Published in final edited form as:

J Neural Eng. 2016 December ; 13(6): 066007. doi:10.1088/1741-2560/13/6/066007.

Using the Detectability Index to Predict P300 Speller Performance

B.O. Mainsah¹, L.M. Collins^{1,*}, and C.S. Throckmorton¹

¹Duke University, Department of Electrical and Computer Engineering, Durham, NC, USA

Abstract

Objective—The P300 speller is a popular brain-computer interface (BCI) system that has been investigated as a potential communication alternative for individuals with severe neuromuscular limitations. To achieve acceptable accuracy levels for communication, the system requires repeated data measurements in a given signal condition to enhance the signal-to-noise ratio of elicited brain responses. These elicited brain responses, which are used as control signals, are embedded in noisy electroencephalography (EEG) data. The discriminability between target and non-target EEG responses defines a user's performance with the system. A previous P300 speller model has been proposed to estimate system accuracy given a certain amount of data collection. However, the approach was limited to a static stopping algorithm, i.e. averaging over a fixed number of measurements, and the row-column paradigm. A generalized method that is also applicable to dynamic stopping algorithms and other stimulus paradigms is desirable.

Approach—We developed a new probabilistic model-based approach to predicting BCI performance, where performance functions can be derived analytically or via Monte Carlo methods. Within this framework, we introduce a new model for the P300 speller with the Bayesian dynamic stopping (DS) algorithm, by simplifying a multi-hypothesis to a binary hypothesis problem using the likelihood ratio test. Under a normality assumption, the performance functions for the Bayesian algorithm can be parameterized with the detectability index, a measure which quantifies the discriminability between target and non-target EEG responses.

Main results—Simulations with synthetic and empirical data provided initial verification of the proposed method of estimating performance with Bayesian DS using the detectability index. Analysis of results from previous online studies validated the proposed method.

Significance—The proposed method could serve as a useful tool to initially assess BCI performance without extensive online testing, in order to estimate the amount of data required to achieve a desired accuracy level.

Keywords

Brain-computer interface (BCI); P300 speller; Dynamic stopping; BCI Performance prediction; Likelihood Ratio; Detectability index

*Corresponding author: leslie.collins@duke.edu.

1 Introduction

The primary focus of brain-computer interface (BCI) research is to restore or enhance control and communication abilities for individuals with severe neuromuscular limitations who retain cognition [1, 2]. BCIs enable neuronal activity encoded in brain signals to be extracted directly, decoded and translated into command outputs to control external devices. The presentation of a rare stimulus event amongst a sequence of more frequently occurring stimuli elicits a distinct brain response or event-related potential (ERP) which includes a large positive deflection called the P300 signal [3]. The P300-based BCI operates like a virtual keyboard by relying on the elicited ERPs to enable a user make desired or target selections from an array of choices to communicate their intent [4].

Various studies have shown that the P300 speller can provide a reliable means of communication, including in target BCI end-users [5, 6, 7]. However, the adoption of P300 spellers for home-based communication has been limited, primarily due to their slow character selection rates when compared to other commercially available alternatives [2, 8, 9]. The ERP control signals are embedded in noisy electroencephalography (EEG) data. Repetitive data measurements are required to enhance the signal-to-noise ratio (SNR) of the elicited ERPs in order to achieve acceptable accuracy levels for communication. The typical strategy is to determine character selection after a fixed amount of data collection, termed static data collection or *static stopping*. The amount of data collection prior to character selection can either be fixed across all users or customized for a given user [10].

Recently in the BCI literature, there has been a shift towards adaptive strategies that allow the BCI system to vary the amount of data collection based on a threshold function [11], termed *dynamic stopping*. The BCI system has the flexibility to collect more or less data depending on the current signal SNR quality. Dynamic stopping algorithms have been shown to improve performance when compared to static stopping [12, 13, 14, 15, 16, 17, 18, 19], with positive preliminary online results in target BCI end-users [7]. For practical reasons, a data collection limit is usually imposed when using a dynamic stopping algorithm as convergence to the stopping threshold is not always guaranteed within a reasonable amount of time. Setting an appropriate data collection limit can have a significant impact on the maximum achievable performance level of a user, and this is particularly important in target BCI end-users where there is need to improve BCI communication rates.

The ability to predict a user's performance has significant theoretical and practical implications. Performance predictions can be used to evaluate the likelihood of success of a user with a BCI prior to online use, to optimize parameter selection with a principled method to improve a user's performance, or to evaluate the potential impact of new BCI algorithms or strategies on a user's performance without the need for extensive online testing. Usually, performance is predicted from cross-validation of the training data collected during a calibration run. In general, the performance predictions are used for parameter selection in a grid search of parameter values that maximize the performance based on the training data (e.g. [10, 13]). However, EEG data are non-stationary, and this can result in a model mismatch of parameters developed from the training dataset if the shift in data statistics from system calibration is significant.

Most of the previous performance prediction methods that have been proposed in the literature focus on developing EEG-based features as performance predictors. The performance functions are usually estimated through regression analysis using empirical estimates obtained from user datasets. Some predictors that have been developed for P300 speller performance include those based on ERP signal latency [20] and signal amplitudes [21]. However, these methods are easily susceptible to noise due to EEG signal variability and granular performance estimates from a low number of observations, and so require a large user database to minimize model variability. For users from a population with disabilities, the varying progressions of their disease may result in significant differences between users.

We hypothesize that methods for assessing long-term BCI performance that do not rely on noisy empirical estimates and that take into account possible changes in EEG data statistics after calibration, provide for more flexibility in predicting performance. A probabilistic model of the BCI selection process provides a means to account for shifting data statistics, particularly if the model is parametric. Colwell *et al.* [22] proposed an underlying theory to predict a user's accuracy based on a probabilistic model of P300 speller operation. However, their approach was limited to a specific static stopping algorithm and the row-column speller paradigm [4]. Given the increased interest in dynamic stopping algorithms and alternative stimulus presentation paradigms, a generalized approach to predict performance is desirable.

In this study, we extend the approach outlined in Colwell *et al.* to develop new framework for predicting BCI performance based on a probabilistic model of a generic BCI. This approach is inclusive of the various types of algorithms used in the literature as it accommodates different stimulus presentation paradigms and data stopping criteria. Using our method, performance functions can either be obtained analytically or via Monte Carlo simulations, eliminating the need for empirical BCI data from a large user database.

We demonstrate the utility of our performance prediction method by deriving performance functions for the Bayesian dynamic stopping algorithm developed in Throckmorton *et al.* [18] for the P300 speller. A user's P300 speller performance depends on how well the system's classifier can distinguish between target and non-target EEG responses. This classifier discriminability can be quantified by a distance metric called the *detectability index*. With the Bayesian algorithm, the classifier detectability index is a statistical measure that can be used as a performance predictor. We present initial evidence to verify our new performance prediction method using results from simulations of P300 spelling runs; all simulations were performed in MATLAB® R2014b (The MathWorks Inc., Natick, MA). We also analyze data from our previous P300 speller studies for validation of the performance prediction method with online BCI use.

2 BCI Performance Prediction

Our performance prediction method is based on a probabilistic model of a generic ERP-based BCI and its operation. A schematic of this model is shown in Figure 1. A user is presented with a set of choices, $\{C_m\}_{m=1}^M$, one of which is assumed to be the user's desired or target character during a selection process. Electrodes are used to acquire EEG signals,

which are amplified, filtered and digitized for further processing. To select a target character, C^* , the user focuses on that character as subsets of characters, $\mathcal{F}_t \subset \{C_m\}_{m=1}^M$, are randomly presented, where t denotes the time index. For example, for the row-column paradigm (RCP) [4], which is the predominant stimulus presentation paradigm that is used in the literature [8], the flash groups are the rows and columns of a grid, presented in a random order without replacement. The presentation of a stimulus to the user elicits an ERP depending on the type of stimulus event.

Following each stimulus presentation, the user's EEG response to the stimulus is scored with the system's classifier. Time windows of EEG data from select electrodes are processed to extract relevant features that convey the user's intent. These features are scored with a user-specific classifier to obtain a classifier score, y_t . The classifier score is used to update a function that quantifies the possibility of each of the BCI choices to be the user's target choice given data collection. We denote this function as the *character cumulative response function* (CCRF), $\{\Theta_m(t)\}_{m=1}^M$, for an M -choice BCI. For the P300 speller, $\Theta_m(t) = g(C_m, \mathcal{F}_t, \mathbf{Y}_t)$, represents a measure of the possibility of character, C_m , of being the target character, given a series of stimulus event presentations, $\mathcal{F}_t = [\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_t]$, and classifier score responses, $\mathbf{Y}_t = [y_1, y_2, \dots, y_t]$. Based on the literature, $\Theta_m(t)$ is usually a cumulative score, e.g. [4], or a probability value, e.g. [18].

The BCI collects data and terminates data collection at a stopping time, t_s , based on a stopping rule that can either be static or dynamic. In static stopping (SS), a fixed amount of data is collected, i.e. $t_s = t_{\max}$. In dynamic stopping (DS), the amount of data collection is varied, i.e. $1 \leq t_s \leq t_{\max}$, based on a threshold function; usually when the maximum CCRF value attains a pre-set threshold value, Θ_{th} . For practical reasons, a data collection limit is imposed when using a DS algorithm as convergence to the stopping threshold is not always guaranteed in order to make a selection within a reasonable amount of time. We characterize a DS algorithm as *untruncated* if an unlimited amount of data collection is allowed, and *truncated* if a data collection limit is imposed. In this study, within the context of data collection, we define a *sequence* as a unit of data collection which consists of the number of defined flash groups in the base pattern of a stimulus presentation paradigm. For example, for the RCP with an $R \times C$ grid, 1 sequence = $R+C$ flashes and $t_{\max} = s \times (R+C)$ flashes, where s is the sequence limit.

When data collection is stopped, the BCI makes a selection based on a decision rule to estimate the user's target character, $\hat{C}^*$, which is usually the character that maximizes the CCRF. The performance of the system's classifier affects not only the ability to distinguish between non-target and target EEG responses and correctly select the user's target character, but also the rate of convergence to the decision threshold, in the case of a DS algorithm.

Given the above representation of BCI operation, the model can be analyzed to study the behavior of the system with a given user to evaluate the average performance. In particular, analyzing the evolution of the random variables, $\{\Theta_m(t)\}_{m=1}^M$, with data collection. Some useful random variable analysis tools include:

- *Random sums:* For some algorithms, $\Theta_m(t)$ can be expressed as an additive operation:

$$\Theta_m(t) = \sum_{\tau=1}^t \theta_m(\tau) \quad (1)$$

where $\theta_m(\tau)$ is a function of the classifier score at time index, τ . For example, the most common function used in the literature is the cumulative moving average (CMA) of classifier scores [4], which is updated for each character only when it is presented. If the probability density functions (pdf) of $\theta_m(\tau)$, $\forall \tau = 1, \dots, t$, are known, the pdf of $\Theta_m(t)$ can be determined accordingly via recursive convolutions (see Appendix A for details).

- *Order statistics:* The maximum of the CCRF values, $\Theta_{\max}(t)$, is used to determine either the selected character, or in the case of a dynamic stopping algorithm, to determine when to stop data collection when compared to the threshold value:

$$\Theta_{\max}(t) = \max \{\Theta_1(t), \Theta_2(t), \dots, \Theta_m(t)\} \quad (2)$$

Determining the probability distribution of the maximum of N random variables is a problem in order statistics (see Appendix C for details). While the most commonly studied order statistics are those for independent random variables [23], we will also use the approximation proposed by Clark [24] for correlated random variables, to account for different stimulus presentation paradigms.

- *Random inequalities:* When determining a user's average performance, the probability of occurrence of events have to be evaluated. Some examples include comparing two random variables, e.g. $P(\Theta_i(t) > \max_{j \neq i} \{\Theta_j(t)\})$, or a random variable and a fixed value, e.g. $P(\Theta_{\max}(t) > \Theta_{th})$ (see Appendix B for details).

A user's performance with a BCI is generally characterized by two measures: the data stopping time and accuracy. Performance functions can be derived based on the stopping and decision rules, respectively. The expected stopping time (EST) is the average number of stimulus event presentations prior to making a decision. With a static stopping criterion, the amount of data collection is fixed. With a dynamic stopping criterion, data collection is stopped when any character's cumulative response attains the pre-set threshold value, Θ_{th} , or the data collection limit is attained, $t_{\max}$. To account for a dynamic stopping criterion where t_s is variable, determining the EST requires averaging over all possible stopping times:

$$\text{EST} = \sum_{t=1}^{t_{\max}} t P(t_s = t) \quad (3a)$$

$$= \sum_{t=1}^{t_{\max}-1} tP(\{\Theta_{\max}(t-1) < \Theta_{th}\} \cap \{\Theta_{\max}(t) \geq \Theta_{th}\}) + t_{\max}P(\Theta_{\max}(t_{\max}-1) \geq \Theta_{th}) \quad (3b)$$

The accuracy, A , is the average probability that the BCI correctly selects the target character after data collection is stopped. The BCI correctly selects the target character if at t_s , the target character's cumulative response value exceeds that of all the non-target characters, or exceeds the maximum response of the non-target characters, i.e. $\hat{C}^* = C^*$ if

$\Theta_i(t_s) \geq \max_{j \neq i} \{\Theta_j(t_s)\} | C_i = C^*$. Similar to the EST, accuracy is determined by averaging over all possible stopping times, as well as over all possible character choices:

$$P(\hat{C}^* = C_i | C_i = C^*) = \sum_{t=1}^{t_{\max}} P\left(\{t_s = t\} \cap \left\{\max_{j \neq i} \{\Theta_j(t)\} < \Theta_i(t)\right\} | C_i = C^*\right) \quad (4a)$$

$$A = \sum_{i=1}^M P(\hat{C}^* = C_i | C_i = C^*) P(C_i = C^*) \quad (4b)$$

Given equations (3)–(4), analytic expressions for the performance measures can be derived. Alternatively, Monte Carlo simulations [25] based on the model shown in Figure 1 can be used to derive the performance functions, if the solutions to (3)–(4) are not tractable.

The performance prediction method developed in Colwell *et al.* [22] falls within this framework, specifically estimating accuracy with the P300 speller using the RCP and the CMA algorithm with a static stopping criterion [4]. Under these conditions, the BCI correctly selects the target character if the following event occurs: $\{\text{mean target row classifier score} > R - 1 \text{ mean non-target row classifier scores}\} \cap \{\text{mean target column score} > C - 1 \text{ mean non-target column classifier scores}\}$. While the Colwell *et al.* method is limited to the RCP and the CMA static stopping algorithm, the approach outlined here enables the performance measures to be analyzed independent of stimulus presentation paradigm and data stopping criterion.

In the next section, we describe the derivation of the performance functions for the Bayesian dynamic stopping algorithm [18], which has tractable analytical solutions under certain conditions. The discriminability of the BCI system's classifier, i.e. how well the BCI system can distinguish between target and non-target EEG responses, determines a user's level of performance. Given this discriminability, it is possible to estimate performance with the Bayesian dynamic stopping algorithm.

3 Theoretical Model

3.1 The Bayesian Dynamic Stopping Algorithm

Throckmorton *et al.* [18] developed a Bayesian approach to dynamically control data collection based on achieving a certain level of confidence that a character is the target character. The CCRF is a probability distribution, $\{P_m(t)\}_{m=1}^M$, which is maintained over the characters in the grid. With each stimulus event presentation, new evidence is incorporated into the model via Bayesian inference to update the character probabilities. Data collection is stopped when a character's probability attains a threshold value, P_{th} , within a data collection limit.

In the offline portion of the algorithm, the likelihood functions of the non-target and target classifier scores, $p(y|H_0)$ and $p(y|H_1)$, respectively, are estimated from the training data and used in the Bayesian probability update process. Prior to data collection, character probabilities are initialized, $P_m(0) = P(C_m = C^*|Y_0, \mathcal{F}_0)$, indicating the initial confidence level that each character, C_m , is the target character, C^* . After each stimulus event presentation, the classifier score, y_t , is used to update the character probabilities:

$$P(C_m = C^* | Y_t, \mathcal{F}_t) = \frac{P(C_m = C^* | Y_{t-1}, \mathcal{F}_{t-1})p(y_t | C_m = C^*, \mathcal{F}_t)}{\sum_{j=1}^M P(C_j = C^* | Y_{t-1}, \mathcal{F}_{t-1})p(y_t | C_j = C^*, \mathcal{F}_t)} \quad (5)$$

where $P(C_m = C^* | Y_t, \mathcal{F}_t)$ is the posterior probability that $C_m = C^*$, given all the classifier responses, $Y_t = [y_1, \dots, y_t]$, and the t^{th} flashed subset, $\mathcal{F}_t$; $P(C_m = C^* | Y_{t-1}, \mathcal{F}_{t-1})$ is the prior probability that $C_m = C^*$; $p(y_t | C_m = C^*, \mathcal{F}_t)$ is the likelihood of the current classifier score, y_t , given that the character was or was not present in $\mathcal{F}_t$; and the denominator normalizes over all the character probabilities. The likelihood, $p(y_t | C_m = C^*, \mathcal{F}_t)$, for each character is set depending on whether the character is in the current flashed subset:

$$p(y_t | C_m = C^*, \mathcal{F}_t) = \begin{cases} p(y_t | H_0), & C_m \notin \mathcal{F}_t \\ p(y_t | H_1), & C_m \in \mathcal{F}_t \end{cases} \quad (6)$$

Data collection is stopped at t_s , either when a character probability attains the threshold value (set at $P_{th} = 0.9$ in [18]) or t_{max} is reached. The character with the highest probability after data collection is selected as the user's intended target character, $\hat{C}^* = \arg\max_m P_m(t_s)$.

For the Bayesian DS algorithm, the random variables that are analyzed to evaluate system performance are the Bayesian probabilities: $\{P_m(t)\}_{m=1}^M$. The classifier score likelihood, $p(y_t | C_m = C^*, \mathcal{F}_t)$, which is incorporated into the model at each probability update step, is also a random variable. The denominator term in the probability update equation (5) renders the analytical derivation of the performance functions mathematically intractable. However, due to the binary choice in the character likelihood assignment (6), we can instead consider the ratio, $p(y_t | H_1)/p(y_t | H_0)$, which is commonly referred to as the *likelihood ratio*.

The likelihood ratio test is a well-established statistical test used in decision making in binary hypothesis testing [26]. It has an equivalent formulation in the Bayesian domain. By operating in the likelihood ratio domain, a user's performance level can be parameterized with a single value via a measure called the *detectability index*. The detectability index is a metric that quantifies the discriminability between two normal probability distributions, i.e. in this case, for target and non-target classifier scores. We will first demonstrate that modeling P300 speller operation in the likelihood ratio domain is equivalent to modeling in the Bayesian domain. Following this substitution with the likelihood ratio domain, we will derive the performance functions for the Bayesian DS algorithm.

3.2 The Likelihood Ratio Test

Statistical hypothesis testing involves the process of determining how likely a set of observations are to be generated under a given statistical hypothesis. Each hypothesis represents a probabilistic model of a random variable, $H_i: \mathbf{x} \sim p(\mathbf{x}|H_i)$, where $p(\mathbf{x}|H_i)$ is the likelihood of observation $\mathbf{x}$ under the hypothesis, H_i . The likelihood ratio test (LRT) is most commonly used for binary hypothesis testing [26]. A decision is made to accept or reject a hypothesis after comparing the likelihood ratio, $\lambda(\mathbf{x})$, to a threshold value, η .

$$\lambda(\mathbf{x}) = \frac{p(\mathbf{x}|H_1)}{p(\mathbf{x}|H_0)} \quad (7)$$

where the *null* or H_0 hypothesis is selected if $\lambda(\mathbf{x}) < \eta$, or the *alternative* or H_1 hypothesis is selected if $\lambda(\mathbf{x}) \geq \eta$. If the hypotheses have prior probabilities, $P(H_0)$ and $P(H_1)$, respectively, the threshold value that minimizes the probability of error, P_e , is $\eta = P(H_0)/P(H_1)$.

In general, for a multi-hypothesis or M -ary hypothesis test, we can determine probabilistically how likely a set of observations was generated under a given hypothesis by using Bayes' rule. We consider another minimum P_e detector which minimizes the Bayes' risk [26]. From Bayes' rule, the posterior probability of a hypothesis given an observation, $P(H_i|\mathbf{x})$, can be determined by incorporating the likelihood of the observation and the hypothesis prior:

$$P(H_i|\mathbf{x}) = \frac{p(\mathbf{x}|H_i)P(H_i)}{p(\mathbf{x})} \quad (8)$$

The hypothesis with the maximum *a posteriori* probability (MAP) is selected as the true hypothesis. The Bayesian DS algorithm is an example of a MAP decision test.

When $M = 2$, i.e. a binary hypothesis test case, by rearranging (8), we can express the posterior Bayesian probability as a function of the likelihood ratio:

$$P(H_1|\mathbf{x}) = \frac{\frac{p(\mathbf{x}|H_1)P(H_1)}{p(\mathbf{x}|H_0)P(H_1)}}{\frac{p(\mathbf{x}|H_1)P(H_1)}{p(\mathbf{x}|H_0)P(H_1)} + \frac{p(\mathbf{x}|H_0)P(H_0)}{p(\mathbf{x}|H_0)P(H_1)}} = \frac{\lambda(\mathbf{x})}{\lambda(\mathbf{x}) + \frac{P(H_0)}{P(H_1)}} \quad (9)$$

$$P(H_0|\mathbf{x}) = \frac{\lambda(\mathbf{x})^{-1}}{\lambda(\mathbf{x})^{-1} + \frac{P(H_1)}{P(H_0)}} \quad (10)$$

Hence, we can decide on a hypothesis either based on a threshold of the likelihood ratio or the corresponding threshold of the Bayesian probability. This allows us to operate in either of the domains to achieve equivalent results.

The sequential likelihood ratio test (SLRT) reformulates the LRT as a sequential analysis problem with a dynamic data collection process [27]. Given a sequence of observations, $\mathbf{x}_t = [x_1, x_2, \dots, x_t]$, the goal of the SLRT is to decide which hypothesis is correct within a desirable level of error, in as few observations as possible. The cumulative likelihood ratio, $\Lambda(\mathbf{x}_t)$, is evaluated based on the ratio of the joint likelihood of the observations, $p(x_1, x_2, \dots, x_t|H_i)$. For notational simplicity, we will refer to $\Lambda(\mathbf{x}_t)$ and $\lambda(\mathbf{x}_t)$ as Λ_t and λ_t for the cumulative and single observation likelihood ratio, respectively.

Decision regions for Λ_t are specified to either continue data collection or stop data collection and select one of the hypotheses based on specified thresholds:

$$\Lambda_t = \frac{p(x_1, x_2, \dots, x_t|H_1)}{p(x_1, x_2, \dots, x_t|H_0)} \quad (11)$$

$$\Lambda_t \begin{cases} \leq \mathcal{A} \Rightarrow \text{choose } H_0, \text{ terminate} \\ \geq \mathcal{B} \Rightarrow \text{choose } H_1, \text{ terminate} \\ \in (\mathcal{A}, \mathcal{B}) \Rightarrow \text{take another sample} \end{cases} \quad (12)$$

where $\mathcal{A}$ and $\mathcal{B}$ are the decision thresholds. $\mathcal{A}$ and $\mathcal{B}$ are chosen based on specified probabilities of detection, $P_D = P(\text{deciding “}H_1\text{”}|H_1 \text{ is the true hypothesis})$ and false alarm, $P_{FA} = P(\text{“}H_1\text{”}|H_0)$:

$$\mathcal{A} = \frac{1 - P_D}{1 - P_{FA}} \quad (13a)$$

$$\mathcal{B} = \frac{P_D}{P_{FA}} \quad (13b)$$

$\mathcal{A}$ and $\mathcal{B}$ can also be specified in terms of class conditional error rates, α and β , where $\alpha = P_{FA}$ and $\beta = 1 - P_D$. Likewise, Bayes' rule can be implemented sequentially, with data collection stopped when a hypothesis' posterior probability attains a preset threshold value prior to decision making, as is the case with the Bayesian DS algorithm.

The performance of the SLRT is characterized by the operating characteristic and the average sample number functions. The operating characteristic (OC) function, $\mathcal{L}(\theta_i)$, is the probability of terminating the test with an H_0 decision, given that H_i is the true hypothesis. The SLRT guarantees minimum error levels of α and β , under each hypothesis case. Also, the average probability of error, P_e , is determined from the hypotheses priors and the class conditional probability of errors:

$$P_e = \alpha P(H_0) + \beta P(H_1) \quad (14)$$

The average sample number (ASN) is the average number of observations prior to termination of the test. For the SLRT, if the observations are assumed to be independent and identically distributed (i.i.d.), Λ_t can be expressed as the product of the individual likelihood ratios of the observations, or alternatively as a logarithmic sum, for convenience:

$$\Lambda_t = \prod_{\tau=1}^t \frac{p(x_\tau|H_1)}{p(x_\tau|H_0)} \quad (15a)$$

$$\ln \Lambda_t = \sum_{\tau=1}^t \ln \lambda_\tau \quad (15b)$$

The ASN for the SLRT in an i.i.d. case can be approximated with Wald's equation (29) for random sums [23] (see Appendix A). Since $\ln \Lambda_t$ can be expressed as a random sum of $\ln \lambda_\tau$'s, the ASN can be estimated under a given hypothesis, H_i :

$$[T|H_i] = \frac{[\ln \Lambda_t|H_i]}{[\ln \lambda_t|H_i]} \quad (16a)$$

$$[T|H_1] = \frac{P_D \ln \mathcal{B} + (1 - P_D) \ln \mathcal{A}}{[\ln \lambda_t|H_1]} \quad (16b)$$

$$[T|H_0] = \frac{(1-P_{FA})\ln\mathcal{B} + P_{FA}\ln\mathcal{A}}{[\ln\lambda_t|H_0]} \quad (16c)$$

where $\mathbb{E}[T|H_i]$ is the ASN; $\mathbb{E}[\ln \Lambda_T|H_i]$ is the expected cumulative log-likelihood ratio (CLLR) when data collection is stopped, and $\mathbb{E}[\ln \Lambda_d|H_i]$ is the expected observational log-likelihood ratio (OLLR), all given H_i is the true hypothesis.

It should be noted that α and β define the maximum error levels attainable when evaluating the OC functions, and (16) define the asymptotic lower bounds for the ASN. These performance approximations omit possible overshoots when the cumulative likelihood ratio crosses either of the stopping thresholds, which occurs commonly in real world testing. For the SLRT with i.i.d. observations, better approximations for the performance functions can be obtained by solving a series of inductive integral equations governing the OC and ASN functions [28, 29].

In the Bayesian DS algorithm as implemented in [18], the observations are considered to be independent. We can express an equivalent cumulative likelihood ratio formulation of the Bayesian probabilities for each character as a CLLR sum. A corresponding likelihood ratio formulation for the Bayesian DS algorithm is more convenient as it avoids having to analyze the more complex denominator term in the Bayesian update equation (5).

The performance of a hypothesis test depends on how discriminable the probability models under each hypothesis are from each other. The detectability index, d , is a measure that quantifies the discriminability between two probability density functions by their separation and spread [26]. For two normally distributed random variables, with means and standard deviations, (μ_0, σ_0^2) and (μ_1, σ_1^2) , as shown in Figure 2, the detectability index can be computed:

$$d = \frac{\mu_1 - \mu_0}{\sqrt{0.5(\sigma_1^2 + \sigma_0^2)}} \quad (17)$$

where we use the formula proposed in [30] to account for unequal variances. The detectability index is used to parameterize the probability distribution of $\ln \lambda$ under each hypothesis [26]:

$$\ln \lambda \sim \begin{cases} \mathcal{N}(-d^2/2, d^2)|H_0 \\ \mathcal{N}(d^2/2, d^2)|H_1 \end{cases} \quad (18)$$

Hence, operating in the likelihood ratio domain provides us the convenience of quantifying a user's performance level via the P300 classifier discriminability. For example, given a user's training data, the respective means and variances of the classifier score likelihoods can be used to determine how likely a user will perform with the system. It is also easy to visualize

and study performance across a range of levels in a 1-D space via d , as opposed to a 4-D space based on the respective probability distribution parameters, $(\mu_0, \sigma_0^2, \mu_1, \sigma_1^2)$, as is required in the Colwell *et al.* [22] method.

3.3 Performance Estimation for the Bayesian Dynamic Stopping Algorithm

Performance measures have been studied for the binary SLRT [27], as well as multi-hypothesis tests [31, 32, 33, 34]. However, additional considerations are required when deriving the performance estimates for the Bayesian DS algorithm within the P300 speller context. Most sequential tests assume the availability of an infinite amount of time to make decisions, which is not practical for real world applications. Also, most sequential tests assume each hypothesis represents a distinct probability model and that the observations are i.i.d.

For this application, the classifier scores in the Bayesian DS algorithm are assumed to be independent but not necessarily identically distributed as the scores are generated as a result of random presentation of both non-target and target stimulus events. If a subset of characters is presented in a stimulus event, each classifier score observation is technically generated from a subset of the hypotheses. Even with single character presentation, the probability model of the classifier score associated with a given character depends on whether it is or is not the target character. Nonetheless, with these considerations, the P300 speller model with the Bayesian DS algorithm can be simplified from an M -ary to a binary hypothesis problem and the likelihood ratio test exploited to derive the performance functions more conveniently.

The importance of the discriminability of the classifier score likelihood functions in the Bayesian update process can be visualized by considering the two cases of non-target and target stimulus events. Figure 3 illustrates two scenarios of classifier scores produced in the presence/absence of the target character in a flashed subset. If the target character is in the flashed subset, $C^* \in \mathcal{C}_b$, a P300 signal will likely be elicited, resulting in a high classifier score, hence $p(y_d|H_1) > p(y_d|H_0)$ or $\lambda > 1$. Conversely, if $C^* \notin \mathcal{C}_b$, it is likely that $p(y_d|H_1) < p(y_d|H_0)$ or $\lambda < 1$. In both cases, the target character probability is updated with a high likelihood value, i.e. $p(y_d|C_m = C^*, \mathcal{C}_d)$ according to (6), and the target character probability should increase progressively and converge to 1.

Consequently, the P300 speller system can also be viewed as evaluating M different binary hypothesis tests (15b). Each time a character is flashed, the BCI evaluates the CLLRs to determine if it is or is not the target character:

$$L_m(t) = \sum_{\tau=1}^t l_m(\tau) \quad (19a)$$

$$l_m(\tau) = \begin{cases} \ln \left(\frac{p(y_\tau|H_1)}{p(y_\tau|H_0)} \right), & \text{if } C_m \in \mathcal{F}_\tau \\ 0, & \text{if } C_m \notin \mathcal{F}_\tau \end{cases} \quad (19b)$$

where $L_m(t)$ is the CLLR and $I_m(t)$ is the OLLR for character C_m . With increased data collection, $L_m(t)$ for the target character should increase progressively towards the upper threshold limit of the likelihood ratio test, while those of the non-target characters should decrease.

Since $L_m(t)$ is a sum of random variables, its probability distribution can be determined accordingly by convolution (Appendix A), with the probability distribution of $I_m(\tau)$ defined according to the presence/absence of a target character in a flash group. If the classifier scores are normally distributed, then $I_m(\tau)$ and $L_m(t)$ are also normally distributed (18). The random variable, $L_m(t)$, can now be analyzed to derive the performance functions.

It should be noted that the derivation of the performance functions for most sequential tests assume the availability of an infinite amount of time to make a decision. For the untruncated, SLRT, α and β define the maximum class conditional error levels. For the untruncated Bayesian DS algorithm, the stopping threshold probability defines the minimum accuracy level that can be achieved. Data truncation will affect both a test's error level and the expected stopping time. For example, in the truncated SLRT, $\ln \Lambda_t$ might not necessarily converge to either of the threshold values by $t_{\max}$ (see Appendix D for details). As a result, a truncated sequential test may achieve higher error levels than the maximum level guaranteed by an untruncated test [35]. The imposition of a data truncation limit is thus an important consideration when deriving the performance functions for the Bayesian DS algorithm.

3.3.1 Accuracy—We use equations (4) to determine accuracy with the Bayesian DS

algorithm. For example, if $Y_i = L_i(t)$ and $X_i = \max_{j \neq i} \{L_j(t)\}$, given that $C_i = C^*$, the probability, $P(X_i < Y_i)$ can be determined accordingly:

$$P(X_i < Y_i) = \int_{-\infty}^{\infty} F_{X_i|Y_i=z}(z) f_{Y_i}(z) dz \quad (20)$$

where f_{Y_i} is the pdf of Y_i and $F_{X_i|Y_i}$ is the conditional cumulative distribution function (cdf) of X_i given Y_i (Appendix B–C).

For a dynamic stopping algorithm, t_s is a random variable with its maximum value capped by the data collection limit, $t_{\max}$. Deriving the accuracy for a truncated Bayesian DS algorithm from an equivalent truncated test in the likelihood ratio domain requires analyzing the CLLRs assuming that data collection stops at any value in the range, $1 \leq t_s \leq t_{\max}$, with an associated probability for each stopping time (4a). This requires numerical analysis [29], which can be computationally expensive if $t_{\max}$ is large.

For simplicity, we will derive the accuracy for a Bayesian algorithm with a static stopping (SS) criterion that has the same data truncation limit, i.e. $t_s = t_{\max}$. With a dynamic stopping

algorithm, i.e. when a threshold probability is imposed, not all of the flash groups defined within $t_{\max}$ may necessarily be illuminated prior to stopping data collection. However, we hypothesize that estimating the accuracy for a Bayesian SS algorithm can still provide a good estimate for the underlying system accuracy of a truncated Bayesian DS algorithm with the same data collection limit:

$$A_{ss} = \sum_{i=1}^M P \left(\max_{j \neq i} \{L_j(t_{\max})\} < L_i(t_{\max}) | C_i = C^* \right) P(C_i = C^*) \quad (21)$$

For a given stimulus paradigm, if the base pattern is replicated for a number of sequences, s , we can alternatively determine accuracy based on the average of the CLLR across sequences or equivalently the sequence average of the OLLR:

$$\begin{aligned} \bar{L}_m(T_b) &= \frac{1}{s} \sum_{n=1}^s L_{m,n}(T_b) \\ &= \sum_{\tau=1}^{T_b} \frac{1}{s} \sum_{n=1}^s l_{m,n}(\tau) \\ &= \sum_{\tau=1}^{T_b} \bar{l}_m(\tau) \end{aligned} \quad (22)$$

where T_b is the number of flashes in the base pattern of a stimulus presentation paradigm; s is the number of sequences; $\bar{L}_m(T_b)$ is the sequence average CLLR for character, C_m ; and $\bar{l}_m(\tau)$ is the sequence average of the OLLR.

3.3.2 Expected Stopping Time—We use (3) to determine the EST for the Bayesian DS algorithm from the equivalent CLLR formulation. Since the BCI eventually selects one of the characters as the user's intended target character, we only need to analyze the rate of convergence of the CLLRs to the stopping threshold given that the target character is one of the M characters. The first step requires determining the average number of times a character is sampled prior to reaching the stopping threshold. Then, taking into account the average interval between a character's presentation, the effective number of stimulus events prior to decision making can be determined.

As mentioned previously, determining the performance functions for a truncated DS algorithm requires recursive numerical analysis. We relax the imposition of a data collection limit to derive the asymptotic lower bound of the EST for an untruncated Bayesian DS algorithm. However, a character's OLLRs, $[l_m(1), l_m(2), \dots, l_m(t)]$, are i.i.d. only for the target character and non-target characters that are not in the same flash group as the target character. Nonetheless, under certain stimulus paradigm conditions where the CLLRs, $\{L_m(t)\}_{m=1}^M$, are statistically independent, such as with the RCP, by conditioning on a specific target character, the OLLRs for each character can be assumed to be i.i.d. This simplifies the derivation of the EST by exploiting Wald's equation, in a manner similar to (16b).

4 Methods

There are two methods of obtaining the performance measures for the P300 speller: analytically by analyzing the character cumulative response function or via Monte Carlo simulations. We describe both methods here for the Bayesian algorithm.

4.1 Analytical Estimation

4.1.1 Row-column paradigm—For the RCP, the BCI correctly selects the target character if the following event occurs: {mean CLLR of the target row $> R-1$ mean CLLRs of the non-target rows} $\cap$ {mean CLLR of the target column $> C-1$ mean CLLRs of the non-target columns}. Due to this property, we can analyze the performance for selecting a target row separately from that of selecting the target column, and combine accordingly to derive the overall performance measure [22]. Due to symmetry of the RCP, we only need to determine a performance measure based on considering one character as the target character, instead of averaging across all characters, as in (21). Also, since the row/column flash groups are pairwise disjoint, the CLLR associated with each flash group during row/column selection are independent of each other. This property leads to tractable solutions for the performance functions of the RCP.

(i) Accuracy: Determining the system accuracy for the RCP requires evaluating the probability that the BCI correctly selects both the target row and the target column. The projected accuracy, A_{pr} , can be determined according to:

$$A = p_r \times p_c = \int_{-\infty}^{\infty} (F_{\bar{U}}(z))^{R-1} f_{\bar{T}}(z) dz \times \int_{-\infty}^{\infty} (F_{\bar{V}}(z))^{C-1} f_{\bar{T}}(z) dz \quad (23)$$

where p_r denotes the probability of correctly selecting the target row; p_c denotes the probability of correctly selecting the target column; $\bar{T}$ denotes the average CLLR for the target row/column; $\bar{U}$ denotes the average CLLR for the non-target rows/columns; R and C denote the number of rows and columns, respectively, in the grid; and $F_X(\cdot)$ and $f_X(\cdot)$ denote the cdf and pdf, respectively, for the random variable X .

The accuracy can be approximated given the grid size, the pdfs of the non-target and target classifier scores, and the maximum number of sequences. The pdfs of $I_m(\tau)$ can be derived from the pdfs of the target, $p(y|H_1)$, and the non-target, $p(y|H_0)$, classifier scores. The pdfs of $\bar{T}$ and $\bar{U}$ can be derived accordingly via numerical convolution of the pdfs of $I_m(\tau)$ under the target and non-target hypothesis cases, respectively.

If the classifier scores are normally distributed, we use the pdf of $I_m(\tau)$ specified by the detectability index (18) to derive the pdfs for $\bar{T}$ and $\bar{U}$. Algorithm 1 outlines pseudocode to obtain the accuracy with the Bayesian SS algorithm for the RCP, quantified by the detectability index under a normality assumption.

Algorithm 1

Analytical estimation of accuracy with the Bayesian static stopping[†] algorithm for the row-column paradigm.

Input: d = detectability index; R = number of rows; C = number of columns, s = maximum number of sequences

Output: A = Projected accuracy

```

1 Function rcAccuracy( $d, R, C, s$ )
2  $\mu_0 = -d^2/2$ 
3  $\mu_1 = d^2/2$ 
4  $\sigma = d/\sqrt{s}$ 
5  $\triangleright \Phi(\cdot)$  and  $\phi(\cdot)$  are cdf and pdf, respectively, of the standard normal distribution.
6 foreach  $j = \{r, c\}$  do
7    $p_j = \int_{-\infty}^{\infty} \left\{ \Phi\left(\frac{t-\mu_0}{\sigma}\right) \right\}^{J-1} \frac{1}{\sigma} \phi\left(\frac{t-\mu_1}{\sigma}\right) dt$   $\triangleright J = \{R, C\}$ 
8 end
9 return  $A = p_r \times p_c$ 

```

[†]The above method can also be used to approximate accuracy with a truncated Bayesian dynamic stopping algorithm that has the same data truncation limit, upper-bounded at round the threshold probability, P_{th} .

(ii) Expected stopping time: Similarly, for the EST, the average number of row flashes prior to making a row selection decision, is evaluated separately from the average number of column flashes prior to a column selection decision. The asymptotic lower bound for the EST of the untruncated DS algorithm can easily be derived from the equivalent CLLR formulation of the Bayesian DS algorithm by evaluating M binary SLRTs until a row or column decision is made.

First, the average number of times a row or column is sampled has to be determined, given that the target row or column is always among the flash groups. The CLLR stopping thresholds for the row and column selection process are derived from the Bayesian probability threshold, P_{th} , according to (9)–(10) [34]. The overall sequential test only terminates when the CLLR of a row/column flash group attains the upper stopping threshold, and that flash group is declared as the target row/column. Hence, it is only required to evaluate the average number of times a row/column is sampled prior to being selected, which is equivalent to the ASN for the SLRT given the H_1 hypothesis is true (16b).

Taking into account the number of rows or column flashes within a sequence, the asymptotic lower bound of the expected stopping time, EST_{as} , can be determined:

$$EST_{as} = R[T_r|H_1] + C[T_c|H_1] \quad (24)$$

where R/C is the number of rows/columns in the grid; $\mathbb{E}[T_r|H_1]/\mathbb{E}[T_c|H_1]$ is the average number of times a row/column is sampled, given the target hypothesis is true.

Algorithm 2

Analytical estimation of the asymptotic lower bound of the expected stopping time of the untruncated Bayesian dynamic stopping[†] algorithm for the row-column paradigm.

Input: d = detectability index; R = number of rows; C = number of columns, P_{th} = stopping probability threshold

Output: EST_{as} = Asymptotic lower bound for the expected stopping time

```

1 Function rcpESTAsymptote( $d, R, C, P_{th}$ )
2  $A_0 = A_1 = (1 - P_{th})/P_{th}$ 
3  $\mathbb{E}[\ln \lambda | H_1] = d^2/2$ 
4 foreach  $j = \{r, c\}$  do
5    $\pi_1(j) = 1/J$ 
6    $\pi_0(j) = 1 - \pi_1(j)$ 
7    $A_j^* = (\pi_0(j)A_0)/\pi_1(j)$ 
8    $B_j^* = \pi_0(j)/(A_1\pi_1(j))$ 
9    $\mathbb{E}[T_j | H_1] = \frac{P_{th} \ln B_j^* + (1 - P_{th}) \ln A_j^*}{\mathbb{E}[\ln \lambda | H_1]}$ 
10 end
11 return  $EST_{as} = R\mathbb{E}[T_r | H_1] + C\mathbb{E}[T_c | H_1]$ 

```

[†]The above method can also be used to approximate the expected stopping time with a truncated Bayesian dynamic stopping algorithm, taking into account convergence to the data collection limit, t_{max} .

For the RCP, given the grid size, the pdfs of the non-target and target classifier scores, and the probability threshold, P_{th} , the EST_{as} can be determined accordingly. The pdfs of $I_m(\tau)$ can be derived from the pdfs of the target, $p(y|H_1)$, and the non-target, $p(y|H_0)$, classifier scores. If the classifier scores are normally distributed, we use the detectability index (18) to quantify EST_{as} via $\mathbb{E}[\ln \lambda | H_1]$ (16b). Algorithm 2 outlines pseudo-code to obtain EST_{as} of the untruncated Bayesian DS algorithm for the RCP, quantified by the detectability index under a normality assumption.

(iii) Similar block-randomized paradigms: The properties of the RCP allows for the derivation of performance functions analytically with tractable solutions. These analytical methods can also be applicable to other similar block-randomized stimulus presentation paradigms that satisfy the general properties of the RCP:

1. The stimulus paradigm renders the BCI character selection process in two stages: selection of a flash group from a set of pairwise disjoint flash groups,

$\mathcal{F}^1 = \{\mathcal{F}_i^1\}_{i=1}^R$, followed by selection from another set of pairwise disjoint flash groups, $\mathcal{F}^2 = \{\mathcal{F}_j^2\}_{j=1}^C$. For two flash groups from each set, $\mathcal{F}_i^1$ and $\mathcal{F}_j^2$, if $\mathcal{F}_i^1 \cap \mathcal{F}_j^2 \neq \emptyset$, then $|\mathcal{F}_i^1 \cap \mathcal{F}_j^2| = 1$, $\mathcal{F}_i^1 \cap \mathcal{F}_n^2 = \emptyset$, $\forall n \neq j$, and $|\mathcal{F}_m^1 \cap \mathcal{F}_j^2| = \emptyset$, $\forall m \neq i$. For the RCP, each row shares one common character with only one of the column flash groups.

2. All of the flash groups in each flash group set have the same number of elements, i.e. $|\mathcal{F}_i^1| = K$ and $|\mathcal{F}_j^2| = L$. For the RCP, there are same number of characters in each row or column group.

3. Within each sequence, the flash groups are sampled without replacement. There is no randomization of character-to-flash group assignment after each sequence. For the RCP, the characters in each row or column are fixed, with the order of presentation randomized without replacement after each sequence.

It should also be noted that a P300 speller with single character presentation lends itself to deriving the performance functions analogous to selecting from only one flash group set, i.e. in Algorithm 1–2, $C = 0$.

4.1.2 Generic stimulus paradigm—There are some stimulus paradigms used in the BCI literature that do not satisfy the general characteristics of the RCP, e.g. [10, 36], and therefore, the proposed methods outlined in Algorithms 1–2 for estimating performance functions are not applicable. The projected accuracy for these paradigms can be derived from the general expression in (21) if the flash pattern of the stimulus paradigm is known. For some stimulus paradigms with character-to-flash group randomization during each sequence, (e.g. [10]), (21)–(22) can be used to estimate a lower bound for accuracy, assuming a fixed character-to-flash group assignment.

The projected accuracy can be approximated given the base sequence flash pattern, the pdfs of the non-target and target classifier scores, and the maximum number of sequences. If the classifier scores are normally distributed, the pdf of $\bar{I}_m(\tau)$ can be used to derive the pdfs for the $\bar{L}_m(t)$. Algorithm 3 outlines pseudo-code to approximate the accuracy with the Bayesian SS algorithm for a generic stimulus paradigm, quantified by the detectability index under a normality assumption.

Algorithm 3

Analytical estimation of accuracy with the Bayesian static stopping[†] algorithm for a generic stimulus paradigm.

Input: d = detectability index; $\mathcal{F}$ = sequence flash pattern matrix; s = maximum number of sequences;
 $\{P(C_m = C^*)\}_{m=1}^M$ = prior probability distribution on characters, $\{C_m\}_{m=1}^M$

Output: A = Projected accuracy

```

1 Function paradigmAccuracy( $d, \mathcal{F}, s, \{P(C_m)\}_{m=1}^M$ )
2  $[M, T] = \text{size}(\mathcal{F})$  ▷  $M$  = number of grid characters,  $T$  = number of flashes/sequence
3 for  $i = 1 : M$  do
4   ▷  $C_i$  is the target character. Let  $Y_i = \bar{L}_i(T)$ .
5   for  $j = 1 : M$  do
6      $\bar{I}_j(\tau) = \begin{cases} N(-d^2/2, d/\sqrt{s}), C_i \notin \mathcal{F}_\tau \\ N(d^2/2, d/\sqrt{s}), C_i \in \mathcal{F}_\tau \\ 0, \text{ otherwise} \end{cases}$ 
7      $\bar{L}_j(T) = \sum_{\tau=1}^T \bar{I}_j(\tau)$ 
8   end
9   - Evaluate the covariance matrix,  $[\bar{L}_1(T), \dots, \bar{L}_M(T)]$ .
10  - Evaluate the probability distribution of  $X_i = \max_{j \neq i} \{\bar{L}_j(T)\}$ . ▷ Appendix C
11  - Determine the correlation coefficient  $\rho_{X_i, Y_i}$ . ▷ Appendix C
12   $P(\hat{C}^* = C_i | C_i = C^*) = \int_{-\infty}^{\infty} F_{X_i | Y_i = t}(t) f_{Y_i}(t) dt$  ▷ Appendix B
13 end
14 return  $A = \sum_{m=1}^M P(\hat{C}^* = C_m | C_m = C^*) P(C_m = C^*)$ 

```

[†]The above method can also be used to approximate accuracy with a truncated Bayesian dynamic stopping algorithm that has the same data truncation limit, upper-bounded around the threshold probability, P_{th} .

4.2 Monte Carlo Estimation

Alternatively, performance functions for any stimulus paradigm can be obtained from Monte Carlo (MC) simulations [25] of P300 spelling runs. This is particularly useful in circumstances where a performance function cannot easily be obtained by analytical means or from numerical evaluation of a closed-form expression.

In general, to simulate P300 speller runs, the following elements are required:

- The stimulus paradigm sequence flash patterns, $\mathcal{F}$, which determines the characters in each flash group and the order of flash group presentation.
- The class conditional likelihood functions, $p(y|H_0)$ and $p(y|H_1)$, which determine the random sampling of the classifier scores in the case of a non-target or a target response, respectively.
- The cumulative classifier response function, $\{\Theta_m(t)\}_{m=1}^M$, which quantifies the possibility that each character is the target character, given all of the user's responses.
- A stopping rule, static or dynamic, to terminate data collection, and a decision rule to decide on a character selection, $\hat{C}^*$.

Algorithm 4

Monte Carlo estimation of P300 speller performance with a generic BCI algorithm.

Input: N = Number of iterations, $\{P(C_m = C^*)\}_{m=1}^M$ = prior probability distribution on characters, $\{p(y|H_0), p(y|H_1)\}$ = class conditional classifier score likelihoods, $\mathcal{F}$ = sequence flash pattern, s = maximum number of sequences, (if applicable) Θ_{th} = stopping threshold

Output: $\hat{A}$ = Accuracy, $E\hat{S}T$ = Expected stopping time

```

1 Function simulateP300Speller( $N, \{P(C_m = C^*)\}_{m=1}^M, p(y|H_0), p(y|H_1), \mathcal{F}, s, \Theta_{th}$ )
2  $[M, T] = \text{size}(\mathcal{F})$  ▷  $M$  = number of grid characters,  $T$  = number of flashes/sequence
3  $t_{\max} = s \times T$ 
4 for  $i = 1 : N$  do
5   - Randomly draw from  $\{P(C_m = C^*)\}_{m=1}^M$  for a target character,  $C_i^*$ .
6   - Generate new flash pattern matrix,  $\mathbb{F}$ , of size  $[M, t_{\max}]$ , by replicating and randomizing  $\mathcal{F}$  according to the stimulus paradigm randomization rules.
7   - Initialize character cumulative classifier response function,  $\{\Theta_m(0)\}_{m=1}^M$ .
8    $t = 1$ 
9   while  $\Theta_{\max}(t) < \Theta_{th}$  and/or  $t \leq t_{\max}$  do
10    - Sample flash group,  $\mathbb{F}_t$ .
11    - Randomly draw classifier score,  $y_t \sim \begin{cases} p(y_t|H_0), C^* \notin \mathbb{F}_t \\ p(y_t|H_1), C^* \in \mathbb{F}_t \end{cases}$ .
12    - Update  $\{\Theta_m(t)\}_{m=1}^M$  according to algorithm update rules.
13     $t = t + 1$ 
14  end
15   $\hat{C}_i^* = \underset{m}{\operatorname{argmax}} \{\Theta_m(t)\}$ 
16   $t_{s,i} = t$ 
17 end
18 return  $\hat{A} = \frac{1}{N} \sum_{i=1}^N \text{equal}(\hat{C}_i^*, C_i^*)$ ,  $E\hat{S}T = \frac{1}{N} \sum_{i=1}^N t_{s,i}$ 

```

Algorithm 4 outlines pseudo-code for simulating P300 speller character selection to estimate accuracy and the expected stopping time.

4.3 Datasets

In general, two types of data were used in this study. We used synthetic data generated from MATLAB® (specifically `normrnd()` function) to perform simulations for initial verification of our performance prediction method. We also used empirical data obtained from our previous online P300 speller studies where the Bayesian dynamic stopping algorithm was implemented [7, 18, 19], to validate our performance prediction method with online BCI use.

For the P300 speller studies, in an experiment session, the BCI evaluation protocol consisted of three steps: (i) a calibration run where EEG data was collected to train the BCI algorithm, (ii) algorithm parameter estimation from the training data (e.g. electrode channel selection, classifier vector estimation, classifier pdf estimation, etc.), and (iii) an online test run which involved copy-spelling tasks using the P300 speller with the estimated parameters. The online performance measures, i.e. accuracy and expected stopping time, were estimated from the P300 speller outcomes during the test run. All the studies utilized electrodes {Fz, Pz, Cz, P3, PO7, P4, PO8, Pz, Oz} for signal analysis, with pre-processing and classification performed according to [37]. A summary of the study differences is provided in Table 1:

5 Results

5.1 Verification of Theoretical Model

We performed MC simulations (Algorithm 4) with synthetic data to verify the performance functions obtained analytically from the equivalent likelihood ratio formulation of the Bayesian DS algorithm. We describe a performance measure, Γ_j^i , where $\Gamma = \{A, EST\}$ is the type of performance measure; $i = \{a, s\}$ denotes whether the measure was determined analytically (a) or via simulations (s); and $j = \{ss, tds, uds, as\}$, denotes the type of Bayesian algorithm, static stopping (ss), truncated DS (tds) and untruncated DS (uds) algorithms, and the asymptotic lower bound of the untruncated DS test (as).

The performance measures obtained analytically, A_{ss}^a and EST_{as}^a , were determined from Algorithm 1 or 3, and 2, respectively. We performed simulations of P300 spelling runs for a given stimulus paradigm, assuming classifier scores that were normally distributed, randomly drawn according to `normrnd()` function in MATLAB®. Simulations were performed over a range of detectability indices, with the classifier pdf parameters, $(\mu_0, \sigma_0^2, \mu_1, \sigma_1^2)$, specified accordingly based on a given detectability index value (17).

In general, three types of simulations were performed:

- Simulation I: The Bayesian SS algorithm with a decision rule,
 $\hat{C}^* = \underset{m}{\operatorname{argmax}} P_m(t_{\max})$, to determine accuracy, A_{ss}^s .
- Simulation II: The truncated Bayesian DS algorithm with a decision rule,
 $\hat{C}^* = \underset{m}{\operatorname{argmax}} P_m(t_s) \geq P_{th}$ at $1 \leq t_s \leq t_{\max}$, to determine accuracy, A_{tds}^s and expected stopping time, EST_{tds}^s .

- Simulation III: The untruncated Bayesian DS algorithm with a decision rule,

$$\hat{C}^* = \underset{m}{\operatorname{argmax}} P_m(t_s) \geq P_{th} \text{ at } 1 \leq t_s \leq t_{\max}, \text{ to determine } A_{uds}^s \text{ and } EST_{uds}^s.$$

Figures 4 and 5 show the results for the accuracy and the expected stopping time, respectively, for the RCP with a 9×8 grid. Figure 4(a) compares the projected accuracy obtained analytically, A_{ss}^a (Algorithm 1), and the accuracy obtained by simulations, A_{ss}^s , for the Bayesian SS algorithm. The results from the simulations match the predicted accuracies, as A_{ss}^s is more or less identical to A_{ss}^a (mean squared error, $MSE = 2.1 \times 10^{-5}$). This confirms that the P300 speller with the Bayesian algorithm can be modeled with an equivalent likelihood ratio formulation, and in the likelihood domain, performance can conveniently be parameterized by the detectability index.

Figure 4(b) compares the effect on accuracy of imposing a threshold probability ($P_{th} = 0.9$) on the Bayesian algorithm to achieve dynamic stopping, with or without a data collection limit. The accuracy of the truncated DS algorithm, A_{tds}^s , is still very similar to the analytical estimate of the Bayesian SS algorithm, A_{ss}^a ($MSE = 1.59 \times 10^{-2}$). In general, accuracy tends to improve with increased data collection. Hence, there is a potential loss in accuracy that can result with a dynamic stopping algorithm due to the varied stopping time, $1 \leq t_s \leq t_{\max}$, as opposed to using all of the data allowed by the data collection limit, $t_s = t_{\max}$. At high detectability indices, there are deviations where $A_{tds}^s < A_{ss}^a$, since it is highly likely that on average, the probability threshold will be attained at $t_s < t_{\max}$.

The accuracy of the truncated DS algorithm at a given detectability index is upper bounded by the accuracy of the untruncated DS algorithm that is usually equal to or greater than the threshold probability, P_{th} . From Figure 4(b), it can be observed that A_{uds}^s defines the maximum value attainable by A_{tds}^s ; specifically, $A_{ss}^a = A_{tds}^s$ if $A_{ss}^a \leq P_{th}$, and $A_{ss}^a > A_{tds}^s$ if $A_{ss}^a > P_{th}$. The range of detectability indices over which the difference, $|A_{ss}^a - A_{tds}^s| \leq 1 - P_{th}$, is appreciable is minimized if P_{th} is closer to 1. Hence, the accuracy with the Bayesian SS algorithm can be used to approximate that of a truncated Bayesian DS algorithm that has the same data collection limit, with the accuracy of the truncated algorithm upper-bounded by the stopping probability threshold.

Figure 5 shows the corresponding EST of the untruncated and truncated Bayesian DS algorithms using the RCP, to verify the EST determined analytically, EST_{as}^a (Algorithm 2). The EST of the untruncated DS algorithm, EST_{uds}^s , is asymptotically lower-bounded by EST_{as}^a , as $EST_{as}^a < EST_{uds}^s$. From the simulations, EST_{tds}^a , more or less follows EST_{uds}^s , but starts to deviate as it approaches and converges to the data collection limit. These results indicate that EST_{as}^a provides an analytic expression for the lower bound of the EST of the untruncated DS algorithm, from which we can infer the EST of the truncated DS algorithm with a given data truncation limit.

For some stimulus paradigms in the BCI literature that do not satisfy the properties to use the analytical solutions for the performance estimates for the RCP, e.g. the checkerboard paradigm (CBP) [10], we proposed a method for estimating accuracy for generic stimulus paradigms. We compared the projected accuracy, determined analytically using Algorithm 3,

A_{ss}^a , to the accuracy determined via simulations of the Bayesian SS algorithm, for the RCP and CBP. Both paradigms were implemented with a 9×8 grid. For the CBP as specified in [10], there is character-to-flash group randomization after each sequence. However, for this analysis, we will assume that the base stimulus pattern for the CBP is replicated based on the number of sequences, with no character-to-flash group randomization. This is required to compare the analytical- and simulated-based outcomes in order to verify the methodology proposed in Algorithm 3.

Figure 6 compares the analytical and simulation results for the RCP and a non-randomized CBP. For both paradigms, it can be observed that A_{ss}^s closely follows A_{ss}^a , with some noticeable amount of deviation, although the difference reduces with an increased number of sequences. For the RCP, the errors for the Bayesian SS algorithm are higher with Algorithm 3 that uses the Clark approximation [24] (Figure 6(a), $MSE = 7.1 \times 10^{-2}$), compared to Algorithm 1 (Figure 4(a), $MSE = 2.1 \times 10^{-5}$) where the cdf of the maximum of independent random variables is determined exactly. Overall, these results provide verification for the methodology proposed in Algorithm 3 to predict performance trends for stimulus paradigms that do not satisfy the properties of the RCP.

5.2 Validation with BCI Datasets

We have demonstrated that the detectability index can be used to estimate P300 speller performance with the Bayesian algorithm under a normality assumption. We proceed to validate the proposed method using empirical data from online BCI studies where the truncated Bayesian DS algorithm was implemented, summarized in Table 1.

In all of the studies, the class conditional pdfs used in the Bayesian DS algorithm [18] were obtained using kernel density estimation (KDE) rather than estimated to be Gaussian. For non-parametric distributions, such as would be derived using KDE, derivation of the performance functions analytically involves numerical recursive convolutions and integration of pdfs and cdfs. The projected performance measures can still be obtained via MC simulations. However, without the normality assumption, performance cannot be parameterized with the detectability index, making it inconvenient for performance visualization and analysis.

The differences between performance estimates obtained with non-parametric probability distributions and those obtained with a normality assumption may be negligible if the non-parametric distributions closely resemble normal distributions. Colwell *et al.* [22] demonstrated that the accuracy with P300 speller simulations using bootstrapped classifier scores (equivalent to non-parametric estimation of the pdfs), converged to within ± 0.01 of the projected accuracy determined analytically based on normal distribution parameters estimated from the randomly drawn classifier scores. This suggests that the pdfs obtained via KDE closely model normal distributions. Since the finite space pdfs obtained via KDE closely model normal distributions, we hypothesize that the detectability index can still be used to approximate user performances.

Previously, the analytical methods for estimating projected accuracy and expected stopping time were validated with synthetic data using simulations with normally distributed classifier

scores drawn according to `normrnd()` function in MATLAB®. With parametric probability distributions, there is the convenience that the performance functions can be precomputed and later used to make performance predictions based on a user's classifier likelihood parameters. We will now refer to performance functions obtained with a normality assumption as the *predicted* accuracy, A_{pr} and expected stopping time, EST_{pr} . In this section, our performance prediction method is validated by comparing the predicted performances based on a user's detectability index, to those obtained from participant data from our online BCI studies. We will denote results obtained with participant data as the *observed* performances, A_{obs} and EST_{obs} .

5.2.1 Simulations with bootstrapped data—We performed simulations with participant data from Throckmorton *et al.* [18] to provide an estimate of a user's long-term performance (assuming no change in data statistics with time in the test run) for a fairer comparison with our simulations using synthetic data, in terms of the same number of observations used to estimate the performance measures. First, the classifier score pdfs, $p(y|H_0)$ and $p(y|H_1)$, to be used for the probability update process in the Bayesian DS algorithm, were developed via KDE from the training data obtained from the calibration run of a participant's BCI session. P300 spelling simulations were performed by bootstrapping classifier scores obtained from processing EEG data from the test run of the same session. Detectability indices of the training and test data were computed from the respective data class conditional classifier scores.

Figure 7(a) shows a comparison of the observed and predicted performances based on detectability indices computed from the training data, as would be expected when making predictions during real-world BCI use. Even though EST_{obs} closely follows EST_{pr} it can be observed that A_{pr} generally tends to overestimate A_{obs} . Some of this mismatch may be due to a drift in the data statistics from training to testing data. As a second comparison, detectability indices obtained from the test data were used, providing a close match between results obtained with normally distributed classifier scores according to the detectability index and results obtained with bootstrapped data. As can be seen in Figure 7(b), the observed performances are closer to those predicted by d of the test data.

These results indicate that the detectability index can be used to estimate performance, confirming the pdfs obtained via KDE closely approximate normal distributions. The results also highlight the limitations of using training data to predict BCI performance, as the model that best fit the data, d for the test data, provided a better estimate of the observed performances. Nonetheless, the training data can provide useful initial information about a user's performance level.

5.2.2 Online performances—In the previous section, the observed participant performances were computed from simulations that relied on participant EEG data. In this section, we analyze the reported online performances from the following studies: Throckmorton *et al.* [18], Mainsah *et al.*, 2014 [19] and Mainsah *et al.*, 2015¹ [7]. Since

¹The participants underwent three BCI sessions on different days. The results from each session were analyzed separately.

these data are intended for validation, the detectability index, d , for each participant session was computed from the classifier scores of the test data.

Figure 8 shows the results for both non-disabled participant studies [18, 19], and Figure 9 shows the results for the ALS participant study [7]. Unlike the simulations with bootstrapped data, the online performances were estimated from a small number of samples (see Table 1). Consequently, the online performance estimates have low resolution and exhibit a higher degree of variation. Nonetheless, it can be observed qualitatively that the trends of the observed performances follow those predicted by the participant detectability indices for the two stimulus presentation paradigms at the respective sequence limits. The results are a final confirmation that the detectability index can provide a useful means to assess BCI performance without the need for extensive online testing.

6 Discussion

We have developed a new BCI performance prediction method based on a probabilistic model of the operation of a generic BCI. Most of the previous approaches to predict BCI performance used regression models that relied on noisy empirical estimates [21, 20]. Consequently, derivation of performance functions with these previous methods required a large user database to minimize model variability. With our method, performance functions can be obtained either analytically or via Monte Carlo simulations. With parametric models, the performance functions can be precomputed without the need for BCI data. Another advantage of our generic model-based approach is that it is flexible to account for different stimulus presentation paradigms, as well as static and dynamic stopping algorithms. In this study, we demonstrated the applicability of our performance prediction method with the Bayesian dynamic stopping algorithm [18]. For convenience, we simplified the P300 speller from an M -ary to a binary hypothesis problem by substituting the probability domain with an equivalent likelihood ratio formulation to derive the performance functions.

Under conditions of normality, we showed that the detectability index represents a statistical measure that can be used to quantify a user performance level with the Bayesian stopping algorithm. In simplifying the analytic derivation for accuracy, some amount of error may result from using the Clark approximation [24], or approximating a truncated dynamic stopping algorithm with a static stopping algorithm. Errors from using the Clark approximation arise as a result of approximating the pdf of the maximum order statistic of correlated random variables with a normal distribution, whereas the actual pdf might be skewed [38]. Some methods have been proposed to reduce the Clark approximation error [38, 39, 40, 41]. From the asymptotic lower bound of the untruncated dynamic stopping algorithm, the expected stopping time of the truncated dynamic stopping algorithm can be inferred, after considering convergence to the data truncation limit. Our performance prediction method using the detectability index was validated with data from our previous online studies, using participants with and without disability, and with different stimulus presentation paradigms.

The probabilistic model representation of a BCI presented in section 2 can be applied to other BCI algorithms to derive their respective performance functions, either analytically,

where tractable using (3)–(4), or via Monte Carlo simulations. Our Bayesian dynamic stopping algorithm is similar to other algorithms used in the BCI literature, e.g. [42, 43], and so our performance prediction method using the detectability index can be relevant for these algorithms. Other algorithms may require the development of alternative statistical measures to quantify performance levels, as the use of detectability index parameter might not always be suitable. For example, under conditions of normality, a user's performance level with the Colwell *et al.* [22] performance prediction method is represented by a 4D vector of the classifier likelihood parameters, $[\mu_0, \sigma_0, \mu_1, \sigma_1]$; however, if $\sigma_0 \simeq \sigma_1$, the detectability index measure can be used for simplicity.

The ability to predict performance provides a quick means to pre-assess BCI performance prior to online use and optimize parameter selection to improve a user's performance. For example, it is useful when deciding on the data truncation limit to quantify the effect of the amount of data collection given a user's performance level. From Figure 4(b), it can be observed that the minimum accuracy specified by the threshold probability of the Bayesian DS algorithm is achieved at different data collection limits across detectability indices. Unsurprisingly, the amount of data collection has little effect at very low and very high detectability indices. Users with low performance levels usually have minimally distinct classifier likelihoods (low d) and represent those who cannot use the BCI system efficiently due to the large amount of data collection needed to achieve acceptable accuracy levels. Users with high performance levels (high d) require only a small amount of data to use the BCI system effectively. The impact of the data collection limit on accuracy is most noticeable at the mid-range detectability indices. Hence, for users with mid-range performance levels, an appropriate data collection limit can significantly increase their maximum attainable accuracy levels.

Another useful application of our probabilistic model approach to performance prediction is that it can be used as an offline analysis tool to potentially explore BCI algorithm parameter spaces when empirical data is not available or might be cumbersome to collect. For some BCI algorithm parameters or strategies, it is easier to do a grid search over parameter values to maximize performance of a given user dataset, as is typically done in the BCI literature for exploratory analysis, e.g. electrode channel selection [44], classifier type [45], dynamic stopping algorithms [42], etc. However, it is not possible to use such an approach for analyzing different stimulus presentation paradigms. While alternative stimulus presentation paradigms have been proposed to improve performance, e.g. [10, 46], these stimulus paradigms are usually constructed based on a set of rules using criteria where potential performance improvements are inferred rather than evaluated objectively. Sometimes, the stimulus presentation paradigms developed with these adhoc rules might not be the optimum or best configuration, e.g. [46]. With the Bayesian dynamic stopping algorithm, the presentation order of non-target and target stimulus events directly affects the rate of algorithm convergence to the decision threshold due to the character likelihood assignment process (6). The availability of analytical methods to predict performance allows for a time-efficient and principled method to explore an exponentially large number of stimulus presentation paradigms without the need for extensive online data collection.

There are shortcomings to using only information from the training data to predict performance. Predictions based on a given dataset may not be applicable to subsequent data collections depending on the significance of the shift in the data statistics. From our simulation results in Figure 7, the predicted performances determined based on detectability indices estimated from the training data tended to overestimate the observed performances obtained from the test data. Nonetheless, the training data provides some useful prior information to ballpark a user's performance level. A probabilistic model allows for the flexibility to account for shifts in data statistics from calibration. The detectability index measure allows for easy visualization of the possible range in a user's performance level over various data collection limits. This information when coupled with the knowledge that the performance curves may be an upper bound, can provide guidance in parameter selection, such as setting an appropriate data collection limit for a dynamic stopping algorithm.

Most BCI studies typically estimate performance with online use from averaging over a low number of samples, like copy-spelling 10 to 40 characters in the case of most BCI speller studies. It is important to be aware of the variance of maximum likelihood estimators (MLE). Depending on the number of samples used, the estimated performance may not necessarily reflect the underlying system performance. For example, A_{obs} follows the MLE of the parameter, p , of a binomial distribution, $n \sim B(N, p)$, where N is the total number of observations, $p = A_{true}$, and A_{true} is the true underlying system accuracy. Hence $A_{obs} = \hat{p}_{MLE} = N_c / N$, where N_c is the number of correctly selected characters or successes. $\hat{p}_{MLE}$ is also a random variable, with mean, $\mathbb{E}[\hat{p}_{MLE}] = p$, and variance, $\text{Var}(\hat{p}_{MLE}) = p(1-p)/N$. Figure 10 illustrates the possible variation that can be exhibited by A_{obs} based on the number of samples used for estimation. The variance of A_{obs} reduces with increased N . In addition, the resolution of the estimate also increases with N . While it is recommended that maximum likelihood estimators be obtained from a large number of samples, this is not always practical in real-world BCI experiment design.

7 Conclusion

We have developed a probabilistic model-based method to predicting BCI performance, an approach which is independent of stimulus presentation paradigm and data stopping criterion. With this method, performance functions for BCI algorithms can be obtained either analytically or via Monte Carlo simulations. We propose a new method to predict BCI performance based on the discriminability of the system's classifier. The detectability index, which quantifies the discriminability of the classes, can be used to estimate the performance bounds of the Bayesian dynamic stopping algorithm for the P300 speller. Our performance prediction method based on the detectability index was validated with results from online BCI studies, including a target BCI end-user population with disability. Our performance prediction framework can provide a useful tool for BCI users to improve performance by using a principled method to optimize parameter selection, such as estimating the amount of data collection needed to achieve a certain accuracy level. Additional work is suggested to apply the probabilistic model-based performance prediction method outlined here to derive performance functions for other BCI algorithms.

Acknowledgments

This research project was funded by the NIH/NIDCD under grant number R33 DC010470. The authors would like to thank Dr. Loren Nolte, Duke University, Durham, NC USA, for his insight on this study.

References

1. Wolpaw JR, Birbaumer N, McFarland DJ, Pfurtscheller G, Vaughan TM. Brain-computer interfaces for communication and control. *Clin Neurophysiol.* 2002; 113(6):767–791. [PubMed: 12048038]
2. Moghimi S, Kushki A, Marie Guerguerian A, Chau T. A review of EEG-based brain-computer interfaces as access pathways for individuals with severe disabilities. *Assist Technol.* 2013; 25(2): 99–110. [PubMed: 23923692]
3. Sutton S, Braren M, Zubin J, John ER. Evoked-potential correlates of stimulus uncertainty. *Science.* 1965; 150(3700):1187–1188. [PubMed: 5852977]
4. Farwell LA, Donchin E. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr Clin Neurophysiol.* 1988; 70(6):510–523. [PubMed: 2461285]
5. Ortner R, Aloise F, Prückl R, Schettini F, Putz V, Scharinger J, et al. Accuracy of a P300 speller for people with motor impairments: a comparison. *Clin EEG Neurosci.* 2011; 42(4):214–218. [PubMed: 22208117]
6. Sellers EW, Ryan DB, Hauser CK. Noninvasive brain-computer interface enables communication after brainstem stroke. *Sci Transl Med.* 2014; 6(257):257re7.
7. Mainsah BO, Collins LM, Colwell KA, Sellers EW, Ryan DB, Caves K, et al. Increasing BCI communication rates with dynamic stopping towards more practical use: an ALS study. *J Neural Eng.* 2015; 12(1):016013. [PubMed: 25588137]
8. Mak JN, Arbel Y, Minett JW, McCane LM, Yuksel B, Ryan D, et al. Optimizing the P300-based brain-computer interface: current status, limitations and future directions. *J Neural Eng.* 2011; 8(2): 025003. [PubMed: 21436525]
9. Kübler A, Holz EM, Riccio A, Zickler C, Kaufmann T, Kleih SC, et al. The User-Centered Design as novel perspective for evaluating the usability of BCI-controlled applications. *PloS ONE.* 2014; 9(12):e112392. [PubMed: 25469774]
10. Townsend G, et al. A novel P300-based brain-computer interface stimulus presentation paradigm: moving beyond rows and columns. *Clin Neurophysiol.* 2010; 121(7):1109–1120. [PubMed: 20347387]
11. Schreuder M, Höhne J, Blankertz B, Haufe S, Dickhaus T, Tangermann M. Optimizing event-related potential based brain-computer interfaces: a systematic evaluation of dynamic stopping methods. *J Neural Eng.* 2013; 10(3):036025. [PubMed: 23685458]
12. Serby H, Yom-Tov E, Inbar GF. An improved P300-based brain-computer interface. *IEEE Trans Neural Syst Rehabil Eng.* 2005; 13(1):89–98. [PubMed: 15813410]
13. Lenhardt A, Kaper M, Ritter HJ. An Adaptive P300-Based Online Brain Computer Interface. *IEEE Trans Neural Syst Rehabil Eng.* 2008; 16(2):121–130. [PubMed: 18403280]
14. Zhang H, Guan C, Wang C. Asynchronous P300-Based Brain-Computer Interfaces: A Computational Approach With Statistical Models. *IEEE Trans Biomed Eng.* 2008; 55(6):1754–1763. [PubMed: 18714840]
15. Jin J, Allison BZ, Sellers EW, Brunner C, Horki P, Wang X, et al. An adaptive P300-based control system. *J Neural Eng.* 2011; 8(3):036006. [PubMed: 21474877]
16. Aloise F, Schettini F, Aricò P, Leotta F, Salinari S, Mattia D, et al. P300-based brain-computer interface for environmental control: an asynchronous approach. *J Neural Eng.* 2011; 8(2):025025. [PubMed: 21436520]
17. Park J, Kim K. A POMDP approach to optimizing P300 speller BCI paradigm. *IEEE Trans Neural Syst Rehabil Eng.* 2012; 20(4):584–594. [PubMed: 22510955]
18. Throckmorton CS, Colwell KA, Ryan DB, Sellers EW, Collins LM. Bayesian approach to dynamically controlling data collection in p300 spellers. *IEEE Trans Neural Syst Rehabil Eng.* 2013; 21(3):508–517. [PubMed: 23529202]

19. Mainsah BO, Colwell KA, Collins LM, Throckmorton CS. Utilizing a Language Model to Improve Online Dynamic Data Collection in P300 Spellers. *IEEE Trans Neural Syst Rehabil Eng.* 2014; 22(4):837–846. [PubMed: 24808413]
20. Thompson DE, Warschausky S, Huggins JE. Classifier-based latency estimation: a novel way to estimate and predict BCI accuracy. *J Neural Eng.* 2013; 10(1):016006. [PubMed: 23234797]
21. Mak JN, McFarland DJ, Vaughan TM, McCane LM, Tsui PZ, Zeitlin DJ, et al. EEG correlates of P300-based brain-computer interface (BCI) performance in people with amyotrophic lateral sclerosis. *J Neural Eng.* 2012; 9(2):026014. [PubMed: 22350501]
22. Colwell K, Throckmorton CS, Collins LM, Morton KD. Projected Accuracy Metric for the P300 Speller. *IEEE Trans Neural Syst Rehabil Eng.* 2014; 22(5):921–925. [PubMed: 25203496]
23. Trivedi, S. Probability & statistics with reliability, queuing and computer science applications. John Wiley & Sons; 2008.
24. Clark CE. The greatest of a finite set of random variables. *Oper Res.* 1961; 9(2):145–162.
25. Kay, SM. Fundamentals of Statistical Signal Processing. Englewood Cliffs, N.J: Prentice-Hall PTR; 2013. Fundamentals of Statistical Signal Processing: Practical algorithm development.
26. Kay, SM. Fundamentals of Statistical Signal Processing, Volume II: Detection Theory. Englewood Cliffs, N.J: Prentice-Hall PTR; 1993.
27. Wald A. Sequential Tests of Statistical Hypotheses. *Ann Math Statist.* 1945; 16(2):117–186.
28. Albert GE. On the computation of the sampling characteristics of a general class of sequential decision problems. *Ann Math Stat.* 1954:340–356.
29. Polyanin, AD., Manzhirov, AV. Handbook of Integral Equations. CRC press; 2008.
30. Simpson AJ, Fitter MJ. What is the best index of detectability? *Psychol Bull.* 1973; 80(6):481–488.
31. Sobel M, Wald A. A Sequential Decision Procedure for Choosing One of Three Hypotheses Concerning the Unknown Mean of a Normal Distribution. *Ann Math Statist.* 1949; 20(4):502–522.
32. Armitage P. Sequential analysis with more than two alternative hypotheses, and its relation to discriminant function analysis. *J R Stat Soc Series B Methodol.* 1950; 12(1):137–144.
33. Eisenberg, B. Multihypothesis Problems. In: Ghosh, BK., Sen, PK., editors. *Handbook of Sequential Analysis.* 1991. p. 229–244.
34. Baum CW, Veeravalli VV. A sequential procedure for multihypothesis testing. *IEEE Trans Inf Theory.* 1994; 40(6):1994–2007.
35. Tantaratana S, Thomas JB. Truncated sequential probability ratio test. *Information Sciences.* 1977; 13(3):283–300.
36. Thomas, E., Clerc, M., Carpentier, A., Daucé, E., Devlaminck, D., Munos, R. Optimizing P300-speller sequences by RIP-ping groups apart. *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on;* 2013; p. 1062–1065.
37. Krusienski DJ, Sellers EW, McFarland DJ, Vaughan TM, Wolpaw JR. Toward enhanced P300 speller performance. *J Neurosci Methods.* 2008; 167(1):15–21. [PubMed: 17822777]
38. Arellano-Valle RB, Genton MG. On the exact distribution of the maximum of absolutely continuous dependent random variables. *Statist Probab Lett.* 2008; 78(1):27–35.
39. Sinha, D., Zhou, H., Shenoy, V. Advances in computation of the maximum of a set of random variables. *Proceedings of the 7th International Symposium on Quality Electronic Design;* 2006; p. 306–311.
40. Nadarajah S, Kotz S. Exact Distribution of the Max/Min of Two Gaussian Random Variables. *IEEE Trans Very Large Scale Integr (VLSI) Syst.* 2008; 16(2):210–212.
41. Shi Y, Xiong J, Zolotov V, Visweswariah C. Order statistics for correlated random variables and its application to at-speed testing. *ACM T Des Automat El(TODAES).* 2013; 18(3):42:1–42:20.
42. Speier W, Arnold C, Lu J, Deshpande A, Pouratian N. Integrating Language Information With a Hidden Markov Model to Improve Communication Rate in the P300 Speller. *IEEE Trans Neural Syst Rehabil Eng.* 2014; 22(3):678–684. [PubMed: 24760927]
43. Kindermans PJ, Tangermann M, Miller KR, Schrauwen B. Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller. *J Neural Eng.* 2014; 11(3): 035005. [PubMed: 24834896]

44. Colwell KA, Ryan DB, Throckmorton CS, Sellers EW, Collins LM. Channel selection methods for the P300 Speller. *J Neurosci Methods*. 2014; 232:6–15. [PubMed: 24797224]
45. Krusienski DJ, Sellers EW, Cabestaing F, Bayouth S, McFarland DJ, Vaughan TM, et al. A comparison of classification techniques for the P300 Speller. *J Neural Eng*. 2006; 3(4):299. [PubMed: 17124334]
46. Verhoeven T, Buteneers P, Wiersema J, Dambre J, Kindermans P. Towards a symbiotic brain–computer interface: exploring the application–decoder interaction. *J Neural Eng*. 2015; 12(6): 066027. [PubMed: 26580120]

Appendix A Random Sums

Let X and Y be two continuous random variables, with probability density functions (pdf), $f_X(x)$ and $f_Y(y)$, respectively. Their sum, $Z = X + Y$ is a random variable with pdf, $f_Z(z)$, where f_Z is the convolution of f_X and f_Y :

$$\begin{aligned} f_Z(z) &= (f_X * f_Y)(z) \\ &= \int_{-\infty}^{\infty} f_X(z-y) f_Y(y) dy \\ &= \int_{-\infty}^{\infty} f_Y(z-x) f_X(x) dx \end{aligned} \quad (25)$$

The pdf of a sum of more than two random variables, $\Omega = X + Y + \dots + Z$, can be obtained via successive convolutions:

$$f_{\Omega}(\omega) = (f_X * f_Y * \dots * f_Z)(\omega) \quad (26)$$

Given a sequence of independent and identical distributed random variables, $Z_1, Z_2, \dots, Z_T$, consider the random sum:

$$S = Z_1 + Z_2 + \dots + Z_T \quad (27)$$

where the number of observations, T is also a random variable. For a fixed value, $T = t$, the conditional expectation of S can be obtained:

$$\begin{aligned} [S|T=t] &= \sum_{i=1}^t [Z_i] \\ &= t[Z] \end{aligned} \quad (28)$$

where $\mathbb{E}[\cdot]$ is the expected value of the random variable. Using the theorem of total expectation, the unconditional expectation of S is:

$$\begin{aligned} [S] &= \sum_t [S|T=t] p_T(t) \\ &= [Z] \sum_t t p_T(t) \\ &= [Z][T] \end{aligned} \quad (29)$$

(29) is referred to as *Wald's equation*.

Appendix B Random Inequalities

Given two random variables, X and Y , with joint probability density function (pdf), $f_{X,Y}(x, y)$, and marginal pdfs, $f_X(x)$ and $f_Y(y)$, respectively, the probability that X is less than Y can be determined accordingly:

$$\begin{aligned} P(X < Y) &= \int_{-\infty}^{\infty} P(X < Y | Y=t) f_Y(t) dt \\ &= \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^t f_{X|Y=t}(t) \right\} f_Y(t) dt \\ &= \int_{-\infty}^{\infty} F_{X|Y=t}(t) f_Y(t) dt \end{aligned} \quad (30)$$

where $F_{X|Y}$ is the conditional cumulative distribution function (cdf) of X given Y .

If X and Y are independent, then their marginal pdfs are independent of each other:

$$P(X < Y) = \int_{-\infty}^{\infty} F_X(t) f_Y(t) dt \quad (31)$$

For example, for a bivariate normal distribution with mean vector, $\mathbf{M}$, and covariance matrix Σ ,

$$\begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathbf{M} = \begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_Y\sigma_X & \sigma_Y^2 \end{bmatrix}$$

The parameters of the conditional pdf, $f_{X|Y}(x|y)$, are:

$$\mu_{X|Y} = \mu_X + \rho \left(\frac{\sigma_X}{\sigma_Y} \right) (Y - \mu_Y) \quad (32a)$$

$$\sigma_{X|Y}^2 = \sigma_Y^2 (1 - \rho^2) \quad (32b)$$

If $\rho = 0$, i.e. X and Y are uncorrelated, hence independent since they are jointly Gaussian, a closed form solution can be obtained:

$$P(X < Y) = \Phi \left(\frac{\mu_Y - \mu_X}{\sqrt{\sigma_Y^2 + \sigma_X^2}} \right) \quad (33)$$

where $\Phi(\cdot)$ is the cdf of the standard normal probability distribution.

Appendix C The Maximum of M Random Variables

Given M random variables, $(X_1, X_2, \dots, X_M)$, the probability that the maximum of the random variables, $\max(X_1, X_2, \dots, X_M)$, is less than or equal to a certain value, z , is equivalent to evaluating the joint cumulative distribution function (cdf) of $(X_1, X_2, \dots, X_M)$ at $\mathbf{1}_{1 \times M}$:

$$P(\max(X_1, X_2, \dots, X_M) \leq z) = P(X_1 \leq z, \dots, X_M \leq z) \quad (34)$$

Alternatively, a random variable can be defined, $Z = \max(X_1, X_2, \dots, X_M)$, with cdf, $F_Z(z) = P(Z \leq z)$.

- If X_i 's are independent, then the cdf of Z is the product of the individual cdfs of the M random variables [23]:

$$F_Z(z) = \prod_{i=1}^M F_i(z) \quad (35)$$

- If the X_i 's are correlated and not necessarily identically distributed, some approaches have been proposed in the literature to approximate the probability distribution of $Z = \max(X_1, X_2, \dots, X_M)$ [24, 38, 39, 40, 41].

The most commonly used approximation for the probability distribution of the maximum of correlated random variables is *Clark's formula* [24]. Given two variables, $X_i \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$, with correlation coefficient, ρ , the distribution of $Z = \max(X_1, X_2)$, can be approximated with another Gaussian, $Z \sim \mathcal{N}(\mu_Z, \sigma_Z^2)$:

$$a = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2} \quad (36a)$$

$$\alpha = (\mu_1 - \mu_2)/a \quad (36b)$$

$$\mu_Z = \mu_1\Phi(\alpha) + \mu_2\Phi(-\alpha) + a\phi(\alpha) \quad (36c)$$

$$\sigma_Z^2 = (\mu_1^2 + \sigma_1^2)\Phi(\alpha) + (\mu_2^2 + \sigma_2^2)\Phi(-\alpha) + (\mu_1 + \mu_2)a\phi(\alpha) - \mu_Z^2 \quad (36d)$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are probability density function (pdf) and cdf values, respectively, evaluated using the standard normal distribution. Extending from (36), the correlation

coefficient of the maximum of two random variables, $\max(X_1, X_2)$, and another random variable, X_3 is:

$$\rho(X_3, \max(X_1, X_2)) = \frac{\sigma_1 \rho_{1,3} \Phi(\alpha) + \sigma_2 \rho_{2,3} \Phi(-\alpha)}{\sigma_Z} \quad (37)$$

where $\rho_{i,j}$ is the correlation coefficient between X_i and X_j .

Hence, given a set of normally distributed random variables, the parameters of the probability distribution of $\max(X_1, X_2, \dots, X_M)$ can be calculated recursively:

$$\begin{aligned} \max(X_1, X_2, \dots, X_M) &= \max(X_M, \max(X_1, \dots, X_{M-1})) \\ &= \max(X_M, \max(X_{M-1}, \max(X_1, \dots, X_{M-2}))) \end{aligned} \quad (38)$$

Appendix D The Truncated Sequential Likelihood Ratio Test

The truncated sequential likelihood ratio test (SLRT) [35] is similar to the untruncated SLRT, but for an imposed data collection limit, $t_{\max}$, and a new decision rule at $t_{\max}$:

$$\text{For } t < t_{\max}, \Lambda_t \begin{cases} \leq \mathcal{A} \Rightarrow \text{choose } H_0, \text{ terminate} \\ \geq \mathcal{B} \Rightarrow \text{choose } H_1, \text{ terminate} \\ \in (\mathcal{A}, \mathcal{B}) \Rightarrow \text{take another sample} \end{cases} \quad (39a)$$

$$\text{For } t = t_{\max}, \Lambda_t \begin{cases} < \Gamma \Rightarrow \text{choose } H_0 \\ \geq \Gamma \Rightarrow \text{choose } H_1 \end{cases} \quad (39b)$$

where $\mathcal{A}$ and $\mathcal{B}$ are defined according to (13), and Γ is a threshold value to decide on a hypothesis at $t = t_{\max}$ with $\mathcal{A} < \Gamma < \mathcal{B}$.

To evaluate the performance of the truncated SLRT conditioned on a hypothesis, H_b , the

probability density function (pdf) of the sum, $\ln \Lambda_t = \sum_{\tau=1}^t \ln \lambda_{\tau}$, has to be evaluated, given that $\ln \mathcal{A} < \ln \Lambda_n < \ln \mathcal{B}$ for $n = 1, \dots, t-1$. For a given hypothesis, H_b , let $P_{\lambda}(z, t)$ denote the cdf probability, $P(\ln \Lambda_t \leq z, \ln \mathcal{A} < \ln \Lambda_n < \ln \mathcal{B}, n = 1, \dots, t-1)$, with the corresponding pdf, $p_{\lambda}(z, t)$. Successive convolutions are required to determine the pdf, $p_{\lambda}(z, t)$:

$$p_i(z, 1) = g_i(z) \quad (40a)$$

$$p_i(z, t) = \int_a^b p_i(u, t-1) g_i(z-u) du, \quad t > 1 \quad (40b)$$

where $g_f(z)$ is the pdf of $\ln \lambda$ under a given a hypothesis, H_i , i.e. $p(\ln \lambda|H_i)$; $a = \ln \mathcal{A}$ and $b = \ln \mathcal{B}$.

The performance functions for the truncated SLRT can then be derived:

$$\begin{aligned} tr[T|H_i] &= \sum_{t=1}^{t_{\max}} tP(T=t) \\ &= \sum_{t=1}^{t_{\max}-1} t \left\{ \int_{-\infty}^a p_i(z, t) dz + \int_b^{\infty} p_i(z, t) dz \right\} + t_{\max} \int_a^b p_i(z, t_{\max}-1) dz \end{aligned} \quad (41)$$

$$P("H_0"|H_i) = \sum_{t=1}^{t_{\max}-1} \left\{ \int_{-\infty}^a p_i(z, t) dz \right\} + \int_{-\infty}^{\ln \Gamma} p_i(z, t_{\max}) dz \quad (42a)$$

$$P("H_1"|H_i) = \sum_{t=1}^{t_{\max}-1} \left\{ \int_b^{\infty} p_i(z, t) dz \right\} + \int_{\ln \Gamma}^{\infty} p_i(z, t_{\max}) dz \quad (42b)$$

where $\mathbb{E}^u[T|H_i]$ is the average number of observations, given that H_i is the true hypothesis; $P("H_j"|H_i)$ is the probability of deciding H_j given that the true hypothesis is H_i .

Figure 1.

Schematic of the target character estimation process of an ERP-based BCI. A user focuses on a target character, C^* , from a set of choices, $\{C_m\}_{m=1}^M$. The BCI presents a stimulus to a user, $\mathcal{F}_t$, and the user's response is scored with the system's classifier. The classifier score, y_t , is used to update a character cumulative response function (CCRF), $\{\Theta_m(t)\}_{m=1}^M$, which quantifies the possibility of each of the BCI choices to be the user's target character. Data collection is continued until it is terminated at a stopping time, t_s , when the maximum CCRF value, $\Theta_{\max}(t)$, attains a threshold value, Θ_{th} , or the data collection limit, $t_{\max}$, is reached. The character that maximizes the CCRF at t_s is selected as the user's intended target character, $\hat{C}^*$.

Figure 2.

Illustrative example of the class conditional probability distribution parameters, (μ_0, σ_0^2) and (μ_1, σ_1^2) , used to determine the detectability index of two normal distributions.

Figure 3.

Role of the classifier score probability density functions (pdfs) during the probability update process in the Bayesian dynamic stopping algorithm. The pdfs for non-target and target classifier scores represent the H_0 and H_1 hypotheses, respectively. A flashed subset, $\mathcal{F}$, that contains the target character, C^* , should elicit a P300, resulting in a higher classifier score such that $p(y/H_1) > p(y/H_0)$. A flashed subset that does not contain the target character should not elicit a P300, resulting in a lower score such that $p(y/H_1) < p(y/H_0)$.

Figure 4.

Accuracy, A , vs. detectability index, d , for the Bayesian static stopping (SS) and dynamic stopping (DS) algorithms for a 9×8 row-column paradigm. P300 spelling runs were simulated assuming normally distributed classifier scores (normrnd() in MATLAB®) specified according to the detectability index, and with sequence limit, s , and data collection limit, $t_{\max} = s \times (9 + 8)$ flashes, if applicable. (a) Comparison of the accuracy obtained analytically, A_{ss}^a (Algorithm 1), to that determined via simulations, A_{ss}^s , for the Bayesian SS algorithm. (b) Comparison of A_{ss}^a to that determined via simulations for the truncated, A_{tds}^s , and untruncated, A_{uds}^s , Bayesian DS algorithms, both with a stopping probability threshold, $P_{th} = 0.9$.

Figure 5.

Expected stopping time, EST, vs. detectability index, d , for the Bayesian dynamic stopping (DS) algorithm with a stopping probability threshold, $P_{th} = 0.9$, for a 9×8 row-column paradigm. P300 spelling runs were simulated assuming normally distributed classifier scores (normrnd() in MATLAB) specified according to the detectability index, and with sequence limit, s , and data collection limit, $t_{\max} = s \times (9+8)$ flashes, if applicable. EST_{as}^a = asymptotic lower bound for untruncated Bayesian DS determined analytically (Algorithm 2); EST_{uds}^s = untruncated Bayesian DS determined via simulations; and EST_{tds}^s = truncated Bayesian DS determined via simulations, with data collection limit, $t_{\max}$.

Figure 6.

Accuracy, A , vs. detectability index, d , for the Bayesian static stopping (SS) algorithm, using row-column (RCP) and checkerboard (CBP) paradigms, both implement with a 9×8 grid. P300 spelling runs were simulated assuming normally distributed classifier scores (normrnd() in MATLAB®) specified according to the detectability index, and with sequence limit, s , and data collection limit, $t_{\max} = s \times 17$ flashes for the RCP and $t_{\max} = s \times 24$ flashes for the CBP. A_{ss}^a = projected accuracy determined analytically (Algorithm 3); and A_{ss}^s = projected accuracy determined via simulations.

Figure 7.

Performance vs. detectability index, d , obtained from simulating Bayesian dynamic stopping using participant data, with a 9×8 row-column paradigm, probability threshold, $P_{th} = 0.9$ and sequence limit, s . For each participant, the classifier score pdfs for the Bayesian probability update process were developed via kernel density estimation using the training data. P300 spelling runs were simulated with bootstrapped classifier scores obtained from the test run. The predicted performances, accuracy, A_{pr} and expected stopping time, EST_{pr} , were obtained from another set of simulations assuming normally distributed classifier scores (`normrnd()` in MATLAB[®]), specified according to the detectability index. (a) Comparison of the observed performances obtained from the simulations using participant test data, A_{obs} and EST_{obs} , to that predicted by d of the training data. (b) Comparison of the observed performances to that predicted by d of the test data.

Figure 8.

Performance vs. detectability index, d , obtained from online studies with the Bayesian dynamic stopping, using a 9×8 row-column paradigm, a probability threshold, $P_{th} = 0.9$ and a sequence limit of 10. The predicted accuracy, A_{pr} , and expected stopping time, EST_{pr} , were obtained from simulations assuming normally distributed classifier scores (normrnd()) in MATLAB®, specified according to the detectability index. The observed performances, A_{obs} and EST_{obs} , are the reported results from Throckmorton *et al.* (T2013) [18] and Mainsah *et al.* (M2014) [19]. The left plot compares A_{pr} and A_{obs} . The right plot compares EST_{pr} and EST_{obs} .

Figure 9.

Performance vs. detectability index, d , obtained from online studies with the Bayesian dynamic stopping, using a 6×6 checkerboard paradigm, a probability threshold, $P_{th} = 0.9$ and a sequence limit of 7. The predicted accuracy, A_{pr} , and expected stopping time, EST_{pr} , were obtained simulations assuming normally distributed classifier scores (normrnd() in MATLAB®), specified according to the detectability index. The observed performances, A_{obs} and EST_{obs} , are the reported results from Mainsah *et al.* (M2015) [7]. The left plot compares A_{pr} and A_{obs} . The right plot compares EST_{pr} and EST_{obs} .

Figure 10.

Variation in the observed accuracy, A_{obs} , estimated from N characters, given the true underlying system accuracy, A_{true} . Displayed is the probability distribution of a binomial random variable, $n \sim B(N, p)$, where n corresponds to the number of correctly selected characters, $p = A_{true}$, and scaled such that $A_{obs} = n/N$. The x -axis is truncated (i.e. A_{obs} 0.6), for visualization purposes.

Table 1

Summary of BCI studies

Study	Participant disability	Number of participants	Sessions per participant/ Characters spelled per session	Stimulus paradigm/Sequence limit
Throckmorton <i>et al.</i> [18]	Non-disabled	26	1/30	9 × 8 row-column/10
Mainsah <i>et al.</i> (2014) [19]	Non-disabled	17	1/36	9 × 8 row-column/10
Mainsah <i>et al.</i> (2015) [7]	ALS	10	3/12	6 × 6 checkerboard/7