End-to-end P300 BCI using Bayesian accumulation of Riemannian probabilities

Quentin Barthélemy, Sylvain Chevallier, Raphaëlle Bertrand-Lalo, Pierre Clisson

▶ **To cite this version:**

Quentin Barthélemy, Sylvain Chevallier, Raphaëlle Bertrand-Lalo, Pierre Clisson. End-to-end P300 BCI using Bayesian accumulation of Riemannian probabilities. Brain-Computer Interfaces, 2022, 10 (1), pp.50-61. 10.1080/2326263X.2022.2140467 . hal-03855169

HAL Id: hal-03855169

https://universite-paris-saclay.hal.science/hal-03855169

Submitted on 16 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

End-to-end P300 BCI using Bayesian accumulation of Riemannian probabilities

Quentin Barthélemy, Sylvain Chevallier, Raphaëlle Bertrand-Lalo, Pierre Clisson

arXiv:2203.07807v3 [cs.LG] 15 Nov 2022

***Abstract*—In brain-computer interfaces (BCI), most of the approaches based on event-related potential (ERP) focus on the detection of P300, aiming for single trial classification for a speller task. While this is an important objective, existing P300 BCI still require several repetitions to achieve a correct classification accuracy. Signal processing and machine learning advances in P300 BCI mostly revolve around the P300 detection part, leaving the character classification out of the scope. To reduce the number of repetitions while maintaining a good character classification, it is critical to embrace the full classification problem. We introduce an end-to-end pipeline, starting from feature extraction, and composed of an ERP-level classification using probabilistic Riemannian MDM which feeds a character-level classification using Bayesian accumulation of confidence across trials. Whereas existing approaches only increase the confidence of a character when it is flashed, our new pipeline, called Bayesian accumulation of Riemannian probabilities (ASAP), update the confidence of each character after each flash. We provide the proper derivation and theoretical reformulation of this Bayesian approach for a seamless processing of information from signal to BCI characters. We demonstrate that our approach performs significantly better than standard methods on public P300 datasets.**

***Index Terms*—Brain-computer interfaces, P300 classification, Riemannian geometry, character classification, Bayesian accumulation.**

I. Introduction

Brain-computer interfaces (BCI) have become a rising domain in neurotechnologies, allowing decoding brain activity, with multiple applications [1] using electroencephalography (EEG). Out-of-the-lab applications in EEG-based BCI face a difficult challenge: on top of the hard problem of decoding brain signals in real-time, most of the existing literature provides results that are difficult or even impossible to reproduce with online applications. A large portion of the BCI literature tackles processing or classification problem relying on offline analysis and dataset-specific preprocessing, or uses machine learning methods not suitable for real-world applications (long calibration process and model training phase). Indeed, the BCI community is continuously pushing toward increased reproducibility, robustness and acceptability of BCI, but the above-mentioned pitfalls are well documented [2]–[5].

Q. Barthélemy (ORCiD: 0000-0002-7059-6028) (corresponding author) is with Foxstream, 6 rue du Dauphiné, 69120 Vaulx-en-Velin, France (e-mail: q.barthelemy@gmail.com).

S. Chevallier (ORCiD: 0000-0003-3027-8241) is with Université Paris-Saclay, UVSQ, LISV, 78124, Vélizy-Villacoublay, France (e-mail: sylvain.chevallier@uvsq.fr).

R. Bertrand-Lalo is an independent scientist, Nantes, France (r.bertrand.lalo@gmail.com).

P. Clisson (ORCiD: 0000-0002-3123-9597) is an independent scientist, Paris, France (pierre@clisson.com).

In this study, we focus on a well-known class of BCI, that is based on the detection of event-related potentials (ERP) induced by an oddball effect. Common applications are P300-spellers [6]–[8] or P300-games, such as Brain Invaders [8] or Raccoons vs Demons [9]. P300 is an ERP elicited when the BCI flashes the target character (also called symbol or item). The BCI's task in these ERP-based paradigms is to detect the presence or absence of a P300 to retrieve the flashed character [6].

The low signal-to-noise ratio (SNR) of P300 waves in EEG signals makes it difficult to classify them on a single trial. Therefore, the first attempts to detect P300 waves consisted in averaging multiple trials (about 15 repetitions for each flashing character). While these attempts were able to successfully discriminate between target and non-target flashing [6], the resulting BCI was slow and the information transfer rate (ITR) [10] was low. To reduce the number of repetitions, further attempts applied advanced spatial filters to increase the amplitude related to P300 waves, like xDAWN which maximizes the signal to signal+noise ratio [7]. To further reduce the required number of repetitions, ERP detection was achieved with classifiers such as LDA or SVM trained on signals enhanced with spatial filters [11].

However, spatial filters are subject- and session-dependent [12]–[14], preventing cross-session and cross-subject transfer capabilities [8]. This implies that a subject should perform a careful calibration phase before being able to use the BCI; this calibration is required for each session even for someone using a BCI on a regular basis. Recently, Riemannian approaches applied to BCI have allowed the design of new classifiers, such as the minimum-distance-to-mean (MDM) [8], having high performances with shorter calibrations [4], [8]. Whereas optimized spatial filters estimate enhanced signals, Riemannian BCI work in the space of covariance matrices to process and classify the signal. While covariance captures the amplitude variations well, as induced by motor imagery tasks [15], it does not apply straightaway for ERP. It is possible to extend the covariance matrix with a P300 prototype [8] to obtain a discriminative representation that is appropriate for ERP classification. When applying those prototype-based covariance matrices, Riemannian BCI increased the robustness of P300-based BCI [4], [8], [16]. It should be noted that these works focus on a signal level, *i.e.* ERP detection, and not at a BCI system level, for character prediction.

In P300 BCI, the aim is to predict character from ERP classification. The most common approach is to increase the confidence of flashed characters if a P300 is detected. Since

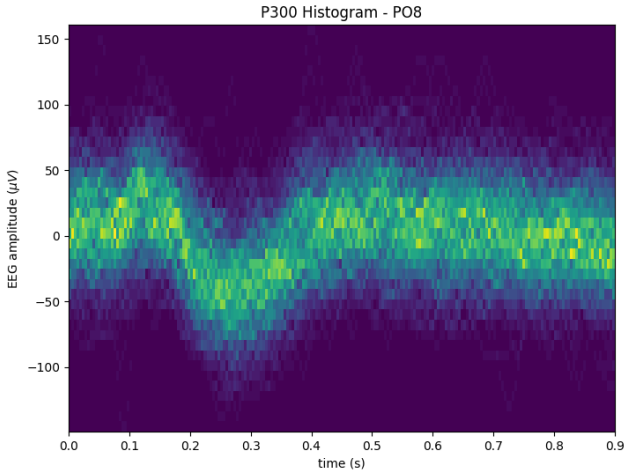

Fig. 1: Histogram of a P300 wave in channel PO8 across trials of a subject, estimated with [17].

Fig. 2: Virtual keyboard of a P300-speller using pseudo-random character flashes.

each character is seldom flashed, as expected in an oddball paradigm, and multiple repetitions are necessary to attain a reliable confidence in classification, the transformation from ERP detection to character prediction is a bottleneck that hinders the BCI speed.

To overcome this limitation, we introduce a novel Bayesian accumulation of Riemannian probabilities (ASAP). This is an end-to-end pipeline for P300 BCI classification, following these successive steps: (1) feature extraction, estimating prototype-based covariance matrices from EEG epochs, (2) ERP-level classification, performed with a probabilistic Riemannian MDM, (3) character-level classification, carried out with a Bayesian inference cumulating confidence after each trial. The ASAP pipeline provides an anytime computation framework, always yielding the best results in the sense of the maximum likelihood. One major contribution of our approach is to use both target and non-target flashes to softly update the character classification, while existing pipelines concentrate only on target flashes. By avoiding this loss of information, each flash contributes to updating the confidence estimation of the characters, leading to a faster P300 BCI.

In this article, Section III reviews the state-of-the-art P300 BCI [8] and its current limitations; Section IV shows the derivation of the new end-to-end pipeline. The new pipeline is applied on real P300-speller data, as described in Section V; and comparative results with the state-of-the-art are given in Section VI, followed by a Discussion and Conclusion.

II. Description of P300 BCI

BCI rely either on the decoding of different mental tasks or on the reaction to an external stimulation. The former are called independent BCI and the latter dependent BCI [18]. In dependent BCI, ERP are good candidates to design synchronous BCI systems. While error-related potentials [19], N170 (movement or face detection) or N400 (face recognition) have been investigated, the most common ERP-based BCI rely on the P300 wave [6]. The P300 wave is an automatic response to an oddball stimulus, *i.e.*, a stimulus which is both awaited and infrequent. The P300 wave is the result of a combination of components [20], the most prominent being a peak occurring 300 ms after the stimulus, as shown in Figure 1.

The most common applications of P300-based BCI are spellers [6]–[8] and games [8], [9]. Such systems display characters (also called symbols or items) that are flashed with a fixed interstimulus interval (ISI). To keep all characters within the visual field of the subject, they are displayed on a grid. Different flashing strategies have been proposed, using either row-column flashing or pseudo-random flashing. The flashing characters are usually presented briefly in inverted colors as shown on Figure 2, but it is possible to use different stimulus [21].

A trial is an EEG epoch of fixed length (often 1 s) starting at the time of the flash. When all the characters have been flashed once, the associated group of trials is called a repetition. P300-based BCI discriminate $K = 2$ ERP classes: target trial (denoted by $k = \text{T}$) where the chosen character has been flashed, evoking a P300 wave; and non-target trial (denoted $k = \text{NT}$) where the chosen character has not been flashed, resulting in an absence of a P300 wave. It is a difficult task to detect a P300 at a trial level (called single-trial detection), this is why the character classification is done after several repetitions to accumulate evidence and increase the accuracy. After several repetitions, the classifier makes a prediction to select a character and is ready to start over for another character prediction.

For a more complete description of P300-based BCI, the reader can refer to [7].

III. State of the art for P300 BCI classification

This section describes the state of the art in P300-based BCI, combining single-trial ERP classification achieved with Riemannian MDM [8] and counting occurrences of detected characters to make a prediction [6]. This approach yields good accuracy with short calibration time [11].

A. Riemannian MDM for ERP classification

Let denote by $X \in \mathbb{R}^{C \times N}$ a trial of EEG signals, recorded on C channels (or electrodes) and on N temporal samples.

This EEG signal has been previously band-pass filtered between 1 and 20 Hz. Riemannian approaches are not applied on the signal itself, but on covariance matrices estimated from this signal. Since the discriminant information is given by the temporal waveform of P300 rather than its spatial covariance, an extended trial $\dot{X} \in \mathbb{R}^{2C \times N}$ is built as [8]:

$$\dot{X} = \begin{bmatrix} P \\ X \end{bmatrix} , \tag{1}$$

where $P \in \mathbb{R}^{C \times N}$ denotes the prototype response, estimated as the grand average of target response. Then, the covariance matrix $\Sigma \in \mathbb{R}^{2C \times 2C}$ can be estimated as:

$$\Sigma = \frac{1}{N-1} \dot{X} \dot{X}^T . \tag{2}$$

Covariance matrices are symmetric positive definite (SPD), *i.e.*, they have strictly positive eigenvalues and are confined in a subspace of the Euclidean space [4]. Endowed with an appropriate distance, it is possible to define a dedicated geometry to consider those matrices, called a Riemannian manifold $\mathcal{M}$ [22], [23]. An appropriate distance could be the affine-invariant, defined between matrices Σ_1 and Σ_2 as [24]:

$$d(\Sigma_1, \Sigma_2) = \left(\sum_{c=1}^{2C} \log^2 e_c \right)^{\frac{1}{2}} ,$$

where e_c, $c = 1, \ldots, 2C$, are the eigenvalues of $\Sigma_1^{-1}\Sigma_2$.

The Riemannian minimum-distance-to-mean (MDM) is a deterministic classifier introduced in [15]. It is a generative classifier, requiring simply to estimate one mean $\bar{\Sigma}_k$ for each class $k = 1, \ldots, K$ during the training step:

$$\bar{\Sigma}_k = \arg \min_{\Sigma \in \mathcal{M}} \sum_{i \in \mathcal{I}(k)} d^2(\Sigma_i, \Sigma) ,$$

where $\mathcal{I}(k)$ is the set of indices of training matrices belonging to class k. This geometric mean has no closed-form solution and therefore has to be computed iteratively [25]. Thus, for a single-trial classification, the trial covariance Σ is assigned to the class with the closest mean:

$$\hat{k} = \arg \min_{k=1 \ldots K} d(\Sigma, \bar{\Sigma}_k) . \tag{3}$$

B. Maximization of occurrences for character classification

The historical character classification is a simple decision rule based on the maximization of occurrences of detected characters, *i.e.* characters flashed when a P300 has been detected in the trial. The character classification is performed after t trials [6]:

$$\hat{l}_t = \arg \max_{l=1 \ldots L} \sum_{\tau=1}^{t} \delta^{\mathrm{T}}_{\hat{k}_{l,\tau}} , \tag{4}$$

where δ^b_a is the Kronecker delta which equals 1 when $a = b$, and 0 otherwise; and where $\hat{k}_{l,\tau}$ denotes the output class/label of a single-trial ERP classification when character l has been flashed during trial τ. In the case of a Riemannian MDM, ERP classification is given by Eq. (3).

Note that the classification score in the right side of Eq. (4) can be easily transformed into a character probability in $[0, 1]$, dividing the sum by the number of trials.

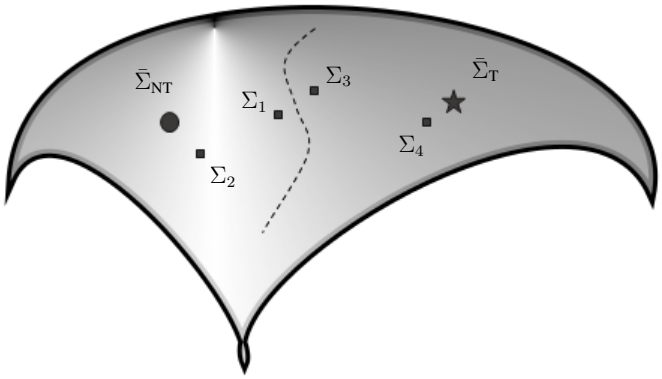

Fig. 3: Illustration of the classification of four trials in the space of SPD matrices, for two classes NT and T with their centers $\bar{\Sigma}_{\mathrm{NT}}$ and $\bar{\Sigma}_{\mathrm{T}}$, their equidistance represented by the dotted line. In this example, the target character has been flashed at trial 4, and not flashed before. At ERP classification, deterministic MDM sees no difference between trials Σ_3 and Σ_4 (same class), whereas probabilistic MDM does (same class but different probabilities); deterministic MDM sees a huge difference between trials Σ_3 and Σ_1 (different classes), whereas probabilistic MDM understands their similarity (different classes but close probabilities). At character classification, the argmax-argmin classification updates its sum only during flashed trial 4, whereas Bayesian accumulation updates its confidence after each trial.

C. Limitations

On the one hand, combining Eq. (3) and (4) gives an argmax-argmin character classification. This approach is not optimal, as it loses information: Riemannian distance to ERP-class centers yields a crucial information to quantify the confidence into the ERP classification. This information is lost in the deterministic classification shown in Eq. (3). The confidence in ERP class could be transformed in a continuous probability, allowing to softly consolidate character classification confidence along trials. This is not possible in the argmax-argmin approach, where single-trial ERP classification error strongly impacts the character classification. The need for a probabilistic MDM rather than a deterministic one is illustrated on Figure 3.

On the other hand, for usual character classifier, the confidence in a character is increased only when the character is flashed and a P300 is detected in the trial. In the case of a Riemannian MDM used for ERP classification [8], this occurs when the trial covariance matrix is close to the mean of the target class. Indeed, it seems possible to increase the confidence for the target character when the character is not flashed and the trial covariance matrix is close to the mean of non-target class. This information is currently not captured by Eq. (4), resulting in a loss of discriminative power since non-target flashes are more frequent than target ones.

IV. End-to-end P300 BCI classification

This section expounds the derivation of an end-to-end P300 BCI pipeline, building on Bayesian accumulation to perform character classification. The details of this derivation are

indicated below, providing all underlying assumptions required to obtain the final expression.

A. Bayesian accumulation of character probabilities

In an online setup, after accumulating $\tau = 1, \dots, t$ trials of the same target character, the cumulative posterior probability of character l can be written as:

$$p(l|X_1, \dots, X_t) = \frac{p(X_t|l, X_1, \dots, X_{t-1})\ p(l|X_1, \dots, X_{t-1})}{p(X_t|X_1, \dots, X_{t-1})} , \quad (5)$$

combining Bayes' rule with the chain rule of probability. Since this expression is intractable, we make an approximation assuming that X_t is independent from $X_1, \dots, X_{t-1}$. While this is true for X_1, it is only partially true for X_{t-1}, as it depends on the overlap between epochs X_{t-1} and X_t). Eq. (5) is simplified into:

$$p(l|X_1, \dots, X_t) = \frac{p(X_t|l)\ p(l|X_1, \dots, X_{t-1})}{p(X_t)} , \quad (6)$$

and, if $p(l|X_1, \dots, X_t)$ is renamed $p_t(l)$, it becomes:

$$p_t(l) = \frac{p(X_t|l)\ p_{t-1}(l)}{p(X_t)} , \quad (7)$$

where $p_{t-1}(l)$ can be seen as the prior from previous trials. It is dynamically updated, as the previous posterior becomes the following prior. Using the law of total probability, the denominator is expanded as:

$$p_t(l) = \frac{p(X_t|l)\ p_{t-1}(l)}{\sum_{\lambda=1}^{L} p(X_t|\lambda)\ p_{t-1}(\lambda)} . \quad (8)$$

Since $p_{t-1}(l)$ is a fraction depending of $p_{t-2}(l)$, and $p_{t-2}(l)$ a fraction depending of $p_{t-3}(l)$, etc., this formula can be expanded in a finite continued fraction until $p_0(l)$, which can be finally simplified into:

$$p_t(l) = \frac{\prod_{\tau=1}^{t} p(X_\tau|l)\ p_0(l)}{\sum_{\lambda=1}^{L} \prod_{\tau=1}^{t} p(X_\tau|\lambda)\ p_0(\lambda)} . \quad (9)$$

where $p_0(l)$ is the initial prior on character l.

B. Bayesian accumulation of ERP probabilities

Reverting the marginalization of the ERP class k out of the likelihood, we have:

$$\begin{aligned} p(X_\tau|l) &= \sum_{k=1}^{K} p(X_\tau, k|l) \\ &= \sum_{k=1}^{K} p(X_\tau|k, l)\ p(k|l) \\ &= \sum_{k=1}^{K} p(X_\tau|k)\ p(k|l) , \end{aligned} \quad (10)$$

because epoch X_τ depends only on ERP class k, not on character l. Considering a P300 BCI with $K = 2$ ERP classes, $k = \mathrm{T}$ for target and $k = \mathrm{NT}$ for non-target, Eq. (10) becomes:

$$p(X_\tau|l) = p(X_\tau|\mathrm{T})\ p(\mathrm{T}|l) + p(X_\tau|\mathrm{NT})\ p(\mathrm{NT}|l) , \quad (11)$$

where $p(\mathrm{T}|l) = 1$ (*resp.* 0) and $p(\mathrm{NT}|l) = 0$ (*resp.* 1) when the character l has been flashed (*resp.* not flashed) at trial τ. This deterministic binarization justifies the equivalence between "target character flashed" and "presence of ERP in trial".

Finally, Eq. (9) becomes:

$$p_t(l) = \frac{\prod_{\tau=1}^{t} \left(p(X_\tau|\mathrm{T})\ p(\mathrm{T}|l) + p(X_\tau|\mathrm{NT})\ p(\mathrm{NT}|l)\right)\ p_0(l)}{\sum_{\lambda=1}^{L} \prod_{\tau=1}^{t} \left(p(X_\tau|\mathrm{T})\ p(\mathrm{T}|\lambda) + p(X_\tau|\mathrm{NT})\ p(\mathrm{NT}|\lambda)\right)\ p_0(\lambda)} . \quad (12)$$

This formula allows a seamless classification for P300 BCI from EEG epochs to characters, without requiring an explicit ERP-level classification. It is valid for any type of probabilistic and generative ERP classifier returning $p(X_\tau|k)$, and thus could be used tomorrow with deep neural networks [26].

C. Bayesian accumulation of Riemannian probabilities

Since Riemannian classifiers obtain best results for ERP classification [8], [11], we derive a probabilistic version of Riemannian MDM, called pMDM. Assuming that epoch X follows a multivariate normal model $\mathcal{N}(0, \Sigma)$, with zero mean (EEG signals are centered after a band-pass filtering) and with a covariance matrix $\Sigma \in \mathcal{M}$, we define $p(X|k) = p(\Sigma|k)$.

To represent covariance matrices belonging to a Riemannian space, the Riemannian Gaussian distribution can be used to model each class. Its probability density function is defined as [29]:

$$p(\Sigma|k) = p(\Sigma|\bar{\Sigma}_k, \sigma_k) = \frac{1}{\zeta(\sigma_k)} \exp\left(-\frac{d^2(\Sigma, \bar{\Sigma}_k)}{2\sigma_k^2}\right) , \quad (13)$$

where $\bar{\Sigma}_k \in \mathcal{M}$ is the center of the Riemannian Gaussian distribution of class k, $\sigma_k > 0$ encodes the dispersion of the distribution, and $\zeta(\sigma_k)$ is a normalization factor. Under the assumption that all classes have identical dispersion σ_k, the Riemannian instantiation of Eq. (12) is:

$$p_t(l) = \frac{\prod_{\tau=1}^{t} e^{-d^2(\Sigma_\tau, \tilde{\Sigma}_{l,\tau})}\ p_0(l)}{\sum_{\lambda=1}^{L} \prod_{\tau=1}^{t} e^{-d^2(\Sigma_\tau, \tilde{\Sigma}_{\lambda,\tau})}\ p_0(\lambda)} , \quad (14)$$

where $\tilde{\Sigma}_{l,\tau} = \bar{\Sigma}_{\mathrm{T}}$ if character l has been flashed during trial τ, and $\tilde{\Sigma}_{l,\tau} = \bar{\Sigma}_{\mathrm{NT}}$ otherwise.

After trial t, the character classification is obtained thanks to the maximum *a posteriori* (MAP), *i.e.* maximizing the posterior probability $p(l|X_1, \dots, X_t)$. The MAP classification rule is:

$$\begin{aligned} \hat{l}_t &= \arg\max_{l=1\dots L}\ p_t(l) \\ &= \arg\max_{l=1\dots L} \prod_{\tau=1}^{t} e^{-d^2(\Sigma_\tau, \tilde{\Sigma}_{l,\tau})}\ p_0(l) , \end{aligned} \quad (15)$$

because the denominator is independent from l. Priors $p_0(l)$ can easily be used to embed letter and word prediction in BCI [30]. Under the assumption of equally likely classes, *i.e.* equal priors $p_0(l)$, it becomes a maximum likelihood (ML) classification rule:

$$\hat{l}_t = \arg\min_{l=1\dots L} \sum_{\tau=1}^{t} d^2(\Sigma_\tau, \tilde{\Sigma}_{l,\tau}) . \quad (16)$$

Name	# Channels	# Subjects	# Characters per subject	# Trials per character (T / NT)	Reference
BNCI 2014-008	8	8	35	20 / 100	[27]
BNCI 2014-009	16	10	18	16 / 80	[28]

TABLE I: Details of datasets used for evaluation of P300-speller.

Contrary to the argmax-argmin approach, this end-to-end derivation leads to a single argmin approach for the ML rule, which is a seamless processing of information from EEG epochs to BCI characters. It prevents loss of information because all characters softly update their confidence on both target and non-target flashes, contributing to increasing the confidence of the target character.

D. Contributions summary

This Bayesian accumulation of Riemannian probabilities is called ASAP. This online algorithm must be re-initialized at the beginning of each new character classification, and it can be stopped with any dynamic stopping strategy [31], [32].

ASAP is an end-to-end P300 BCI, with a classification scheme occurring at character-level, which is the actual end of ERP-based BCI. Most of the existing approaches are assessed at ERP-level, *e.g.* as detailed in [4], [8], [16]. While these are useful for signal processing and classification, they do not encompass the character classification which was regarded as an engineering issue. We demonstrate in this article that a principled and theoretical approach is possible, and that combining Riemannian geometry and Bayesian inference allow us to devise a character classification pipeline, integrating ERP probabilization (instead of binarization) in the process.

Note that taking into account non-target flashes has already been considered for P300 classification with a Bayesian framework [33], but for a discriminative classifier. This kind of classifier provides $p(k|X_t)$ instead of $p(X_t|k)$ for generative ones [34], and requires a large training set to avoid overfitting, *e.g.* 264,000 trials are used in [33]), which prevents reproducibility, scalability and out-of-the-lab deployment.

V. Application to P300-speller

The introduced pipeline can be applied to all ERP-based BCI. As validation, it is applied here on the P300-speller task described in Section III-B. While most works evaluate their performances at ERP-level [4], [8], [16], we evaluate them at character-level, which is the actual end of the BCI pipeline.

A. Data

EEG data come from 2 datasets containing 18 subjects and gathering 50,880 trials. The important information of the datasets is provided in Table I.

1) BNCI 2014-008 dataset: This EEG dataset is publicly available [27] and gathers the recordings of 8 patients with amyotrophic lateral sclerosis visually focusing on a 6-letter matrix. The data are recorded with a $C = 8$ channels amplifier (g.MOBILAB, g.tec, Austria) using active electrodes (g.Ladybird, g.tec, Austria) that are located at Fz, Cz, Pz, Oz, P3, P4, PO7 and PO8. Reference is the right ear lobe and the ground is taken from the left mastoid. The signal is acquired at 256 Hz. The speller randomly highlights the lines and columns from the letter matrix for 125 ms followed by 125 ms of ISI, resulting in a stimulus onset asynchrony of 250 ms.

2) BNCI 2014-009 dataset: This EEG dataset is publicly available [28] and gathers the recordings of 10 healthy subjects visually focusing on a 6-letter matrix. The data are recorded with a $C = 16$ electrodes that are located at Fz, FCz, Cz, CPz, Pz, Oz, F3, F4, C3, C4, CP3, CP4, P3, P4, PO7, and PO8. Each electrode is referenced to the linked earlobes and the ground is taken from the right mastoid. The EEG was acquired at 256 Hz.

B. Methods

Raw EEG signals are band-pass filtered between 1 to 20 Hz. Trials are extracted as EEG epochs of fixed length (1 s) starting at the time of flashes.

Four single-trial character-level classifiers for P300 BCI are compared in this article:

- ASAP with equal priors, *i.e.* the ML version in Eq. (16), which is the novel Bayesian accumulation (BA) of probabilistic Riemannian MDM, a kind of pMDM+BA pipeline;
- MDM+OM: Riemannian MDM [8], see Eq. (3), followed by occurrences maximizing (OM), see Eq. (4), which is the state of the art described in Section III;
- xDAWN+OM: downsampling at 128 Hz, followed by xDAWN [7], regularized LDA on vectorized data, and OM;
- RegLDA+OM: downsampling at 128 Hz, followed by regularized LDA on vectorized data [35], [36], and OM.

C. Evaluation

The current experiment is a within-session classification of P300-speller characters, comparing evolution of performances (probabilities, accuracy and BCI ITR [10]) along time (flash and repetition). For each session of each subject, the training set contains the trials associated to the first 6 characters (to calibrate the $K = 2$ class centers $\bar{\Sigma}_{\mathrm{T}}$ and $\bar{\Sigma}_{\mathrm{NT}}$ for MDM+OM and ASAP), and the test set contains remaining trials.

The code of the experiment relies on the rich ecosystem of open source libraries available for BCI datasets, EEG signal processing and machine learning, that is Timeflux [37], MOABB [38], MNE [39], scikit-learn [40] and pyRiemann [17].

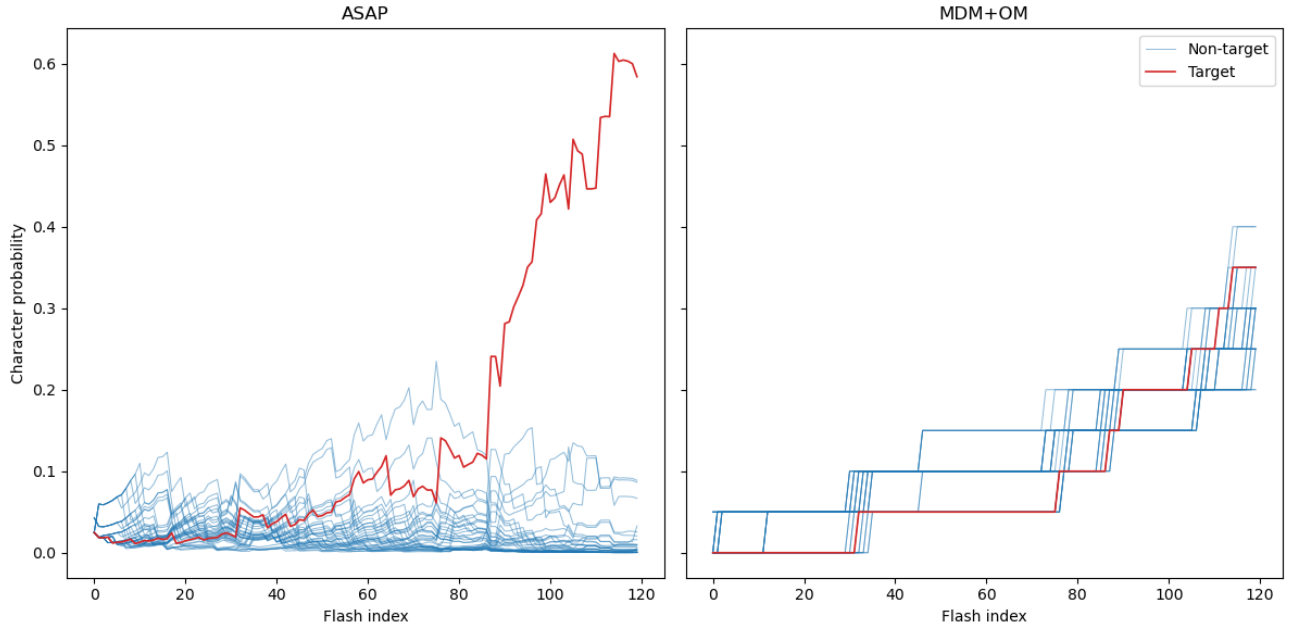

Fig. 4: Character probabilities $p_t(l)$ as a function of flash t, for only one character classification, for the target character in red and the non-target characters in blue. Left: ASAP, softly updated after each flash. Right: MDM+OM, hardly updated only on target flashes.

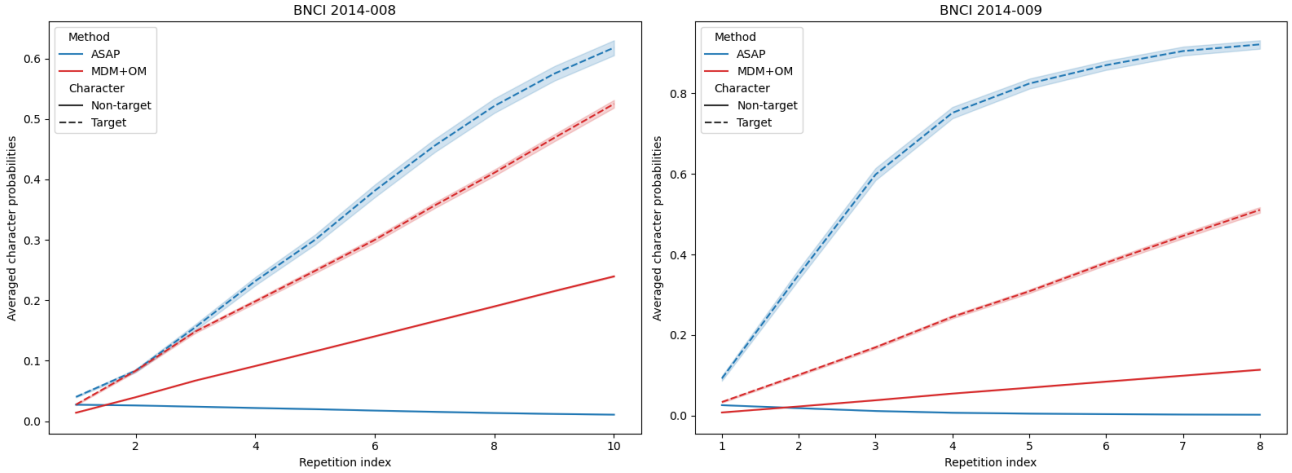

Fig. 5: Character probabilities as a function of repetition, for MDM+OM and ASAP, averaged across all character classifications of each dataset.

For research reproducibility, code of offline experiments is available at https://github.com/sylvchev/asap-p300-bci. Open-source implementation of ASAP is available as an online Timeflux [37] application: https://github.com/timeflux/demos/tree/main/speller/P300. This free and open framework allows to evaluate the proposed implementation in a real online setup.

VI. Results

Figure 4 displays the character probabilities $p_t(l)$ as a function of flash t, for only one character classification. The target character is in red and non-target characters are in blue. On the right side, we observe that, for MDM+OM, only probabilities of flashed characters are updated, and that probabilities of non-target characters increase with time. In this example, the classification will be incorrect because target character does not have the highest probability. On the left part, we can see that all probabilities of ASAP are updated after each flash: it softly consolidates confidence across target flashes, but also non-target ones, which are much more frequent. We also see that probabilities of non-target characters decrease over time and actually contribute to increasing confidence in the target character. Dependence between character probabilities comes the denominator of Eq. (14).

Character probabilities are averaged across all character classifications of each dataset. Figure 5 shows the character probabilities as a function of repetition, with the average and its confidence interval at 95 %. For MDM+OM, the averaging has smoothed the steps visible on the right part of Figure 4. This figure confirms that ASAP increases probabilities of

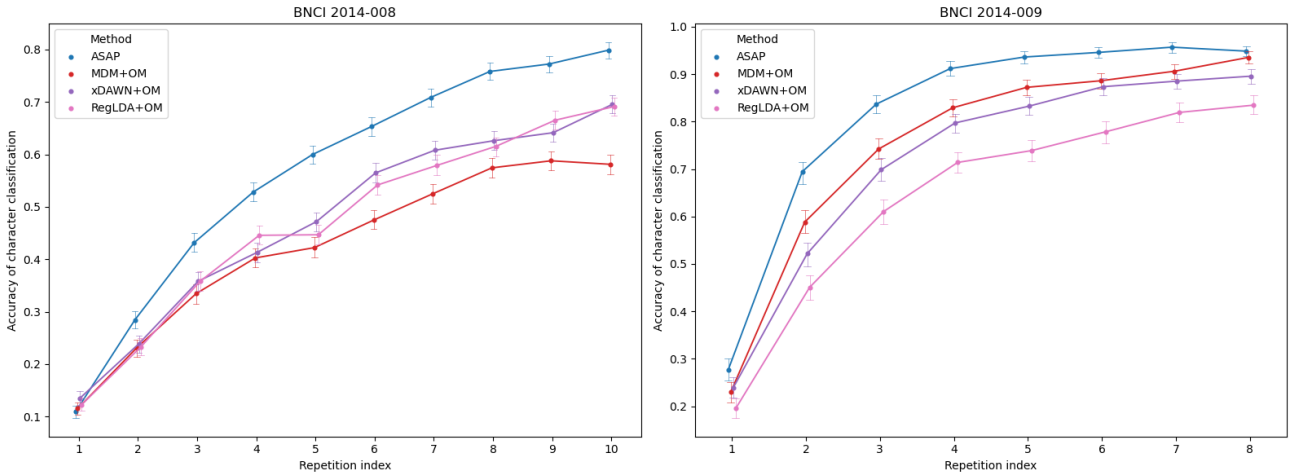

Fig. 6: Character classification accuracy as a function of repetition, for ASAP, MDM+OM, xDAWN+OM and RegLDA+OM, averaged across all character classifications of both datasets.

target characters faster than MDM+OM, and that ASAP decreases probabilities of non-target characters contrary to MDM+OM. Combining these two properties provides a faster and larger divergence between probabilities of target and non-target characters.

Figure 6 displays the accuracy of character classification performed after each repetition, averaged across all character classifications of each dataset. We can see that trends observed in probabilities have a strong impact on classification accuracy. Divergences between probabilities allow for a better discrimination between target and non-target characters. The first dataset (left side) is more challenging than the second one (right side) where subjects had previous experience with P300-based BCI, but ASAP is always better and faster than MDM+OM, xDAWN+OM and RegLDA+OM. For a fixed number of repetitions, ASAP gives a better classification than all other methods; and for a fixed level of confidence ASAP allows for a faster classification than all other methods.

Figure 7 displays the BCI ITR in bits/min computed after each repetition, averaged across all character classifications of each dataset. For each dataset, methods reach their optimal peak of BCI ITR for the same number of repetitions. We can see that differences previously observed in accuracy have an obvious impact on BCI ITR: ASAP is always higher than MDM+OM, xDAWN+OM and RegLDA+OM, giving better performance of communication.

One the one hand, the RegLDA+OM and xDAWN+OM methods use the same classifier, a regularized LDA. It is applied on downsampled trials for RegLDA+OM, and on downsampled and spatially filtered trials for xDAWN+OM. Results show that xDAWN spatial filter, applying a supervised dimension reduction, enhances most of the time classification performances. On the other hand, MDM+OM and ASAP methods use the same features (covariance matrices Σ), the same training models (covariance centers $\bar{\Sigma}_{\mathrm{T}}$ and $\bar{\Sigma}_{\mathrm{NT}}$), and the same dissimilarity measure between features (Riemannian distance). Thus, all observed differences come only from the way to exploit information, from the distance between covariance matrices to character classes. While MDM+OM suffers of all limitations described in Section III-C, like the argmax-argmin character classification, ASAP consolidates character confidences using probabilities which are softly accumulated after each trial, that is illustrated on the left side of Figure 4.

To conclude these experiments performed on real data, we have shown that ASAP outperforms MDM+OM, xDAWN+OM and RegLDA+OM, providing a better and faster P300 BCI. It reduces the number of repetitions necessary for the correct classification of characters. Numerical values of figures can be found in Appendix.

VII. Discussion

By updating confidence on non-target flashes too, this approach aims to exploit all information available in EEG trials. There are several advantages, as it reduces the time required to spell a letter, it preserves the concentration of the subject by avoiding the tiredness of seeing multiple flashes, and it avoids wasting data, as the storage of EEG can be optimized. In experiments, we have compared evolution of performances along time. We have not used dynamic stopping [31], [32], because compared methods are independent of it. Adding this step will be done in a future work.

Note that Eq. (9) has been derived for any type of online classification. So it could be applied to other BCI paradigms, especially asynchronous BCI making the assumption of temporal persistence in the class across several epochs, like asynchronous SSVEP [41], motor imagery [42] or neurofeedback [43]. More generally, the Bayesian scheme derived in Section IV-A makes no assumption on the type of input data X and is independent from the two-level classification specific to P300 BCI. Consequently, it remains valid for any type of sequential data. Derived in this article for multivariate biological time series, it could be applied to other data such as audio stream [44], video stream [45], *etc*.

Underlying assumptions of the derivation of Section IV have been explicitly listed, in order to be reconsidered in future works. Notably, the deterministic binarization performed after

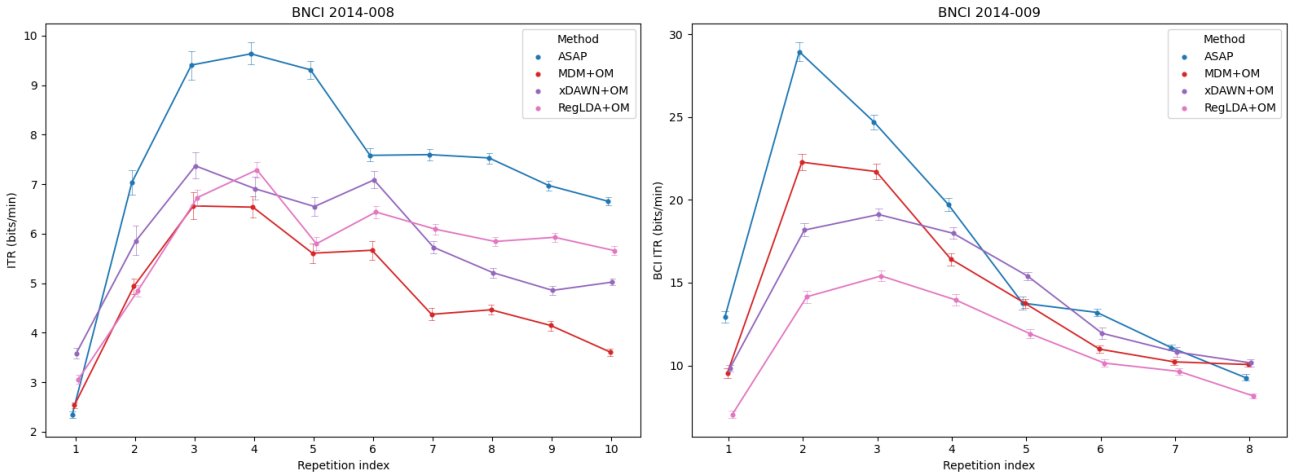

Fig. 7: BCI ITR (in bits/min) as a function of repetition, for ASAP, MDM+OM, xDAWN+OM and RegLDA+OM, averaged across all character classifications of each dataset.

Eq. (11) does not take into account that a lack of concentration of the subject can prevent ERP to be elicited, damaging BCI performances. A new generation of algorithms should be able to break this simple deterministic link between flash and ERP, and to model uncertainty thanks to continuous probabilities for $p(\mathrm{T}|l)$ and $p(\mathrm{NT}|l)$, which could be defined monitoring concentration level in real-time.

VIII. Conclusion

Using a Bayesian accumulation of Riemannian probabilities, this article introduces an end-to-end P300 BCI, providing a seamless processing of information from EEG epochs to BCI characters. The derivation of this end-to-end pipeline details all underlying assumptions required to obtain the final formula, for further potential generalizations. Validated on real EEG data, this new P300 BCI is better and faster than the state-of-the-art, reducing the number of repetitions necessary for the correct classification of characters.

Future works are to combine this new approach with the following features: online adaptation of centers of class [8]; cross-subject transfer learning of centers of class [46]–[48]; dynamic stopping for early classification [31], [32]; and letter or word prediction, embedded in BCI thanks to character-dependent priors $p_0(l)$, to boost BCI performances [30].

IX. Disclosure of Interest

The authors report there are no competing interests to declare. This work was not supported by any grant nor funding and QB/RBL/PC have conducted this research during their spare time.

Appendix
Numerical Values

Numerical values of Figure 6 are given in Table II, and Figure 7 in Table III.

References

[1] F. L. C. S. Nam, A. Nijholt, Ed., *Brain-Computer Interfaces Handbook*. CRC Press, 2018.

[2] J. E. Huggins, C. Guger, B. Allison, C. W. Anderson, A. Batista, A.-M. Brouwer, C. Brunner, R. Chavarriaga, M. Fried-Oken, A. Gunduz *et al.*, "Workshops of the fifth international brain-computer interface meeting: defining the future," *Brain-Computer Interfaces*, vol. 1, pp. 27–49, 2014.

[3] C. Brunner, N. Birbaumer, B. Blankertz, C. Guger, A. Kübler, D. Mattia, J. R. Millán, F. Miralles, A. Nijholt, E. Opisso, N. Ramsey, P. Salomon, and G. R. Müller-Putz, "BNCI Horizon 2020: towards a roadmap for the BCI community," *Brain-Computer Interfaces*, vol. 2, pp. 1–10, 2015.

[4] M. Congedo, A. Barachant, and R. Bhatia, "Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review," *Brain-Computer Interfaces*, vol. 4, pp. 1–20, 2017.

[5] M. Rashid, N. Sulaiman, A. P. Majeed, R. M. Musa, A. F. A. Nasir, B. S. Bari, and S. Khatun, "Current status, challenges and possible solutions of EEG based brain-computer interface: A comprehensive review," *Front Neurorobot*, vol. 14, p. 25, 2020.

[6] L. A. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr Clin Neurophysiol*, vol. 70, pp. 510–523, 1988.

[7] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert, "xDAWN algorithm to enhance evoked potentials: Application to brain-computer interface," *IEEE Trans Biomed Eng*, vol. 56, pp. 2035–2043, 2009.

[8] A. Barachant and M. Congedo, "A Plug&Play P300 BCI using information geometry," *arXiv preprint arXiv:1409.0107*, 2014.

[9] V. Goncharenko, R. Grigoryan, and A. Samokhina, "Raccoons vs Demons: Multiclass labeled P300 dataset," *arXiv preprint arXiv:2005.02251*, 2020.

[10] P. Yuan, X. Gao, B. Allison, Y. Wang, G. Bin, and S. Gao, "A study of the existing problems of estimating the information transfer rate in online brain–computer interfaces," *J Neural Eng*, vol. 10, p. 026014, 2013.

[11] L. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update," *J Neural Eng*, vol. 15, p. 031005, 2018.

[12] R. Tomioka, K. Aihara, and K.-R. Müller, "Logistic regression for single trial EEG classification," in *NIPS*, vol. 19, 2007, pp. 1377–1384.

[13] B. Reuderink and M. Poel, "Robustness of the common spatial patterns algorithm in the BCI-pipeline," Univ. Twente, Tech. Rep., 2008.

[14] M. Grosse-Wentrup, C. Liefhold, K. Grasmann, and M. Buss, "Beamforming in noninvasive brain-computer interfaces," *IEEE Trans Biomed Eng*, vol. 56, pp. 1209–1219, 2009.

[15] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass brain–computer interface classification by Riemannian geometry," *IEEE Trans Biomed Eng*, vol. 59, no. 4, pp. 920–928, 2012.

[16] L. Korczowski, M. Congedo, and C. Jutten, "Single-trial classification of multi-user P300-based brain-computer interface using riemannian geometry," in *EMBC*, 2015, pp. 1769–1772.

Repetition index	1	2	3	4	5	6	7	8	9	10
BNCI 2014-008, ASAP	0.11	0.28	0.43	0.53	0.6	0.65	0.71	0.76	0.77	0.8
BNCI 2014-008, MDM+OM	0.12	0.23	0.33	0.4	0.42	0.47	0.52	0.57	0.59	0.58
BNCI 2014-008, xDAWN+OM	0.13	0.24	0.36	0.41	0.47	0.56	0.61	0.63	0.64	0.7
BNCI 2014-008, RegLDA+OM	0.12	0.23	0.36	0.45	0.45	0.54	0.58	0.62	0.67	0.69
BNCI 2014-009, ASAP	0.28	0.69	0.84	0.91	0.94	0.95	0.96	0.95		
BNCI 2014-009, MDM+OM	0.23	0.59	0.74	0.83	0.87	0.89	0.91	0.94		
BNCI 2014-009, xDAWN+OM	0.24	0.52	0.7	0.8	0.83	0.87	0.89	0.9		
BNCI 2014-009, RegLDA+OM	0.2	0.45	0.61	0.71	0.74	0.78	0.82	0.83		

TABLE II: Numerical values of Figure 6: averaged character classification accuracy as a function of repetition.

Repetition index	1	2	3	4	5	6	7	8	9	10
BNCI 2014-008, ASAP	2.4	7.0	9.4	9.6	9.3	7.6	7.6	7.5	7.0	6.7
BNCI 2014-008, MDM+OM	2.5	4.9	6.6	6.5	5.6	5.7	4.4	4.5	4.1	3.6
BNCI 2014-008, xDAWN+OM	3.6	5.9	7.4	6.9	6.5	7.1	5.7	5.2	4.9	5.0
BNCI 2014-008, RegLDA+OM	3.1	4.8	6.7	7.3	5.8	6.4	6.1	5.8	5.9	5.7
BNCI 2014-009, ASAP	12.9	28.9	24.7	19.7	13.8	13.2	11.1	9.3		
BNCI 2014-009, MDM+OM	9.6	22.3	21.7	16.4	13.8	11.0	10.2	10.1		
BNCI 2014-009, xDAWN+OM	9.8	18.2	19.1	18.0	15.4	11.9	10.8	10.2		
BNCI 2014-009, RegLDA+OM	7.0	14.2	15.4	14.0	11.9	10.1	9.6	8.2		

TABLE III: Numerical values of Figure 7: averaged BCI ITR as a function of repetition.

[17] pyRiemann, "https://pyriemann.readthedocs.io/."

[18] J. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, pp. 767–791, 2002.

[19] R. Chavarriaga and J. d. R. Millan, "Learning from EEG error-related potentials in noninvasive brain-computer interfaces," *IEEE Trans Neural Syst Rehabil Eng*, vol. 18, pp. 381–388, 2010.

[20] S. J. Luck, *An introduction to the event-related potential technique*. MIT Press, 2014.

[21] J. Jin, B. Z. Allison, T. Kaufmann, A. Kübler, Y. Zhang, X. Wang, and A. Cichocki, "The changing face of P300 BCIs: a comparison of stimulus changes in a P300 BCI involving faces, emotion, and movement," *PloS One*, vol. 7, p. e49688, 2012.

[22] M. Congedo, "EEG source analysis," Grenoble University, Tech. Rep. HDR, 2013.

[23] F. Yger, M. Berar, and F. Lotte, "Riemannian approaches in Brain-Computer Interfaces: a review," *IEEE Trans Neural Syst Rehabil Eng*, vol. 25, pp. 1753–1762, 2017.

[24] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM J Matrix Anal Appl*, vol. 26, no. 3, pp. 735–747, 2005.

[25] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Trans Med Imaging*, vol. 23, no. 8, pp. 995–1005, 2004.

[26] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalographyanalysis: a systematic review," *J Neural Eng*, vol. 16, p. 051001, 2019.

[27] A. Riccio, L. Simione, F. Schettini, A. Pizzimenti, M. Inghilleri, M. Olivetti Belardinelli, D. Mattia, and F. Cincotti, "Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis," *Front Hum Neurosci*, vol. 7, p. 732, 2013.

[28] F. Aloise, P. Aricò, F. Schettini, A. Riccio, S. Salinari, D. Mattia, F. Babiloni, and F. Cincotti, "A covert attention P300-based brain-computer interface: Geospell," *Ergonomics*, vol. 55, pp. 538–551, 2012.

[29] S. Said, L. Bombrun, Y. Berthoumieu, and J. Manton, "Riemannian Gaussian distributions on the space of symmetric positive definite matrices," *IEEE Trans Inf Theory*, vol. 63, pp. 2153–2170, 2017.

[30] T. Kaufmann, S. Völker, L. Gunesch, and A. Kübler, "Spelling is just a click away – a user-centered brain-computer interface including auto-calibration and predictive text entry," *Front Neurosci*, vol. 6, 2012.

[31] M. Schreuder, J. Höhne, B. Blankertz, S. Haufe, T. Dickhaus, and M. Tangermann, "Optimizing event-related potential based brain-computer interfaces: a systematic evaluation of dynamic stopping methods," *J Neural Eng*, vol. 10, p. 036025, 2013.

[32] P.-J. Kindermans, M. Tangermann, K.-R. Müller, and B. Schrauwen, "Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller," *J Neural Eng*, vol. 11, p. 035005, 2014.

[33] E. Daucé and E. Thomas, "Evidence build-up facilitates on-line adaptivity indynamic environments: example of the BCI P300-speller," in *ESANN*, 2014, pp. 377–382.

[34] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes," in *NIPS*, 2002, pp. 841–848.

[35] D. J. Krusienski, E. W. Sellers, F. Cabestaing, S. Bayoudh, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, "A comparison of classification techniques for the P300 speller," *J Neural Eng*, vol. 3, p. 299, 2006.

[36] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K. R. Müller, "Single-trial analysis and classification of ERP components - A tutorial," *NeuroImage*, vol. 56, pp. 814–825, 2011.

[37] P. Clisson, R. Bertrand-Lalo, M. Congedo, G. Victor-Thomas, and J. Chatel-Goldman, "Timeflux: an open-source framework for the acquisition and near real-time processing of signal streams," in *GBCIC*, 2019.

[38] V. Jayaram and A. Barachant, "MOABB: trustworthy algorithm benchmarking for BCIs," *J Neural Eng*, vol. 15, p. 066011, 2018.

[39] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen, "MNE software for processing MEG and EEG data," *NeuroImage*, vol. 86, pp. 446–460, 2014.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J Mach Learn Res*, vol. 12, pp. 2825–2830, 2011.

[41] E. K. Kalunga, S. Chevallier, Q. Barthélemy, K. Djouani, E. Monacelli, and Y. Hamam, "Online SSVEP-based BCI using Riemannian geometry," *Neurocomputing*, vol. 191, pp. 55–68, 2016.

[42] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172–178, 2013.

[43] M. Arns, J. M. Batail, S. Bioulac, M. Congedo, C. Daudet, D. Drapier, T. Fovet, R. Jardri, M. Le Van Quyen, F. Lotte, D. Mehler, M.-F. J. A., D. Purper-Ouakil, and F. Vialatte, "Neurofeedback: One of today's techniques in psychiatry?" *Encephale*, vol. 43, pp. 135–145, 2017.

[44] S. Sen, A. Dutta, and N. Dey, *Audio processing and speech recognition: concepts, techniques and research overviews*. Springer, 2019.

[45] Y. Wang, J. Ostermann, and Y. Q. Zhang, *Video Processing and Communications*. Prentice hall Upper Saddle River, NJ, 2002.

[46] E. K. Kalunga, S. Chevallier, and Q. Barthélemy, "Transfer learning for SSVEP-based BCI using Riemannian similarities between users," in *EUSIPCO*, 2018, pp. 1685–1689.

[47] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, "Transfer learning: a Riemannian geometry framework with applications to brain-computer interfaces," *IEEE Trans Biomed Eng*, vol. 65, pp. 1107–1116, 2018.

[48] P. L. C. Rodrigues, C. Jutten, and M. Congedo, "Riemannian Procrustes analysis: Transfer learning for brain-computer," *IEEE Trans Biomed Eng*, vol. 66, pp. 2390–2401, 2019.