Computer Methods and Programs in Biomedicine 215 (2022) 106623

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Robust asynchronous control of ERP-Based brain-Computer interfaces using deep learning

Eduardo Santamaría-Vázquez[a,b,*], Víctor Martínez-Cagigal[a,b], Sergio Pérez-Velasco[a], Diego Marcos-Martínez[a], Roberto Hornero[a,b]

[a] Biomedical Engineering Group, E.T.S Ingenieros de Telecomunicación, University of Valladolid, Paseo de Belén 15, 47011, Valladolid, Spain
[b] Centro de Investigación Biomédica en Red en Bioingeniería, Biomateriales y Nanomedicina, (CIBER-BBN), Spain

ARTICLE INFO

Article history:
Received 23 August 2021
Revised 11 December 2021
Accepted 4 January 2022

Keywords:
Brain–computer interfaces
Event-related potentials
P300
Asynchrony
Control state detection
Deep learning
Convolutional neural networks

ABSTRACT

Background and Objective. Brain-computer interfaces (BCI) based on event-related potentials (ERP) are a promising technology for alternative and augmented communication in an assistive context. However, most approaches to date are synchronous, requiring the intervention of a supervisor when the user wishes to turn his attention away from the BCI system. In order to bring these BCIs into real-life applications, a robust asynchronous control of the system is required through monitoring of user attention. Despite the great importance of this limitation, which prevents the deployment of these systems outside the laboratory, it is often overlooked in research articles. This study was aimed to propose a novel method to solve this problem, taking advantage of deep learning for the first time in this context to overcome the limitations of previous strategies based on hand-crafted features. *Methods*. The proposed method, based on EEG-Inception, a novel deep convolutional neural network, divides the problem in 2 stages to achieve the asynchronous control: (*i*) the model detects user's control state, and (*ii*) decodes the command only if the user is attending to the stimuli. Additionally, we used transfer learning to reduce the calibration time, even exploring a calibration-less approach. *Results*. Our method was evaluated with 22 healthy subjects, analyzing the impact of the calibration time and number of stimulation sequences on the system's performance. For the control state detection stage, we report average accuracies above 91% using only 1 sequence of stimulation and 30 calibration trials, reaching a maximum of 96.95% with 15 sequences. Moreover, our calibration-less approach also achieved suitable results, with a maximum accuracy of 89.36%, showing the benefits of transfer learning. As for the overall asynchronous system, which includes both stages, the maximum information transfer rate was 35.54 bpm, a suitable value for high-speed communication. *Conclusions*. The proposed strategy achieved higher performance with less calibration trials and stimulation sequences than former approaches, representing a promising step forward that paves the way for more practical applications of ERP-based spellers.

1. Introduction

Brain-computer interfaces (BCI) based on visual event-related potentials (ERP) are a promising technology for alternative and augmented communication in an assistive context, directly decoding the user's brain signals to provide a new channel of communication for people with severe motor disabilities [1]. These systems take advantage from the natural response of the brain to external visual stimuli, which generates waveforms that can be detected in the electroencephalography (EEG) [1]. There are many stimulation paradigms that elicit ERPs with different characteristics, but the most extended in BCI is the *oddball* paradigm [2]. In this paradigm, the subject has to identify and respond to an infrequent target stimulus amid different and more frequent stimuli, triggering an ERP known as P300 for its distinctive positive peak 300 ms after the target stimulus onset [2]. A common implementation of the *oddball* paradigm is the ERP-based speller, which displays on a screen several options or commands that are sequentially highlighted [3]. To select one of the options, the user has to stare at the desired command, triggering a P300 response whenever they perceive the target stimulus [3]. Then, the system de-

* Corresponding author.
E-mail addresses: eduardo.santamaria@gib.tel.uva.es (E. Santamaría-Vázquez), victor.martinez@gib.tel.uva.es (V. Martínez-Cagigal), sergio.perez@gib.tel.uva.es (S. Pérez-Velasco), diego.marcos@gib.tel.uva.es (D. Marcos-Martínez), robhor@tel.uva.es (R. Hornero).

https://doi.org/10.1016/j.cmpb.2022.106623

tects these ERPs and decodes the command that the user wanted to select. Of note, each target is usually highlighted several times in each trial to increase the robustness of the system due to the low signal-to-noise ratio (SNR) of the ERPs [3]. Using this strategy, ERP-based spellers have shown advantages in comparison to other BCIs for their high accuracy, large number of possible choices and adaptability to different contexts, allowing to control complex applications such as web browsers or home automation systems [4]. Moreover, recently developed models based on deep learning have improved the performance of these systems significantly, showing very promising results [5,6].

Despite these advances, there is still a major drawback that is often overlooked: ERP-based spellers are synchronous systems. By default, it is assumed that the user is always interacting with the speller (i.e., control state), systematically selecting a command in each trial [7]. This synchronous behaviour is not suitable for practical applications, where the user should be able to switch between different tasks swiftly by simply ignoring the stimuli (i.e., non-control state) without the intervention of a supervisor [7]. In fact, for ERP-based spellers to be successful in real-world environments, a robust asynchronous control is a key requirement. An illustrative example would be a system for wheelchair control, where the user will only interact with the system when he wants to move. In this context, an undesired selection (e.g., move forward, move back, etc) is not acceptable. Unfortunately, this issue is still far from being fully resolved, and the dynamic detection of the user's control state over the system through monitoring of user attention has proven to be a challenge as hard as command decoding [7].

The ideal solution to this problem is the dynamic detection of the user's control state for each trial to turn the inherently synchronous behaviour of ERP-based spellers into asynchronous, avoiding undesired selections when the user is not interacting with the system [8]. In recent years, several studies addressed this limitation. Table 1 summarizes the key points of these studies, which followed 2 main strategies. The most extended approach is to define a threshold on the output score of the command decoding algorithm [7–13]. These methods assume that the command selection has low confidence (i.e., score below the defined threshold) whenever the user is not attending the stimuli, allowing the system to ignore the selection. Nevertheless, these approaches are greatly affected by non-stationary properties of the EEG over time that modify the probability distribution of the classifier scores for ERP detection [1]. Even slight differences in amplitude and latency of ERPs or impedance and position of sensors can invalidate the threshold. In our own experience, the performance of this approach is reduced drastically in short periods of time and requires frequent recalibration, making them unpractical [9,13]. More advanced techniques used specific neural activity associated with the operation of ERP-based spellers [14–17]. These studies showed that there are measurable patterns in the EEG that can be detected only when the user is interacting with the system, allowing to discriminate the control state using features based on fast Fourier transform (FFT), canonical correlation analysis (CCA), power spectral density (PSD) and sample entropy (SampEn). In general, these methods showed greater robustness and performance than thresholds [15]. However, the design of hand-crafted features to discriminate the user's control state in ERP-based spellers is complex, especially taking into account the effect of inter-subject and inter-session variability. Therefore, the probability of loosing discriminative information in this process is high, often resulting in a suboptimal feature set.

In this context, novel approaches for control state detection could help to overcome current limitations. Particularly, deep-learning models showed excellent results in other BCI areas, such as ERP, SMR and SSVEP classification, for their ability to extract complex features from raw signals [18]. In fact, these methods not only increase the classification accuracy in these tasks, but also can take advantage from cross-subject transfer learning to reduce the calibration time [6]. Thus, deep-learning approaches have great potential to improve the control state detection stage. Nonetheless, to the best of our knowledge, deep-learning models have not been explored for this purpose yet.

The main goal of this study is to design, develop and validate a novel method to achieve an accurate asynchronous control of ERP-based spellers by means of deep learning. Concretely, the proposed method is based on EEG-Inception, a novel deep convolutional neural network (CNN) specifically designed for EEG processing [6]. To this end, we divide the problem in two stages: control state detection and command decoding. Each stage uses a specialized model, allowing to detect the user's control state independently of the command decoding task. This approach has been validated in an experiment that involved 22 healthy subjects, the largest sample among related studies, assuring the generalization of our results. In order to promote future research in the field, the dataset, along with useful code to replicate the results presented in this paper, has been made publicly available at https://www.kaggle.com/esantamaria/asynchronous-erpbased-bci.

2. Methods

2.1. Subjects and signals

Twenty-two healthy subjects (age: 24.7±4.3 years; 15 males) participated in the experiments. All participants had normal or corrected-to-normal vision. The experimental protocol was approved by the local ethics committee and all participants gave their informed consent.

Signals were recorded using a g.USBampg (g.tec medical engineering, Austria) with a sample frequency of 256 Hz and using 8 active electrodes in positions Fz, Cz, Pz, P3, P4, PO7, PO8, Oz according to the international 10-10 system. The ground and reference were placed at FPz and the earlobe, respectively. This montage was proposed by Krusienski et al. [19] for ERP detection and is commonly used for ERP-based spellers. A novel python-based BCI platform, called Medusa, was used to record the signals and display the stimulation paradigm [20].

2.2. Experimental setup

Participants were sat on a comfortable chair in front of 2 screens keeping a distance of 50 cm, as displayed in Fig. 1a. The screen on the right showed the BCI application, whereas the screen on the left displayed a web browser. Accordingly, the experiment comprised 2 different procedures: the control task and the non-control task. In the control task, participants were asked to make selections with an ERP-based speller using the row-column paradigm (RCP) [3]. In this paradigm, commands are displayed in a matrix, whose rows and columns are highlighted sequentially in random order. When each row and column is highlighted once, the algorithm completes a sequence. Thus, participants had to stare at the desired command, which was indicated by the supervisor. Of note, participants were instructed to mentally count the stimuli on the target to maintain the concentration [19]. For this task, we used the 6×6 matrix displayed in Fig. 1a, with an inter-stimulus interval (ISI) of 100 ms and a stimulus duration (SD) of 75 ms. The target commands were selected randomly. In the non-control task, participants had to use the web browser at their will to read a document or watch a video while ignoring the stimuli on the right screen, simulating the real use of the system for assistive applications.

The experiment flow is described in Fig. 1b. The experiment comprised 2 sessions of 10 runs (i.e., 5 control and 5 non-control),

Table 1
Summary of former asynchronous ERP-based spellers.

Study	Paradigm	Strategy	Description of the method for control state detection
Zhang et al. 2008 [7]	RSVP	Analysis of output scores for ERP detection	ROC threshold using SVM scores for ERP detection
Aloise et al. 2011 [8]	RCP		ROC threshold using LDA scores for ERP detection
Martínez-Cagigal et al. 2017 [9]	RCP		ROC threshold using LDA scores for ERP detection
He et al. 2017 [11]	RCP		Classification of SVM scores for ERP detection using an additional SVM
Tang et al. 2018 [10]	RCP		ROC threshold using LDA scores for ERP detection
Aydin et al. 2018 [12]	RBP		ROC threshold using classifier labels for ERP detection
Martínez-Cagigal et al. 2019 [13]	RCP		ROC threshold using LDA scores for ERP detection
Pinegger et al. 2015 [14]	RCP	Hand-crafted features	Threshold using FFT features combined with ROC threshold using LDA scores
Martínez-Cagigal et al. 2019 [16]	RCP		SampEn features and LDA classification
Santamaría-Vázquez et al. 2019 [15]	RCP		PSD and CCA features and LDA classification
Gong et al. 2020 [17]	RCP		FFT features and LDA classification

RSVP: rapid serial visualization paradigm; RCP: row-column paradigm; RBP: region-based paradigm; ROC: receiver operating characteristic; SVM: support vector machine; LDA: linear discriminant analysis; ERP: event-related potentials; SSVEP: steady-state visual evoked potentials; SMR: sensorimotor rhythms; FFT: fast Fourier transform; SampEn: sample entropy; PSD: power spectral density; CCA: canonical correlation analysis.

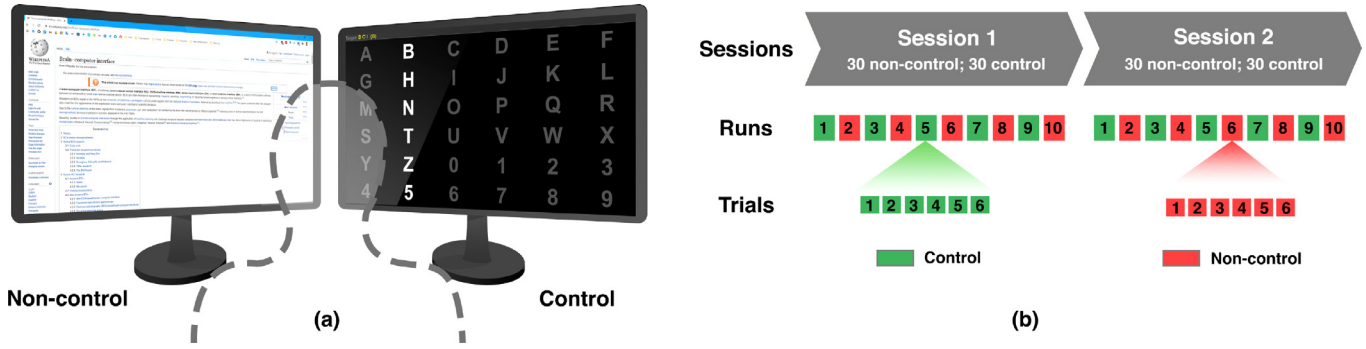

Fig. 1. Experimental setup. (a) Schematic representation of the subject and both screens. The screen on the left displayed the browser that was used during the non-control task, whereas the speller was showed on the right screen. Although the paradigm was active during both tasks, subjects only had to attend to the stimuli during the control task. (b) Overview of the experiment, which comprised 2 sessions of 10 runs, 6 trials of 15 sequences each. Both tasks were intercalated to avoid excessive fatigue of the subject.

which had 6 trials of 15 sequences each. Noteworthy, the tasks were intercalated in order to avoid excessive fatigue. Therefore, the database was composed by 60 control trials and 60 non-control trials for each subject.

2.3. Proposed method for control state detection

In this study, EEG-Inception was used to detect the user's control state and decode the commands in the proposed BCI. This CNN, specifically designed for EEG processing, was presented in our previous work [6], showing excellent results for synchronous ERP-based spellers. Nevertheless, to the best of our knowledge, neither EEG-Inception nor any other deep-learning model has been used to discriminate the user's control state in ERP-based spellers yet.

The architecture of EEG-Inception, which is shown in Fig. 2, is composed by 2 Inception modules and an output block. The first Inception module includes 3 branches that perform 2D convolutions in the temporal axis (i.e., EEG samples) followed by depth-wise convolutions in the spatial axis (i.e., EEG channels). Each branch has filters with different receptive fields (i.e., kernel sizes) to process the signal in different temporal scales. The second In-

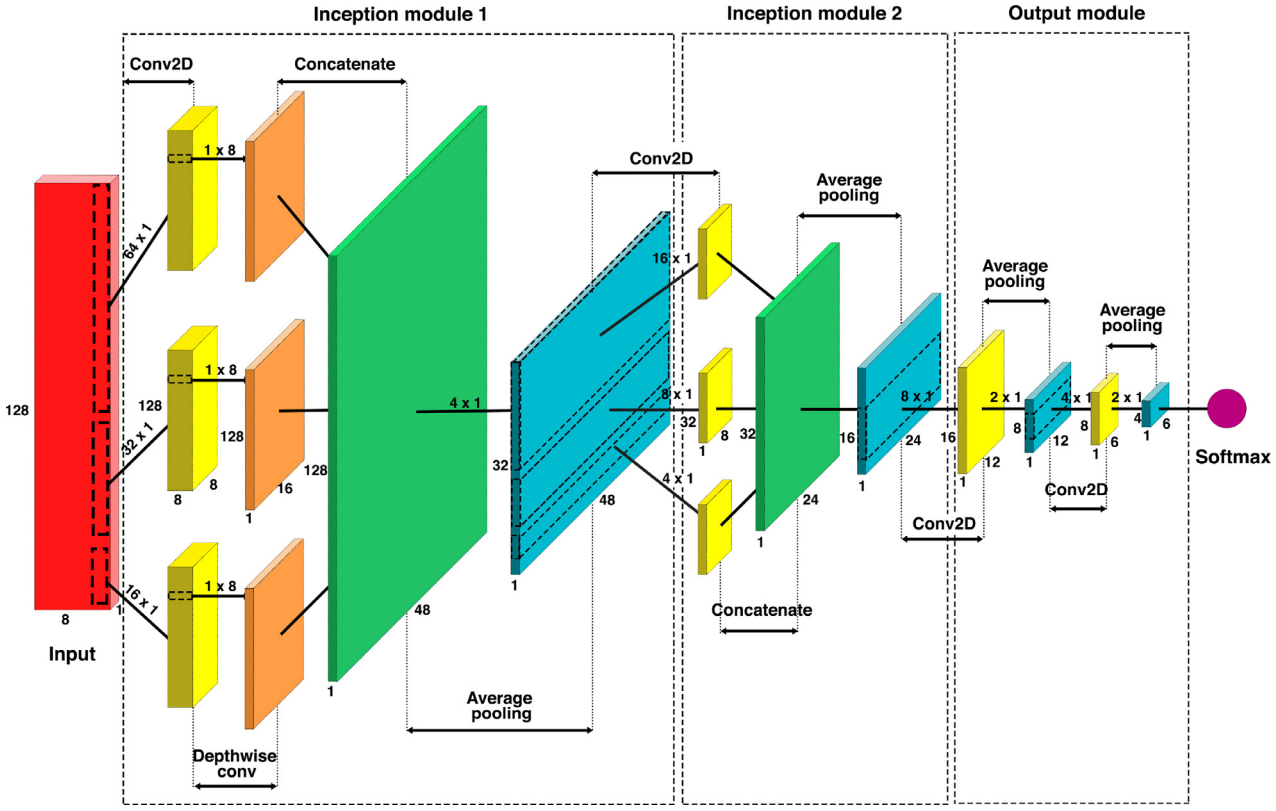

Fig. 2. Schematic representation of EEG-Inception. All convolutional layers (i.e., Conv2D and Depthwise convolutions) include batch normalization, ELU activation and dropout.

ception module only includes temporal convolutions, extracting features in the same temporal scales but taking into account the available spatial information. Finally, the output block synthesizes the information extracted by the previous modules in few high-level features that are classified with a softmax output, following a bottleneck structure specifically designed to avoid overfitting. Additionally, average pooling, batch normalization, exponential linear unit (ELU) activations and dropout normalization were used to improve the performance of the model [6]. Please refer to the original study for additional information and an open source implementation [6].

In contrast to preceding deep-learning models for EEG processing, which implemented single-scale approaches, EEG-Inception process the input signal in multiple temporal scales, increasing its adaptability to different tasks [6]. Thus, we hypothesized that this architecture could be applied to provide a robust asynchronous control of ERP-based spellers, targeting different patterns associated with the operation of ERP-based spellers, such as steady-state visual evoked potentials (SSVEPs) provoked by stimuli at constant rates [14], or measurable differences in the EEG complexity, especially in the prefrontal cortex, related to the concentration state of the user [16]. To this end, the signal processing pipeline must be adapted to facilitate the integration of EEG-Inception for this task. As exposed in the introduction, previous approaches based on hand-crafted features [14–17] used 2 separated processing pipelines with different preprocessing, feature extraction and classification methods: one to detect the control state and the other to decode commands. This implies duplicating the computing cost and increasing the complexity of the system, which could be a limitation in online experiments, especially for BCIs implemented in portable devices (e.g., smartphones and tablets) [13]. In order to avoid this problem, our method uses the same observations for both classification tasks, using the same preprocessing and signal conditioning. Then, 2 different EEG-Inception instances are used for each of the 2 classification tasks. The complete processing pipeline has 4 stages:

2.3.1. Preprocessing.

In this stage, raw EEG is preprocessed to increase the SNR of the target signals. First, the signal is filtered between 0.5 and 45 Hz with a finite impulse response filter and resampled to 128 Hz, keeping the most discriminative information for control state and ERP classification [15]. Then, common average reference (CAR) is used to remove noisy artifacts [21].

2.3.2. Feature extraction.

Deep CNNs extract features from raw EEG automatically thanks to their multi-layer design, learning hierarchical representations of the data at different levels of abstraction [22]. Nevertheless, the input signal must be prepared to make observations with the shape expected by the model. To this end, we extracted the epochs of signal for each stimulus from 0 to 1000 ms after the onset. Additionally, z-score normalization was applied taking a baseline window of 250 ms before the stimulus onset. At the end of this process, each observation had 128 samples $\times$ 8 channels, which are the input dimensions required by EEG-Inception [6]. Taking into account that the experiment comprised 2 sessions per subject, 10 runs per session, 6 trials per run, 15 sequences per trial and 12 stimuli per sequence (6 rows and 6 columns), the total number of EEG epochs for each subject was 21,600. This makes a total of 475,200 observations for the 22 subjects.

2.3.3. Control state detection.

This stage dynamically detects the user's control state to turn the speller into an asynchronous system. The workflow in this stage is as follows: (*i*) the epochs of each trial are fed to the model

trained to discriminate between control and non-control states; (*ii*) the model outputs one score between 0 and 1 per observation, representing the probability of each state; (*iii*) the scores of the trial are averaged in a post-processing stage that determines non-control state if the probability is less than 0.5 and control state otherwise. If non-control state is determined, the system starts a new trial without selecting a command or giving feedback to the user. On the other hand, if control state is determined, the system continues to the next stage to decode the command. Note that the algorithm assumes that all the observations of each trial have the same control state. Therefore, the user should not switch tasks before the trial ends.

2.3.4. Command decoding.

Once the system has determined the control state for a trial, this stage decodes the command the user wanted to select. The strategy is similar to the control state detection stage. The observations are fed into the model trained to discriminate between target, which are characterized by an ERP with the P300 response, and non-target epochs. Therefore, the output is again a score between 0 and 1 representing the probability of each case [6]. In this case, the output scores are associated with the row and column that were highlighted and thus, they are averaged according to this association. The command corresponding to the row and column with maximum score is then selected by the system, which gives the proper feedback to the user. After this stage, a new trial begins and the cycle is repeated.

2.4. Training/testing strategy and model validation

In order to train and validate our approach, we simulated the real use of the speller using leave-one-subject-out (LOSO) cross validation combined with cross-subject transfer learning and fine-tuning [23,24]. For each iteration of the LOSO algorithm, the models for control state detection and command decoding were initialized with the training subjects. This means that the control state detection model was initialized with 453,600 observations (21 subjects $\times$ 120 trials $\times$ 180 observations/trial), whereas the command decoding model was initialized with 226,800 observations (21 subjects $\times$ 60 control trials $\times$ 180 observations/trial). Then, the models were fine-tuned using $N = \{0, 5, 10, 20, 30\}$ control trials from the test subject for the command decoding model, and $2N$, N control and N non-control, for the control state detection model. The fine-tuning trials were randomly selected and were not used for testing. Therefore, the number of test trials for each N was the total number of trials minus the number of fine-tuning trials. This procedure was repeated 100 times for each subject, averaging results to achieve a robust validation. This analysis allows to study the dependence of the system's performance on the number of training trials, where $N = 0$ simulates a plug-and-play device and $N = 30$ requires a calibration session of approximately 30 minutes, including control and non-control trials.

Regarding the training process, both models were trained separately with different labels. In the case of the control state detection model, EEG epochs were labelled according to the control and non-control classes. Specifically, the epochs where the user was attending to the stimuli were labelled as control (positive class), whereas the epochs where the user was using the web browser were labelled as non-control (negative class), resulting in a balanced dataset. For the command decoding model, only control trials were used for training. In this case, target epochs were labelled as P300 (positive class), and the epochs corresponding to non-target commands were labelled as non-P300 (negative class). Models were trained using the same configuration: Adam optimizer with default hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$; categorical cross-entropy loss; batch size of 1024; and a maximum of 500 training iterations over the entire dataset, applying early stopping when the validation loss did not improve for 10 consecutive iterations and restoring the weights that minimized this metric [6].

The proposed training/testing approach has several advantages. For instance, it allows to have subject specific models with very few training trials by exploiting cross-subject transfer learning [6]. Ideally, in the initialization phase, the model will learn common features across subjects to detect the target patterns in each case. Then, in the fine-tuning phase, the model will particularize these features to the specific characteristics of each subject, resulting in an improved performance [24]. In fact, this strategy proved to be more adequate for deep-learning approaches in BCI than the classic intra-subject or cross-subject training methods, since these models are able to extract high-level features robust to inter-subject variability [6]. At the same time, this method allows to take advantage from all the available signals, improving its scalability for real use, where data from new subjects could be incorporated to the models to increase their performance.

3. Results

3.1. Control state detection

Fig. 3 and Table 2 summarize the results of the cross validation experiment for control state detection, showing the normalized confusion matrices and accuracy averaged across all subjects broken down by the number of fine-tuning trials and stimulation sequences. Both analysis give a complete overview of the system's performance in this task. Accuracy is the most widely used metric, accounting for the percentage of trials correctly classified. Therefore, it allows easy and direct comparison with former works. On the other hand, confusion matrices provide more complete information about the model performance with useful insight on the distribution of false positives and false negatives that can help to understand the system dynamics. Additionally, Fig. 4 characterizes the EEG in time and frequency to analyze differences between correctly and incorrectly classified trials in the control state detection task and understand which could be the main factors affecting the performance of the model.

Overall, the proposed method was able to discriminate the control state with high accuracy. As expected, a greater number of sequences, which implies more observations for each trial, allowed to increase the confidence of the selection and reduce the impact of outliers, thus increasing the system's performance. On the other hand, more stimulation sequences imply to reduce the selection speed of the system. Our approach stands out especially in this point, reaching accuracies above 91% with only 1 sequence of stimulation for $N = 20$ and $N = 30$, a suitable value for high-speed communication. The number of fine-tuning trials also proved to be important to achieve peak performance in exchange for increasing the calibration time. Nevertheless, the proposed approach also reached suitable accuracies even with none or very few observations. In fact, for $N = 0$, which simulates a plug-and-play device with no calibration for the test subject, the model already achieved accuracies near 90%. This proves the efficacy of our training strategy, which uses cross-subject transfer learning to initialize the model with different subjects. Furthermore, the fine-tuning process for $N > 0$ was able to adapt the model to the individual characteristics of each test subject, increasing the performance of the method after a short calibration. For instance, the accuracy for control state detection with $N = 30$ achieved 91.91% and 96.95% using 1 and 15 sequences of stimulation, respectively. In this regard, Fig. 5 shows the training graphs for 1 subject for each N and averaged across the 100 repetitions. As can be seen, the convergence of the model is more consistent for higher values of N. Nevertheless,

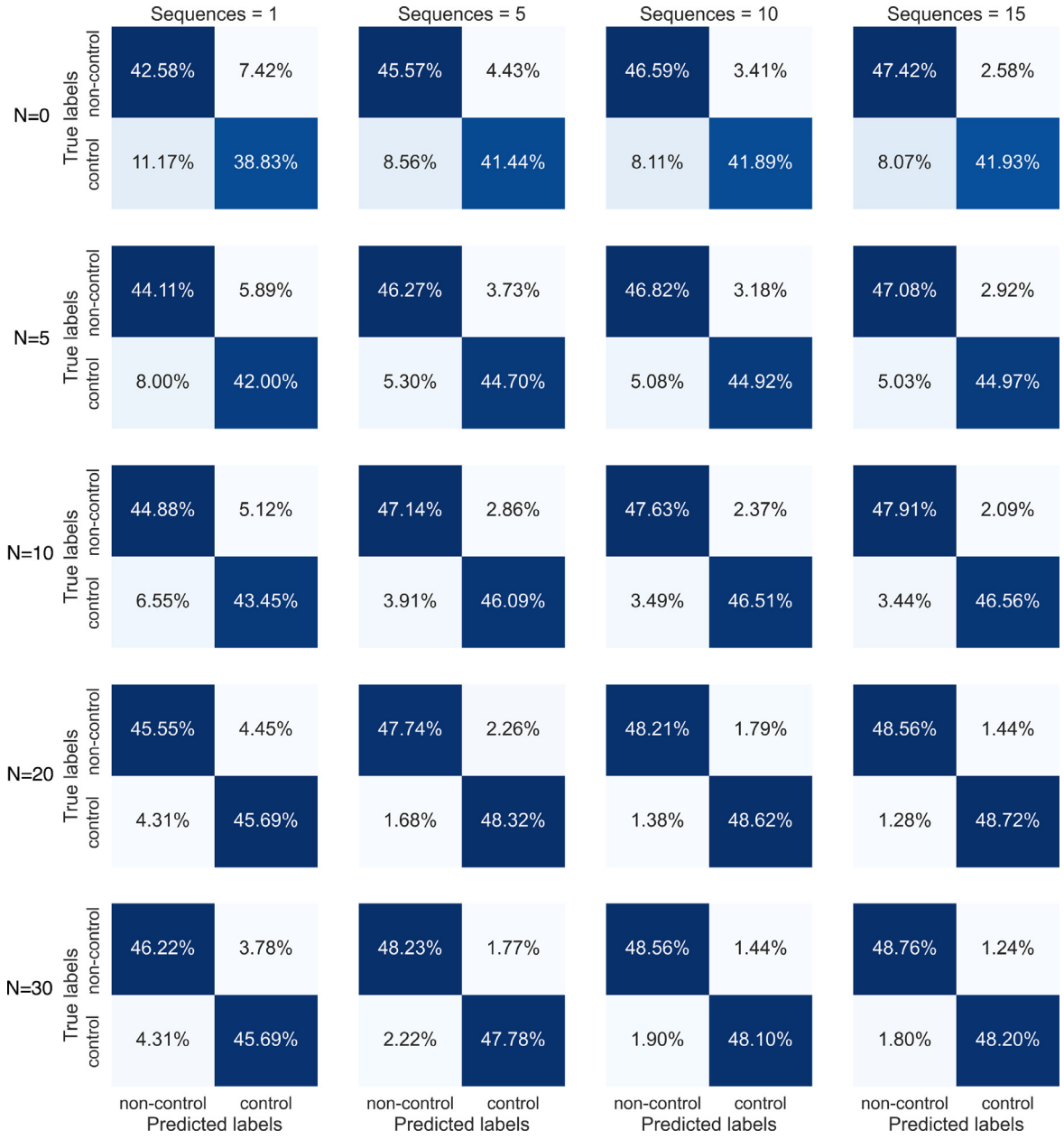

Fig. 3. Normalized confusion matrices averaged across subjects.

Table 2
Control state detection accuracy (%).

N	No. Sequences														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	81.4	85.23	86.63	86.89	87.01	87.69	88.26	88.22	87.95	88.48	89.02	89.05	89.28	89.13	89.36
5	86.11	88.96	89.99	90.62	90.98	91.06	91.29	91.45	91.63	91.74	91.89	91.91	91.98	92	92.05
10	88.33	91.16	92.27	92.84	93.23	93.47	93.61	93.82	94.05	94.14	94.23	94.3	94.29	94.42	94.48
20	91.24	94.37	95.31	95.76	96.06	96.31	96.45	96.6	96.69	96.83	96.98	97.06	97.07	97.21	97.28
30	91.91	94.41	95.23	95.67	96.02	96.33	96.53	96.53	96.52	96.66	96.77	96.76	96.85	96.88	96.95

N: number of fine-tuning trials in control state for each subject. Thus, the total number of calibration trials used to fine-tune the model for this task was 2N (i.e., N control, N non-control). Test accuracy (%) for the control state detection task averaged over the 22 subjects.

the fine-tuning process is beneficial even with very few training examples, and helps the model to learn subject-specific features.

Regarding the normalized confusion matrices, Fig. 3 shows that the percentage of false negatives tend to be higher than the percentage of false positives, especially for lower values of *N*. This difference can be explained taking into account the workflow of the system. During control trials, subjects had to stare at the desired command to select it, a task that requires a high level of concentration and mental effort. Conversely, for non-control trials, subjects had to ignore the stimuli of the RCP while watching a video or reading a web page. Therefore, we hypothesize that if the subject loses concentration their EEG would be similar to that

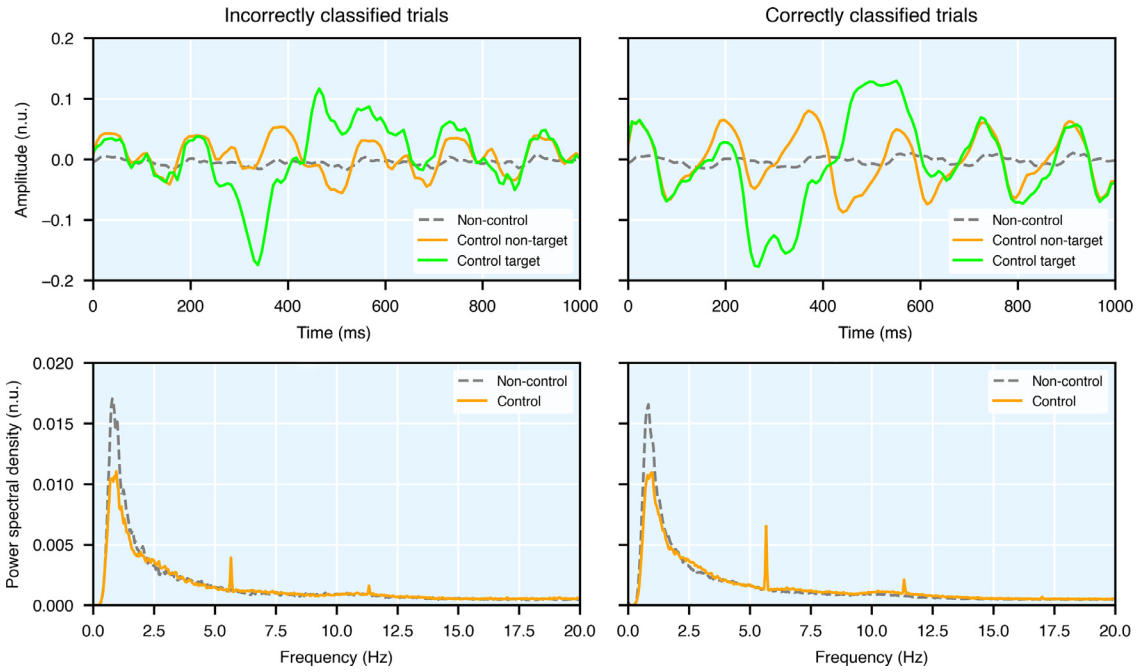

Fig. 4. n.u: normalized units. Characterization of correctly and incorrectly classified trials for the control state detection task. The upper graphs show the averaged EEG epochs for the 3 different conditions: non-control, control non-target and control target. The lower graphs show the power spectral density of the entire trials.

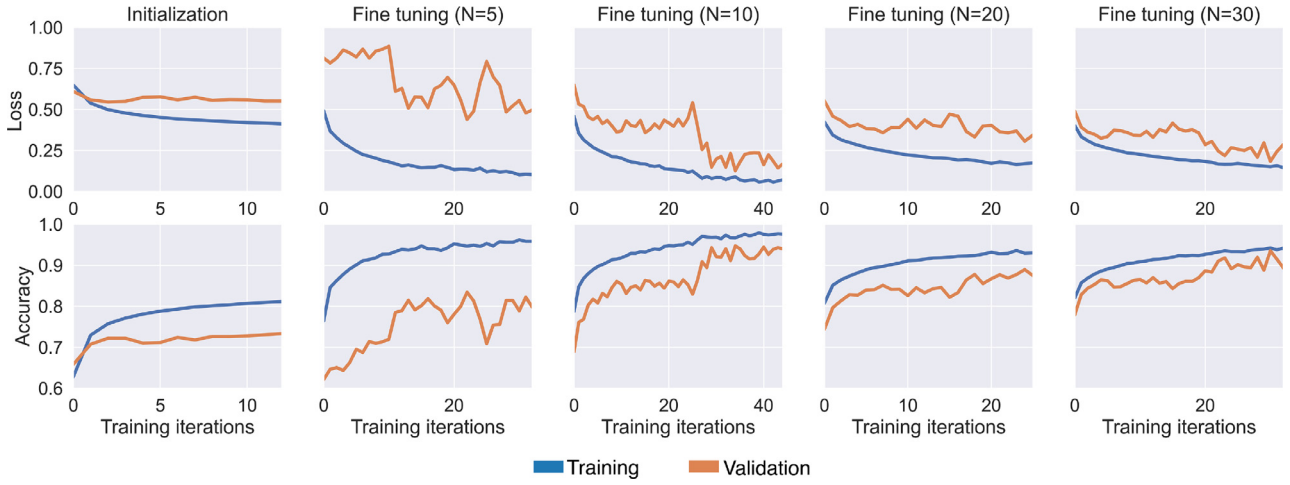

Fig. 5. Training graphs for one subject averaged across the 100 iterations of the cross-validation algorithm. The convergence of the model is more consistent for higher values of *N*.

during the non-control task, confusing the model (false negative). The opposite is more unlikely because the visual stimulation received by the subject provokes specific patterns on the EEG that are not present in non-control trials, such as ERPs and SSVEPs [15]. This hypothesis is supported by the characterization of the EEG for control state detection task shown in Fig. 4. As can be seen, the averaged waveforms associated to the control state were weaker for misclassified epochs. The amplitude of the ERP was lower and the shape was less clear in these trials (see green curve in time graphs). Similarly, the SSVEP was noisy and with significantly less power too (see the orange curve in time graphs and the peaks at 5.71 Hz and harmonic at 11.42 Hz in frequency graphs). In this regard, the frequency of the SSVEP corresponds to the stimulation rate in our experiment, which was 1 / (ISI + SD) = 1 / (0.1 + 0.075) = 5.71 Hz. These differences may be caused by fatigue and momentary loss of concentration during the control task due to the strong visual stimulation, which has proven to be an issue in previous synchronous ERP-based spellers [25].

3.2. Overall system

Table 3 shows results including both stages for control state detection and command decoding, considering that both classifications must be correct at the same time. Therefore, if one stage fails, it is considered as a mistake. As before, the accuracy is averaged across subjects and broken down by the number of fine-tuning trials and stimulation sequences. Additionally, Fig. 6 shows the theoretical information transfer rate (ITR) reached by the asynchronous speller in bits per minute (bpm). This metric takes into account the speed and the accuracy of the system, allowing a direct comparison between different BCIs [26]. The ITR was calculated with the following equation [26]:

$$ITR = \left(\log_2 N_s + P\log_2 P + (1-P)\log_2 \frac{1-P}{N_s-1}\right)S, \quad (1)$$

Table 3
Overall system accuracy, including control state detection and command decoding. (%).

N	No. Sequences														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	52.05	60.49	66.7	70.00	73.45	75.49	77.77	79.43	79.62	80.95	82.20	83.26	83.94	84.24	85.38
5	59.70	70.30	76.05	80.77	83.48	85.11	86.47	87.76	88.6	89.22	89.72	90.09	90.33	90.68	90.99
10	62.63	73.73	80.34	84.55	87.09	88.75	90.07	91.27	91.99	92.44	92.78	93.11	93.20	93.58	93.79
20	64.90	77.34	83.98	88.20	90.73	92.32	93.32	94.42	94.98	95.49	95.82	96.21	96.20	96.51	96.68
30	66.49	78.20	84.33	88.69	91.30	92.96	93.71	94.61	94.95	95.34	95.70	96.01	96.09	96.30	96.47

N: number of fine-tuning trials in control state for each subject. Thus, the total number of calibration trials used to fine-tune the model for control state detection was 2N (i.e., N control, N non-control), whereas the model for command decoding only used N control trials. Overall test accuracy (%) for the control state detection and command decoding tasks averaged over the 22 subjects. Noteworthy, one trial is considered correct only if both conditions were correctly classified at the same time.

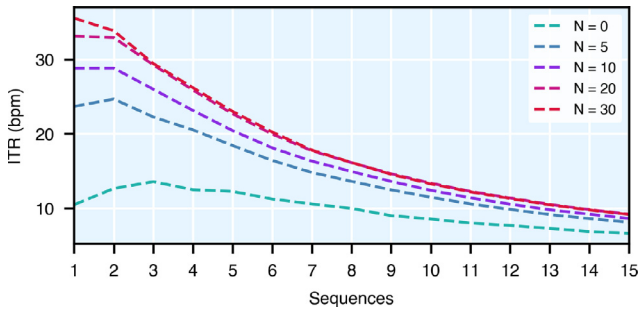

Fig. 6. ITR: information transfer rate (bpm); *N*: number of fine-tuning trials in control state for each subject. Average ITR including control state detection and command decoding stages and only considering control trials.

where N_s is the total number of targets, P is the accuracy, and S is the number of selections per minute. It should be noted that the ITR was calculated only for control trials but considering the overall accuracy, including control state detection and command decoding stages.

As shown in Table 3, the system also achieved high overall performance. For instance, a test accuracy of 91.3% with $N = 30$ and 5 stimulation sequences could be a suitable value for practical applications, taking into account that the system would be fully asynchronous. Moreover, as can be seen in Fig. 6, the system reached a maximum average ITR of 35.54 bpm for $N = 30$ and 1 sequence, with a peak ITR of 88.60 bpm for one subject. Regarding our calibration-less approach (i.e., $N = 0$), the maximum accuracy was 85.38%, and the maximum ITR 13.57 bpm.

4. Discussion

4.1. Comparative analysis

The proposed method achieved promising results in the control state detection and command decoding tasks. Moreover, this was achieved with the largest number of subjects among related studies, which assures the generalization of our results. Nevertheless, a direct comparison with previous studies [7–17] is difficult in many cases due to important differences in experimental setups, analyses and number of subjects. For these reasons, we only compare results with studies that used the RCP as stimulation paradigm. Statistical differences were evaluated with Mann-Whitney *U*-test when results broken down by subject were available, correcting the False Discovery Rate for multiple comparisons with Benjamini-Hochberg approach.

The performance for the control state detection task was significantly higher than preceding approaches based on hand-crafted features. Pinegger et al. [14] reached 79.5% accuracy with 15 sequences of stimulation using FFT features, compared to 96.95% in this work (p-value < 0.01). Similarly, the proposed method also outperformed approaches based on PSD, CCA and SampEn features with paired number of sequences and $N = 30$ (p-value < 0.05) [15,16]. Interestingly, these differences are maximized for few stimulation sequences. In fact, taking results from 1 and 5 stimulation sequences, the average accuracy was improved in this work by 10.41% and 19.60% with respect to these 2 studies. Therefore, the proposed strategy is a significant step forward towards practical asynchronous ERP-based spellers that require high-speed communication.

With respect to the overall system results, there are also some points that are worth discussing. The analyzed studies reached lower performance in terms of overall accuracy and ITR, demonstrating the superiority of our proposal. In this work, the maximum average ITR was 35.54 bpm. In comparison, Zhang et al. [7], Aloise et al. [8] and Santamaría-Vázquez et al. [15] reported maximum ITRs of 15.0 bpm, 11.2 bpm (p-value < 0.01) and 12.3 bpm (p-value < 0.01), respectively. On the other hand, Tang et al. [10] reported a maximum average accuracy of 90.30% compared to 96.47% in this study.

Regarding the calibration time, this is the first work that explored an asynchronous ERP-based speller without calibration for the test subject (i.e., $N = 0$). This approach achieved satisfactory performance for the control state detection and command decoding tasks, with a maximum overall accuracy of 85.38% and maximum ITR of 13.57 bpm. These results are above the minimum performance of 70% required for successful control of BCIs [27]. In our opinion, the reduction, and even suppression, of the calibration stage is key for the development of asynchronous ERP-based spellers outside the laboratory, increasing their usability for practical applications. Therefore, we consider this point as one of the strengths of our study, paving the way for future efforts in this line.

4.2. Contributions

This study puts forward a novel and more practical signal processing framework to achieve a robust asynchronous control of ERP-based spellers taking advantage of deep-learning to avoid the use of thresholds or hand-crafted features. The proposed method not only reached higher performance than the approaches presented in related studies, but also solved some of the main drawbacks that limited the use of these systems for practical applications. Firstly, the control state detection and command decoding stages are independent, solving the instability and calibration complexity of coupled methods based on thresholds, which are vulnerable to the dynamic properties of the EEG over time and need to be recalibrated several times per session to maintain peak accuracy, especially with challenging subjects [7–13]. Secondly, the automatic extraction of optimal features using EEG-Inception solves the limitation of methods based on FFT, PSD, CCA and SampEn, whose accuracy is drastically reduced for few stimu-

lation sequences, where our method showed clear advantages [14–17]. Finally, it should be noted that none of the related works explored the benefits of cross-subject transfer learning or fine tuning so far. Therefore, our proposal is the first to take advantage of these methods to reduce the number of calibration trials in asynchronous ERP-based spellers, even simulating a plug-and-play device with fair results. Ideally, this training strategy could rise the accuracy of calibration-less approaches to the level of fine-tuned models, given the ability of deep neural networks to make the most of large amounts of data to reach robust classification [22].

4.3. Limitations and future work

Despite the successful results achieved in this work, several limitations must be considered. For instance, we did not test the designed speller with motor disabled subjects, the target users of these systems [26]. In this regard, numerous studies proved that field experiments outside the laboratory with real applications and users could affect the final performance of the system [9,13]. For this reason, additional experiments are needed to study the robustness of EEG-Inception for control state detection in different scenarios and databases. Moreover, our framework has been tested with the RCP. However, more engaging stimulation paradigms, such as the Face Speller [28] or motion-VEP-based systems [29], can help to decrease the false negative rate due to fatigue or loss of concentration. As the confusion matrices showed, this effect had an important impact on our experiments. Therefore, the use of these stimulation paradigms, rather than the RCP, could help to increase the performance by improving the subject's attention level. In fact, to the best of our knowledge, the asynchronous control of these BCIs has not been tested yet, representing a promising future research line.

5. Conclusion

In this study, we explored a novel method to achieve effective asynchronous control of ERP-based spellers using deep learning. The proposed speller takes advantage of EEG-Inception, a novel CNN specifically designed for EEG processing and ERP detection, to reach significantly higher performance than previous works. We reported accuracies above 91% for the control state detection with 1 sequence of stimulation, a suitable value for high-speed communication with asynchronous ERP-based spellers, and a maximum ITR of 35.54 bpm for the overall asynchronous system. Moreover, we used a novel training strategy based on cross-subject transfer learning and fine tuning to reduce the calibration time in comparison to previous studies, even exploring a calibration-less approach. Additionally, the proposed signal processing framework simplifies the design of the system and solves the main limitations of former approaches, increasing its feasibility for practical applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research has been developed under the grants PID2020-115468RB-I00 and RTC2019-007350-1 funded by 'Ministerio de Ciencia e Innovación/Agencia Estatal de Investigación/10.13039/501100011033/' and European Regional Development Fund (ERDF) A way of making Europe; under the R+D+i project 'Análisis y correlación entre la epigenética y la actividad cerebral para evaluar el riesgo de migraña crónica y episódica en mujeres' ('Cooperation Programme Interreg V-A Spain-Portugal POCTEP 2014–2020') funded by 'European Commission' and ERDF; and by 'Centro de Investigación Biomédica en Red en Bioingeniería, Biomateriales y Nanomedicina (CIBER-BBN)' through 'Instituto de Salud Carlos III' co-funded with ERDF funds. E. Santamaría-Vázquez, S. Pérez-Velasco and D. Marcos-Martínez were in a receipt of a grant from the 'Consejería de Educación de la Junta de Castilla y León', and the European Social Fund.

References

[1] J. Wolpaw, E.W. Wolpaw, Brain-computer interfaces: Principles and practice, OUP USA, 2012.
[2] J. Polich, Updating P300: an integrative theory of P3a and P3b, Clinical Neurophysiology 118 (10) (2007) 2128–2148.
[3] L.A. Farwell, E. Donchin, Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials, Electroencephalogr Clin Neurophysiol 70 (6) (1988) 510–523.
[4] L.F. Nicolas-Alonso, J. Gomez-Gil, Brain computer interfaces, a review, Sensors 12 (2) (2012) 1211–1279.
[5] V.J. Lawhern, A.J. Solon, N.R. Waytowich, S.M. Gordon, C.P. Hung, B.J. Lance, EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces, J Neural Eng 15 (056013) (2018) 1–17.
[6] E. Santamaria-Vazquez, V. Martinez-Cagigal, F. Vaquerizo-Villar, R. Hornero, EEG-Inception: A Novel Deep convolutional neural network for assistive ERP-based brain-Computer interfaces, IEEE Trans. Neural Syst. Rehabil. Eng. 28 (12) (2020) 2773–2782.
[7] H. Zhang, C. Guan, C. Wang, Asynchronous P300-based brain-computer interfaces: a computational approach with statistical models, IEEE Trans Biomed Eng 55 (6) (2008). 1754–63
[8] F. Aloise, F. Schettini, P. Aricò, F. Leotta, S. Salinari, D. Mattia, F. Babiloni, F. Cincotti, P300-Based brain-computer interface for environmental control: an asynchronous approach, J Neural Eng 8 (2) (2011).
[9] V. Martínez-Cagigal, J. Gomez-Pilar, D. Álvarez, R. Hornero, An asynchronous P300-Based brain-Computer interface web browser for severely disabled people, IEEE Trans. Neural Syst. Rehabil. Eng. 25 (8) (2017) 1332–1342.
[10] J. Tang, Y. Liu, J. Jiang, Y. Yu, D. Hu, Z. Zhou, Toward brain-Actuated mobile platform, Int J Hum Comput Interact 30 (10) (2019) 846–858.
[11] S. He, R. Zhang, Q. Wang, Y. Chen, T. Yang, Z. Feng, Y. Zhang, M. Shao, Y. Li, A P300-Based threshold-Free brain switch and its application in wheelchair control, IEEE Trans. Neural Syst. Rehabil. Eng. 25 (6) (2017) 715–725.
[12] E.A. Aydin, O.F. Bay, I. Guler, P300-Based Asynchronous brain computer interface for environmental control system, IEEE J Biomed Health Inform 22 (3) (2018) 653–663.
[13] V. Martínez-Cagigal, E. Santamaría-Vázquez, J. Gomez-Pilar, R. Hornero, Towards an accessible use of smartphone-based social networks through brain–computer interfaces, Expert Syst Appl 120 (2019) 155–166.
[14] A. Pinegger, J. Faller, S. Halder, S.C. Wriessnegger, G.R. Müller-Putz, Control or non-control state: that is the question! an asynchronous visual P300-based BCI approach, J Neural Eng 12 (1) (2015) 014001.
[15] E. Santamaría-Vázquez, V. Martínez-Cagigal, J. Gomez-Pilar, R. Hornero, Asynchronous control of ERP-Based BCI spellers using steady-State visual evoked potentials elicited by peripheral stimuli, IEEE Trans. Neural Syst. Rehabil. Eng. 27 (9) (2019) 1883–1892.
[16] V. Martínez-Cagigal, E. Santamaría-Vázquez, R. Hornero, Asynchronous control of P300-Based brain-Computer interfaces using sample entropy, Entropy 21 (3) (2019) 230.
[17] M. Gong, G. Xu, M. Li, F. Lin, An idle state-detecting method based on transient visual evoked potentials for an asynchronous ERP-based BCI, J. Neurosci. Methods 337 (March) (2020) 108670.
[18] A. Craik, Y. He, J.L. Contreras-Vidal, Deep learning for electroencephalogram (EEG) classification tasks: a review, J Neural Eng 16 (3) (2019).
[19] D.J. Krusienski, E.W. Sellers, D.J. McFarland, T.M. Vaughan, J.R. Wolpaw, Toward enhanced P300 speller performance, J. Neurosci. Methods 167 (1) (2008) 15–21.
[20] E. Santamaría-Vázquez, V. Martínez-Cagigal, R. Hornero, MEDUSA: Una nueva herramienta para el desarrollo de sistemas brain-Computer interface basada en python, Cognitive Area Networks 5 (1) (2018) 87–92.
[21] D.J. McFarland, L.M. McCane, S.V. David, J.R. Wolpaw, Spatial filter selection for EEG-based communication, Electroencephalogr Clin Neurophysiol 103 (3) (1997) 386–394.
[22] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[23] I.H. Witten, E. Frank, M.A. Hall, Data mining: practical machine learning tools and techniques, Acm Sigmod Record, 2006.
[24] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, 2016.
[25] M. Xu, X. Xiao, Y. Wang, H. Qi, T.P. Jung, D. Ming, A brain-Computer interface based on miniature-Event-Related potentials induced by very small lateral visual stimuli, IEEE Trans. Biomed. Eng. 65 (5) (2018) 1166–1175.
[26] J.R. Wolpaw, N. Birbaumer, D.J. McFarland, G. Pfurtscheller, T.M. Vaughan, Brain computer interfaces for communication and control, Clinical neurophysiology 4 (113) (2002) 767–791.

[27] A. Kübler, N. Neumann, B. Wilhelm, T. Hinterberger, N. Birbaumer, Predictability of brain-computer communication, J Psychophysiol 18 (2/3) (2004) 121–129.
[28] J. Jin, B.Z. Allison, Y. Zhang, X. Wang, A. Cichocki, An ERP-based BCI using an oddball paradigm with different faces and reduced errors in critical functions, Int J Neural Syst 24 (08) (2014) 1450027.
[29] S. Schaeff, M.S. Treder, B. Venthur, B. Blankertz, Exploring motion VEPs for gaze-independent communication, J Neural Eng 9 (4) (2012), doi:10.1088/1741-2560/9/4/045006.