *IEEE transactions on neural systems and rehabilitation engineering* **11** 110–113

[21] Polich J 1990 Probability and inter-stimulus interval effects on the P300 from auditory stimuli. *International journal of psychophysiology* **10** 163–170

[22] Fitzgerald P G and Picton T W 1981 Temporal and sequential probability in evoked potential studies. *Canadian journal of psychology* **35** 188–200

[23] Allison B Z and Pineda J A 2006 Effects of SOA and flash pattern manipulations on ERPs, performance, and preference: implications for a BCI system. *International journal of psychophysiology* **59** 127–140

[24] Gonsalvez C L and Polich J 2002 P300 amplitude is determined by target-to-target interval. *Psychophysiology* **39** 388–396

[25] Croft R J, Gonsalvez C J, Gabriel C and Barry R J 2003 Target-to-target interval versus probability effects on P300 in one- and two-tone tasks. *Psychophysiology* **40** 322–328

[26] Squires K C, Wickens C, Squires N K and Donchin E 1976 The effect of stimulus sequence on the waveform of the cortical event-related potential. *Science (New York, N.Y.)* **193** 1142–1146

[27] Gonsalvez C J, Gordon E, Anderson J, Pettigrew G, Barry R J, Rennie C and Meares R 1995 Numbers of preceding nontargets differentially affect responses to targets in normal volunteers and patients with schizophrenia: a study of event-related potentials. *Psychiatry research* **58** 69–75

[28] Bonala B, Boutros N N and Jansen B H 2008 Target probability affects the likelihood that a P300 will be generated in response to a target stimulus, but not its amplitude. *Psychophysiology* **45** 93–99

[29] Rakotomamonjy A and Guigue V 2008 BCI competition III: dataset II- ensemble of SVMs for BCI P300 speller *IEEE transactions on bio-medical engineering* **55** 1147–1154

[30] Martens S M M, Hill N J, Farquhar J and Schölkopf B 2009 Overlap and refractory effects in a brain-computer interface speller based on the visual P300 event-related potential. *Journal of neural engineering* **6** 026003

[31] Citi L, Poli R and Cinel C 2009 Exploiting P300 amplitude variations can improve classification accuracy in Donchin's BCI speller *4th International IEEE EMBS Conference on Neural Engineering* (Antalya) pp 478–481

[32] Salvaris M and Sepulveda F 2009 Perceptual errors in the Farwell and Donchin matrix speller *4th International IEEE EMBS Conference on Neural Engineering* (Antalya, Turkey) pp 275 – 278

[33] Blankertz B, Müller K R, Krusienski D J, Schalk G, Wolpaw J R, Schlögl A, Pfurtscheller G, del R Millán J, Schröder M and Birbaumer N 2006 The BCI competition III: Validating alternative approaches to actual BCI problems. *IEEE transactions on neural systems and rehabilitation engineering* **14** 153–159

[34] Kaper M, Meinicke P, Grossekathoefer U, Lingner T and Ritter H 2004 BCI competition 2003–data set IIb: support vector machines for the P300 speller paradigm. *IEEE transactions on bio-medical engineering* **51** 1073–1076

[35] Guan C, Thulasidas M and Wu J 2004 High performance P300 speller for brain-computer interface *IEEE international workshop on biomedical circuits and systems* pp S3/5/INV–S3/13–16

[36] Bayliss J D 2003 Use of the evoked potential P3 component for control in a virtual apartment. *IEEE transactions on neural systems and rehabilitation engineering* **11** 113–116

[37] Wilk M B and Gnanadesikan R 1968 Probability plotting methods for the analysis of data. *Biometrika* **55** 1–17

[38] Kübler A, Mushahwar V K, Hochberg L R and Donoghue J P 2006 BCI meeting 2005–workshop on clinical issues and applications. *IEEE transactions on neural systems and rehabilitation engineering* **14** 131–134

[39] Rakotomamonjy A and Guigue V online at `http://asi.insa-rouen.fr/enseignants/~arakotom/code/bciindex.html`

[40] Hill J, Farquhar J, Martens S M, Biessmann F and Schölkopf B 2008 Effects of stimulus type and of Error-Correcting code design on BCI speller performance *The 22nd Annual Conference on Neural Information Processing Systems* (Vancouver)

[41] Brunner P, Joshi S, Briskin S, Wolpaw J, Bischof H and Schalk G 2010 Does the "P300" speller depend on eye gaze? *Tools for Brain-Computer Interaction (TOBI) Workshop on Integrating brain-computer interfaces with conventional assistive technology* (Graz) (poster)

[42] Treder M S and Blankertz B 2010 (C)overt attention and visual speller design in an ERP-based Brain-Computer Interface *Behavioral and brain functions* **6** 28

[43] Schreuder M, Blankertz B and Tangermann M 2010 A new auditory multi-class Brain-Computer Interface paradigm: spatial hearing as an informative cue *PLoS One* **5** e9813

[44] Brouwer A M and van Erp J B F 2010 A tactile P300 Brain-Computer Interface. *Frontiers in neuroscience* **4** 19

[45] Ulyanov V V 1981 An estimate for the rate of convergence in the central limit theorem in a real separable Hilbert space *Mathematical notes* **29** 78–82 translated from Matematicheskie Zametki, Vol. 29, No. 1, pp. 145–153, January, 1981.